Data Transmission Schemes for a New Generation of Interactive Digital Television

by

Mehran Azimi

B.Sc., Amir-Kabir (Polytechnic) University of Technology, Tehran , 1994

M.Sc., Sharif University of Technology, Tehran, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES

(Department of Electrical and Computer Engineering)

We accept this thesis as conforming
to the required standard

The University of British Columbia

MARCH 2004

Abstract

Interactive television (ITV) is an attractive technology, which changes the way TV viewers experience home entertainment. In this thesis, we design an interactive television system, which truly gives TV viewers freedom to control and individualize the presentation of TV program content. In this context, we present methods that add extra video and audio streams (called *incidental streams*) containing interactive content, to the transmission line of a digital TV system. The addition of these extra streams does not result in increasing the transmission bandwidth, nor in degrading the picture or sound quality of the main TV program content.

Our design consists of two major transmission mechanisms for transmitting the incidental data, called *deterministic* and *stochastic* service classes. The deterministic service class is designed such that no incidental stream data packet is lost during transmission. On the other hand, the stochastic service class is designed such that some incidental data loss is possible; however, the data loss rate is bounded. We present a strategy based on scalable video coding, which in conjunction with the deterministic and stochastic service classes, achieves the best possible picture and sound quality for the incidental streams under the constraint of available bandwidth.

We also present data transmission methods for the deterministic and the stochastic service classes. In the context of the deterministic service class, we employ a deterministic traffic model for modelling the traffic of main streams, and then design a data transmission scheme based on the 'Network Calculus' theory. In the context of the stochastic service class, we employ Hidden Semi-Markov Models (HSMM) for modelling the traffic of main streams. We then design a data transmission scheme based on the 'Effective Bandwidth' theory.

Furthermore, we design a data multiplexing system for the transmitter head-end of the proposed interactive TV system. This includes a novel scheduling algorithm for controlling the multiplexing of incidental and main streams data packets.

We present numerical results of simulation experiments, which support the theoretical methods presented in this thesis.

Contents

Abstract ii

Contents iv

List of Tables viii

List of Figures ix

Acknowledgements xiii

Dedication xiv

1 Introduction 1
1.0.1 A New Vision of Interactivity for Television 3
1.1 Challenges 5
1.2 Thesis Scope 6
1.3 Our Approach 7
1.3.1 Framework 7
1.3.2 Maximum Waiting Time 8
1.4 Structure and Mechanisms of the System 10
1.4.1 Admission Control 10
1.4.2 Traffic Characterization 11
1.4.3 Service Classes 12
1.4.4 Scalable Coding 13
1.4.5 Data Multiplexing 14
1.5 Thesis Contribution and Structure 15

2 Deterministic Service Class 17
2.1 Introduction 17

2.2 Traffic Characterization : 18
2.2.1 Deterministic Traffic Characterization 19
2.2.2 Parameterized Traffic Characterization Models 21
2.3 Fitting the $(\vec{\sigma}, \vec{\rho})$ Model to a Source 26
2.3.1 Overview of Current Methods for Fitting the $(\vec{\sigma}, \vec{\rho})$ Model to a Traffic Source 28
2.3.2 Our Approach to Constructing the Empirical Envelope 31
2.3.3 Obtaining the $(\vec{\sigma}, \vec{\rho})$ Parameters from the Empirical Envelope 34
2.4 Admission Control 37
2.4.1 Network Calculus Basics 37
2.4.2 Waiting-Time Bound in the proposed ITV application. 40
2.4.3 Our Algorithm to Find T_w 42
2.5 Results 43
2.5.1 Numerical Results of 'Model Fitting' methods 44
2.5.2 Numerical Results of 'Admission Control Mechanism' . 47
2.6 Conclusion 53

3 Stochastic Service Class **56**
3.1 Introduction 56
3.2 Effective Bandwidth 57
3.2.1 Large Deviation Principle 59
3.2.2 Effective Bandwidth Definition 61
3.2.3 Physical Interpretation of Effective Bandwidth 62
3.3 Quantifying Data Loss in Communication Networks Using Effective Bandwidth 63
3.3.1 Data Loss Probability in a Queuing Model with Priority 64
3.4 Admission Control for the Stochastic Service Class 67
3.5 Numerical Estimation of Effective Bandwidth 68
3.6 Conclusion 72

4 Video Traffic Modelling **74**
4.1 Introduction 74
4.2 Existing Stochastic Video Traffic Models 76
4.2.1 Stochastic Models for Full-Screen Broadcast Video . . 80
4.3 Mathematical Background of General Markovian Models . . . 81
4.3.1 Markov Chain 81

4.3.2 Hidden Markov Models 82

4.3.3 Limitation of HMM Models 83

4.3.4 Selecting Between HMM and HSMM Modelling Approach for the Proposed ITV Application 84

4.4 General Background on Semi-Markovian Signal Models 86

4.4.1 Semi-Markov Chains 86

4.4.2 Hidden Semi-Markov Models 89

4.4.3 General Background on Identification of HSMMs . . . 89

4.5 New Formulation of HSMMs 94

4.5.1 Hidden (or Modulating) Layer Modelling 94

4.5.2 Observation Layer Modelling 97

4.5.3 Model Parameterizations 98

4.6 A New Algorithm for Off-line Identification of HSMMs 99

4.6.1 Implementation Issues 103

4.7 Online Identification of HSMMs 104

4.7.1 General RPE technique 105

4.7.2 Online Identification of HSMMs Using the RPE Method 106

4.7.3 Model Parameterizations 108

4.7.4 Estimation of the Hidden Layer Variables 109

4.7.5 Gradient Vector Update Equations 110

4.7.6 Parameter Update Equations 114

4.7.7 Updating the Hessian Matrix Estimate, R_t 114

4.7.8 Implementation Issues 114

4.8 Effective Bandwidth of a Hidden Semi-Markov Model 116

4.8.1 Effective Bandwidth of Hidden Markov Models 116

4.8.2 Reformulating a Semi-Markov chain as a Markov chain 118

4.8.3 Effective Bandwidth of an HSMM by Reformulating as an HMM 119

4.9 Numerical Results 120

4.9.1 Numerical Results for Synthetic Data 120

4.9.2 Numerical Results for Empirical TV Traffic Traces . . 127

4.10 Chapter Conclusion 137

5 Broadcast Head-End Architecture 138

5.1 Introduction 138

5.2 Multiplexing System Structure of Standard Digital TV System . . . 140
5.3 Multiplexing System for Our Interactive TV System . . . 140
5.3.1 TS packetizer . . . 143
5.3.2 Traffic Shaping Unit . . . 143
5.3.3 Multiplex Buffer . . . 145
5.4 Scheduling . . . 145
5.4.1 Scheduling Unit Objective . . . 145
5.4.2 Prioritizing Policy . . . 146
5.4.3 Limitations Imposed on Scheduling by the Digital TV Standard . . . 146
5.4.4 Scheduling Algorithm . . . 147
5.4.5 Packet Drop-Off . . . 148
5.5 Chapter Conclusion . . . 149

6 Thesis Summary **150**
6.1 Thesis Contributions Summary . . . 152
6.2 Future Research . . . 153

Appendix A Proof of Theorem 1 in Section 2.4 **155**

Appendix B Q-Q Plot **156**

Appendix C Likelihood Ratio Test **158**
C.1 General Likelihood Test . . . 158
C.2 Likelihood Ratio Test for testing the HMM model against HSMM model . . . 159

Bibliography **162**

List of Tables

2.1 Execution time in seconds for calculating $E(t)$ for $1 \leq t \leq 10000$. Simulations were run on a PC with pentium IV processor at 1.7 GHz, using Matlab implementation. 45

2.2 Comparison of our method with the method in [1] for finding the $(\vec{\sigma}, \vec{\rho})$ parameters. $E(t)$ constructed for all $1 \leq t \leq N$. . . . 46

2.3 The model parameters constructed from the entire envelope (second column) and from $E(t)$ for $1 \leq t \leq \tau$ and samples of $E(t)$ at $t = i\delta$ for $1 \leq i \leq n$. $\tau = 200$, $n\delta = N$. 46

2.4 Video sequences used in our study. 48

2.5 Encoding parameters for the video streams used in our study. 48

2.6 Numerical values of $(\vec{\sigma}, \vec{\rho})$ model parameters for the main video sequences used in our simulation. 51

2.7 Simulation parameters. 52

4.1 Results of likelihood ratio test for different typical TV video sequences. The number of states in HSMM and HMM models is $N = 3$. 86

4.2 Actual and initial values of the parameters of HSMM models used in simulating our off-line identification algorithm. 121

4.3 Actual and initial values of the parameters of HSMM models used in simulating our online identification algorithm. 122

4.4 Estimated parameter values of HSMM model for empirical traffic traces of typical TV programs. 129

4.5 Simulation parameters. 133

List of Figures

1.1 Screen shot of an Electronic Program Guide, EPG. EPG lets you search, navigate, and find out what's on your TV, all while watching TV. 2

1.2 Screen shot of an Enhanced TV. Enhanced TV lets you view extra HTML based information about the current TV program, or surf the Internet, while watching TV 3

1.3 A transmission time-line, which illustrates the effect of T_w; the gray boxes show when the data of the i^{th} data unit are actually transmitted. 9

1.4 Structure of the proposed Transmission System. 11

2.1 $E(t)$ for a CBR and a typical VBR source. 21

2.2 Traffic constraint function for different traffic models. 24

2.3 $A^*(t)$ for $(\vec{\sigma}, \vec{\rho})$ model for $m = 1, 2$ and 3. 25

2.4 $E(t)$ for two typical TV programs. The traffic is normalized to its maximum rate, i.e., the maximum rate of these sources is 1. 27

2.5 $E(t)$ obtained from the direct approach, largest subadditive closure extrapolation, repetition extrapolation for $\tau = 400$ and sampling with $\delta = 400$. The source is a VBR MPEG-2 sequence, selected from the motion picture 'Mission Impossible' with 720 × 480 resolution, frame rate 30 and maximum bitrate of 5 Mbps. 30

2.6 Pseudo-code of our algorithm for obtaining the $(\vec{\sigma}, \vec{\rho})$ model parameters from $E(t)$. 34

2.7 Divide and Conquer approach in our algorithm. (a) First step: a single (σ_1, ρ_1) is fitted into the $E(t)$. (b) Second Step: since $A^*(t)$ in step one was not a satisfactory estimate of $E(t)$ for $[T_1, T_2]$, the (σ_2, ρ_2) is fitted to $E(t)$ for $t \in [T_1, T_2]$. (c) Third Step: the (σ_3, ρ_3) is fitted to $E(t)$ for $t \in [T_1, T_2]$. 36
2.8 The General system $\mathcal{S}$. 37
2.9 The arrival and departure functions: (a) Continuous time, where A_{in} and A_{out} are defined for all $t \geq 0$, (b) Discrete time, where A_{in} and A_{out} are only defined at discrete times denoted by dots. 38
2.10 a) The transmitter model with N high priority (main) streams. b) The equivalent model, where the N high priority input streams are replaced with one equivalent stream. 40
2.11 Pseudo-code of our algorithm to find T_w. 42
2.12 $\beta_L(t)$ and $A^*_L(t)$. 43
2.13 Utilization ratio, $U(A^*, d)$. 47
2.14 $A^*(t)$ and $E(t)$ of video sequences' of typical TV programs, a) Mission Impossible, b) News, c) Talk show, d) Documentary. 49
2.15 $A^*(t)$ and $E(t)$ of video sequences' of typical TV programs, a) Court show, b) Muppets, c) Soap Opera, d) Cartoon. 50
2.16 R versus T_w for an incidental stream. T_w is obtained for each R using the admission control algorithm presented in Section 2.4.3. 54

3.1 A sample effective bandwidth curve $\alpha(\theta)$, and $\Lambda(\theta)$, $\Lambda^*(\theta)$ and inverse of the effective bandwidth function $I(c) = \alpha^{-1}(\theta)$. The average rate of the source is .448. 62
3.2 A simple buffer of size B, filled at variable rate y_t and emptied at constant rate c. 64
3.3 A simple multiplexer model with prioritized inputs. 65
3.4 A simple buffer filled at constant rate R and emptied at variable rate c. If the buffer workload is W_L, then we know that the first data unit has waited W_L/R seconds in the buffer. 67
3.5 A simple multiplexer model with N high-priority and one low-priority inputs. 68

4.1 Normalized bitrate and 'State transition path' for two typical TV programs. 85

4.2 a) Histogram of state (or cluster) duration for one of the states in the News Sequence. b) Geometrical probability mass function. c) Discretized Gamma probability distribution function. d) Q-Q plot for Geometrical distribution. e) Q-Q plot for Gamma distribution. 87

4.3 State transition paths that should be considered for computing $\alpha_t(j)$ in the HMM and HSMM cases. 91

4.4 Comparison of the signal generation process in our model with that of existing signal models for HSMMs. 98

4.5 '**E**' and '**M**' steps of our algorithm for off-line identification of HSMMs. 103

4.6 Parameter update equations performed in each step of a general RPE algorithm. 107

4.7 Recursive Prediction Error scheme for estimating HSMM parameters. 108

4.8 a-c) Parameter estimates for the first model versus the iteration number. The dotted lines show the actual value of the parameters. d) The log-likelihood function, $\log(\mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}))$, versus the iteration number. As shown, $\log(\mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}))$ increases in each iteration. 123

4.9 a-c) Parameter estimates for the second model versus the iteration number. The dotted lines show the actual value of the parameters. d) The log-likelihood function, $\log(\mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}))$, versus the iteration number. As shown, $\log(\mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}))$ increases in each iteration. 124

4.10 Online estimation of a 3 state HSMM: a) state transition probability a^o_{12}; b) observation mean for state 1, μ_1; c) state duration mean for state 1, $\mu_{s,1}$. The dotted line shows the actual value of the parameter. The parameter estimate converges to the actual value of the parameter. 125

4.11 Online estimation of a 3 state HSMM, where the actual parameter changes at $t = 5000$: a) state transition probability a^o_{21}; b) observation mean for state 1, μ_1. The dotted lines show the actual value of the parameter. The parameter estimates follow the temporal changes in the actual value of the parameter. . . 126

4.12 Online estimation of HSMM parameters for the 'Talk show' sequence. 130
4.13 Online estimation of HSMM parameters for the Documentary (Best Places in Canada) sequence. 131
4.14 Online estimation of HSMM parameters for the 'Mission Impossible' sequence. 132
4.15 Effective Bandwidth curves of video sequences of typical TV programs. 134
4.16 T_w versus R for an incidental stream that uses the stochastic service class. 135
4.17 Probability of loss p versus rate R, where T_w is constant. The solid lines illustrate the loss probabilities obtained by the admission control scheme, and '+' signs illustrate the actual loss probabilities obtained from simulating the multiplexing system. As shown, actual loss probabilities obtained from simulation are close to what obtained from the admission control. 136

5.1 Conceptual diagram of a Transport Multiplexer. 141
5.2 Diagram of multiplexing system. 142
5.3 Scheduling unit decides which TS packet should be sent at the next transmission time-slot. 146
5.4 Packet drop off in Multiplex buffers of stochastic and best-effort service classes. When a new packet arrives to a full buffer, it is *pushed* into the buffer. 148

B.1 Q-Q plot 157

Acknowledgements

This thesis is the result of five years of work wherein I have been accompanied and supported by many people. It is with pleasure that I now take the opportunity to express my gratitude to all of them.

First and foremost, I would like to thank my supervisors, Dr. Rabab K. Ward, and Dr. Panos Nasiopoulos, who have always been extremely kind and supportive. I would never have been able to finish my thesis without my supervisors' insights, criticisms and thoughtful suggestions. I am most grateful to Rabab because of her continuous moral support, and for showing me the proper research methodology; and I am most grateful to Panos for his academic advice, and for being a mentor to me.

I am thankful to all my colleagues at the 'Image Processing Lab' of the department of Electrical and Computer Engineering at UBC. Especially, I am grateful to the following individuals for their academic help or moral support during my studies, and for being my friends: Alex Paquet, Kemal Ugur, Parvin Mousavi, Vanessa Mirzaee, Yasser Pourmohammadi, Alen Docef, Parvaneh Saeedi, Farhad Ghasemi, and Purang Abolmaesoumi. I am also grateful to Dr. Vikram Krishnamurthy for his comments on one of my research papers, and to Kirsty Barclay for all her editorial help during my studies.

I am grateful to my wife, Paria, for her love and patience, and for her inspirational and moral support during my studies. She is the love of my life, and I dedicate this thesis to her.

Last but certainly not least, I would like to thank my parents, and my sisters, for their continuous love and support. I would not have accomplished this much in my life without their sacrifices for me throughout my life.

Mehran Azimi

The University of British Columbia
MARCH 2004

To my beloved wife, Paria.

Chapter 1

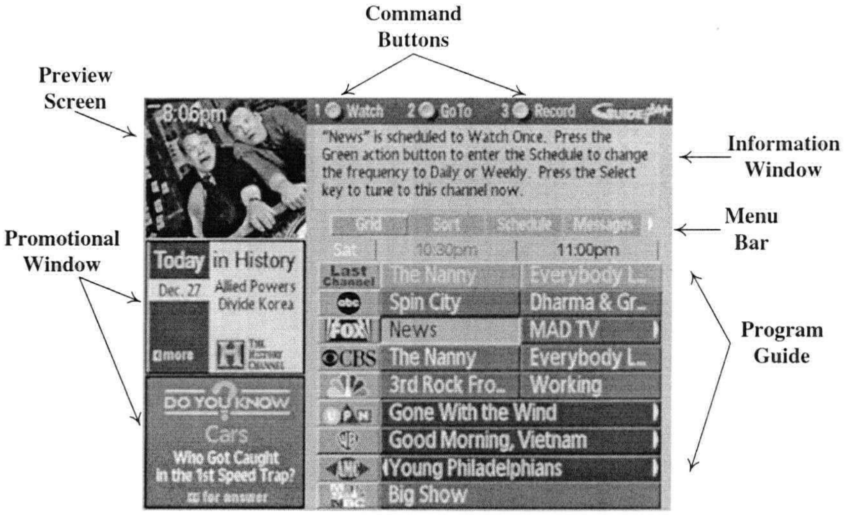
Introduction

Transport of the mails, transport of the human voice, transport of flickering pictures-in this century as in others our highest accomplishments still have the single aim of bringing men together.

- Antoine de Saint-Exupery (1939)

TELEVISION systems are migrating to digital technology. More Digital Television (DTV) systems are being deployed around the world everyday. This change is creating an incredible technological revolution in the entertainment industry. The new DTV technology not only delivers crystal clear picture and superior sound quality, but also allows new services to be added to TV programs. These add-in services enhance the TV viewer's experience by adding extra features or content to TV programs. For example, the presently available Electronic Program Guide (EPG) (see Figure 1.1) is an add-in service for DTV systems. EPG lists the programs that are or will be available on each TV channel, plus a short summary or a commentary for each program.

Interactivity is the most attractive enhancement promised to be added to digital TV systems. However, the concept of 'interactivity' for TV is not clearly defined, and the term 'Interactive TV' has been used for many different TV systems with many different features. For example, Video-On-Demand systems and TV systems with VCR-like functionality are occasionally referred to as 'Interactive TV.' In Video-On-Demand systems, viewers select a movie or TV show from a library, and that movie is played back on their TV. In TV



"News" is scheduled to Watch Once. Press the Green action button to enter the Schedule to change the frequency to Daily or Weekly. Press the Select key to tune to this channel now.

Channel	10:30pm	11:00pm
Last Channel	The Nanny	Everybody L..
abc	Spin City	Dharma & Gr..
FOX	News	MAD TV
CBS	The Nanny	Everybody L..
NBC	3rd Rock Fro..	Working
TNT	Gone With the Wind	
WB	Good Morning, Vietnam	
AMC	Young Philadelphians	
	Big Show	

Figure 1.1: Screen shot of an Electronic Program Guide, EPG. EPG lets you search, navigate, and find out what's on your TV, all while watching TV.

systems with VCR-like functionality, TV viewers can pause or rewind a live TV show, or save a TV show to watch later. This system is commercially available now. Today, 'Interactive TV' mostly refers to the so-called '*Enhanced TV*' system defined by the Advanced Television Enhancement Forum (ATVEF) [2]. ATVEF is an industry alliance of many major companies. The goal of this forum is to standardize HTML enabled TV systems. With the Enhanced TV technology, a web page is displayed alongside a TV picture (see Figure 1.2) on the TV screen. Viewers surf this web page using their remote-controls to get more information about the program, do e-shopping, and so forth. There are two scenarios for sending the web page content. In the first scenario, the web page contents are seamlessly inserted into the broadcasted TV signal. In this scenario, no Internet connection is required. However, the size of the inserted web data is limited, and users are limited to the inserted web pages. In the second scenario, an internet link is inserted into the TV signal. This link is used by the receiver set-top box to download the web page content via an Internet connection. The 'Enhanced TV' system is now commercially available (e.g., Microsoft WebTV©), and many TV shows are currently using this service.

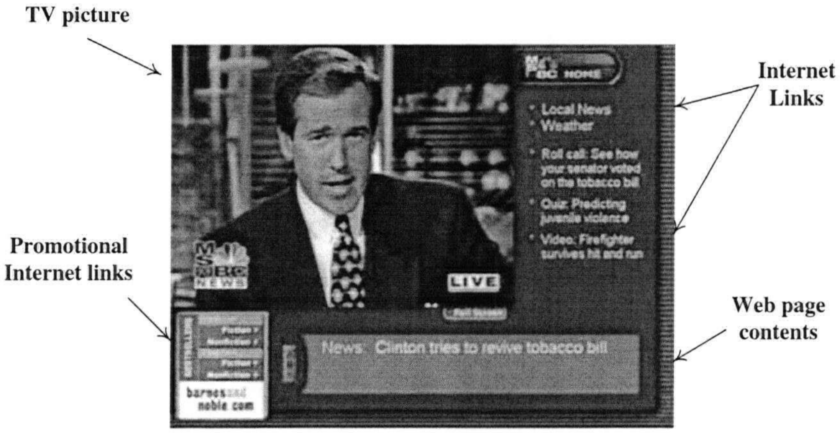

TV picture
Internet Links
Promotional Internet links
Web page contents
MSNBC HOME
Local News
Weather
Roll call: See how your senator voted on the tobacco bill
Quiz: Predicting juvenile violence
Video: Firefighter survives hit and run
MSNBC NEWS
LIVE
News: Clinton tries to revive tobacco bill
barnesandnoble.com

Figure 1.2: Screen shot of an Enhanced TV. Enhanced TV lets you view extra HTML based information about the current TV program, or surf the Internet, while watching TV

Thus, current interactive TV systems are limited to providing World Wide Web content. This content is mostly composed of *text* or *still pictures.* Besides, user interactions in current interactive TV systems are limited to web-surfing-like actions, such as menu selection, typing a URL, and so on. Actually, in practice users interact with the web page content, and not with the TV program content. It is evident that *current interactive systems for digital TV remain limited.* This limitation lies in the fact that none of the current interactive TV systems provide any control over the video or audio content of the TV program.

1.0.1 A New Vision of Interactivity for Television

Our vision of interactivity in Television is a system which allows TV viewers to control the *final presentation* of the TV program *content.* Such interactivity has been successfully implemented in DVD, which is a non-broadcast interactive media. Therefore, this Interactive TV system could be considered as a technology that offers DVD-like interactivity in broadcast digital TV systems. That is, the new Interactive TV system would enable TV viewers to control the final presentation of the video or audio content of a TV program as in a DVD player system. Important applications would include the followings

Multi-lingual audio, where viewers select the language of choice from an available set of languages.

Parental management, where viewers select the content rating of a TV program. If the viewer selects the 'non-adult' version of a program, the receiver then seamlessly replaces the adult rated scenes of the program with a non-adult rated video sequence.

Multi-angle video, where viewers can view a scene from one or more spatially different angles. For example, during a soccer program, viewers can select to view the important scenes of the game from various angles.

Video-in-Video, where viewers may select to open a small window in the corner of their displays. This small window shows a separate video that enhances the viewer's experience. For example, in a soccer program, the small window may display an important incident on the other side of the field.

In the proposed system, viewers use their remote controls as if using a DVD-player remote control, to select choices regarding a TV program from a menu overlayed on top of the main video. This can include, for example, selecting a different language audio track, or switching between adult and non-adult versions of a movie. These choices are conveyed to the receiver and are used to select the appropriate video and audio sequences to be displayed. Hence, the viewer's experience of a TV program is *customized* based on his or her own *choices.*

Thus, our proposed interactive TV system requires the transmission of extra video or audio streams alongside the main TV program stream. We call these extra streams '*incidental streams.*' Incidental streams carry the extra video or audio sequences required for an interactive TV application. For example, the audio stream of another language track in the multilingual audio application, or the video stream of a secondary camera angle in the multi-angle video application are incidental streams.

Depending on the application, an incidental stream may carry a video or audio sequence with a limited and specified length, or an unspecified length. For instance, in the parental management application for a movie, the main stream carries the regular version of the movie, which we assume has a few 'offending' scenes. For each offending scene, a non-offending version of the same scene is transmitted as an incidental stream. In this case, each incidental stream has a limited length, equal to its corresponding scene length. On the other hand, in the case of the multilingual application, the incidental stream carries the audio stream of the other language track for the entire program. In this case, the incidental stream is a continuous stream whose length is equal to the length of the program.

In order to receive and display a digital TV signal, the TV receiver must be equipped with a digital set-top box. A set-top box designed and programmed specifically for the proposed ITV application will be able to display the incidental streams. Conventional set-top boxes simply ignore the incidental streams and only display main streams. This makes the system backwards compatible with the presently available digital TV set-top boxes. Therefore, adding the incidental streams to a TV program does not effect the compatibility of the broadcasted TV signal with conventional digital receivers.

1.1 Challenges

In designing the proposed interactive TV system several issues and challenges must be addressed.

Limited Transmission Bandwidth The channel bandwidth available for transmitting digital TV signal is limited. In most transmission media, such as Cable, Terrestrial and Satellite, a fixed channel is shared among a number of TV programs, where each program uses a fixed share of the transmission bandwidth. In the proposed interactive TV system, incidental streams data must be accommodated in the same transmission channel as the main streams. However, reserving part of the transmission channel bandwidth for incidental streams is not an attractive approach for two reasons. First, the contents of incidental streams are not as important as the main program contents. This is because incidental streams usually carry *enhancement* content for a TV program; hence, incidental streams are expected to be viewed by much smaller TV audiences than the main program content. Therefore, it is not cost-effective to reserve bandwidth for incidental data, which are of secondary importance. Second, depending on the application, incidental streams may carry video or audio clips, each with a limited length. For example, in the parental management application for a movie, an incidental stream carries the non-offending version of the movie *only when there is an offending scene*; otherwise, no incidental stream is broadcasted. Therefore, it is wasteful to reserve a fixed portion of bandwidth for an incidental stream, which may be utilized only during a few time intervals.

Keeping the Quality of Main Streams Intact As noted, main streams carry more important content than incidental streams do. Therefore, *adding incidental streams to a TV program should not degrade the quality of the main streams.*

Compatibility with Standard TV Receivers Adding incidental streams to a TV program should not make the broadcast signal incompatible with conventional TV receivers. Thus, it is necessary that our system design be compliant with the Digital TV standards.

Same Channel Transmission The incidental streams should be accommodated within the same transmission channel used for main streams. It is not an attractive option to use other transmission mechanisms for sending incidental data. For example, consider a possible scenario for implementing the proposed Interactive TV system via adding Internet links to TV programs. These links would point to incidental video and audio streams located on the Internet. Then, the set-top box receivers would use these links to download the incidental streams via a fast internet connection, and display them on the TV screen. However, the problem with this approach is that *each TV receiver* would be required to have a fast internet connection. In fact, this approach has been taken and implemented by a few manufacturers, but it has failed as its concept has been rejected by both the TV broadcasters and consumers.

1.2 Thesis Scope

As noted, the most important challenge in designing the proposed Interactive TV system arises from accommodating the incidental streams within the fixed bandwidth allocated for the main streams in the transmission line of a digital television broadcast system. In this thesis, we address this challenging problem. We present novel solutions for adding incidental streams without introducing any degradation whatsoever in picture quality of main streams, and without increasing transmission bandwidth. This is possible since the rate of main streams varies with time, and does not occupy its entire allocated bandwidth at all times. The incidental streams are transmitted using the available bandwidth. Our method strives to make the most productive use of the available bandwidth, and delivers incidental video and audio content with the best possible picture and sound quality.

Unlike current interactive TV technologies, the proposed system is a *one way* system. That is, no return path from TV viewers to the transmitter or no Internet connection are required. Furthermore, adding incidental streams to a TV program using our method would not affect the compatibility of the broadcast signal with conventional digital TV receivers. In other words, a broadcast signal that carries both incidental and main streams is receivable by both conventional digital TV receivers and by receivers specifically programmed for the proposed ITV application. Conventional receivers will display the main streams, while receivers programmed for the proposed ITV application will be capable of displaying *both incidental and main* streams. These features make the proposed interactive TV system even more attractive to both consumers and TV broadcast companies.

1.3 Our Approach

1.3.1 Framework

We encode the main video sequences with constant picture quality. Therefore, the main video streams are encoded at variable bitrate (VBR). It is well known that simple and slow activity video scenes require a smaller number of encoding bits than complex video scenes do; such that bitrate for complex scenes may reach the maximum allowed bitrate. Digital TV transmission media (e.g., cable or terrestrial) allow a fixed reserved bandwidth for each TV channel equal to the source maximum rate. Therefore, during simple scenes the allowed bandwidth is under-utilized. We propose to use these unused portions of the bandwidth for transmitting incidental stream data.

Each data unit of an incidental stream contains time sensitive data. This means, each data unit should be transmitted before a certain transmission deadline, so that it is available at the receiver at a certain time for decoding and presentation. Incidental stream data units are only transmitted at opportune moments, when the transmission bandwidth is not fully utilized by the main streams. In order to transmit incidental data units at these opportune moments, we propose that the transmitter receives each incidental data unit *ahead* of its transmission deadline, say by t_w seconds. Each data unit is first buffered at the transmitter and then transmitted whenever some free bandwidth becomes available. It is *vital to choose t_w large enough so that the incidental data are transmitted and received by the receiver by the time they are to be decoded and presented to the viewer.* Since decoders may receive the the incidental data units prior to their presentation time, these data have to be buffered at the decoder until their decoding time.

1.3.2 Maximum Waiting Time

An important question arises here: *"for an incidental stream with a given bitrate R, what is the minimum t_w?"* We will denote the minimum t_w by T_w. Therefore, T_w is defined as the maximum time that the data units of an incidental stream with rate R might wait in the transmitter buffer before being transmitted. Once T_w is found, the transmitter should then receive the incidental data units $t_w \geq T_w$ seconds before their transmission deadline. This ensures that all incidental data units are transmitted on time and made available at the decoder prior to or by their decoding time. We discuss our approach to finding T_w in the next section.

A small T_w is extremely desirable for three reasons. First, a small T_w reduces the inescapable delay in starting the presentation of an incidental stream in a live program. Suppose that the first data bit of an incidental stream is delivered to the transmitter system for transmission at time t. If we ignore the constant delays caused by multiplexing, transmission and demultiplexing, then the decoding of this incidental stream can start at the receivers at $t + T_w$. Hence, it is very attractive to have a small T_w, so that playback of incidental streams in live TV programs can start very shortly after they have actually been captured. Second, since the receiver buffers must be capable of storing T_w seconds of an incidental stream data, a smaller T_w then requires a smaller buffer size at the receivers. Third, a small T_w is advantageous when viewers change from one TV channel to another. With a smaller T_w, viewers experience

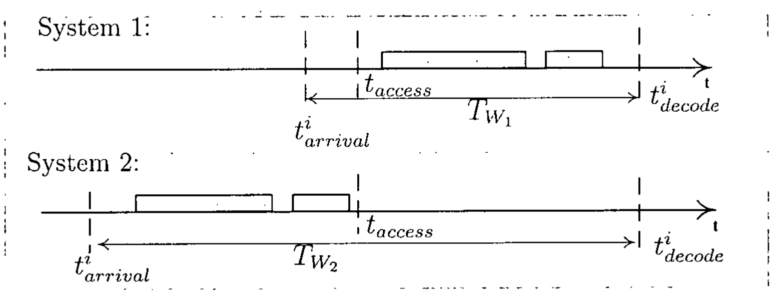


Figure 1.3: A transmission time-line, which illustrates the effect of T_w; the gray boxes show when the data of the i^{th} data unit are actually transmitted.

a shorter 'random access delay' for the incidental streams while they switch channels. By 'random access delay,' we mean the delay TV viewers experience from the moment they switch to a new channel to the time the playback of the new program actually starts. For the main video and audio streams, the random access time rarely becomes more than 0.5 seconds. This is because the coded main video and audio frames are broadcasted very close to their decoding and presentation times. To justify the effect of T_w on random access delay time for incidental streams, consider two transmitters offering two different maximum waiting times, T_{W_1} and T_{W_2}, to an incidental stream of rate R, where $T_{W_1} < T_{W_2}$. Assume the data units of the same incidental stream are sent to these two transmitters. In order to simplify the discussion, we ignore the delays caused by the transmission line and buffering at the receivers. Let t^i_{decode} denote the decoding time of the i^{th} data unit of the incidental stream (see Figure 1.3). Since each data unit arrives at the transmitter buffer T_w seconds before its decoding time, then $t^i_{arrival} = t^i_{decode} - T_W$ denotes the time when the data bits of the i^{th} data unit arrive at the transmitter buffer. The gray boxes in Figure 1.3 show the time instances when the data bits of this data unit are actually transmitted. Now, suppose a TV viewer changes the channel on his or her receiver to this program at t_{access}. In this case, the TV receiver starts receiving the data of this channel at t_{access}. In the first system, the receiver completely receives the i^{th} data unit of the incidental stream, while in the second system, the receiver misses this data unit. Therefore, the presentation of the incidental stream in the first system starts sooner (i.e., from the i^{th} data unit) than in the second system.

1.4 Structure and Mechanisms of the System

In this section, we introduce the mechanisms that we employ in the proposed system. Our objective here is to introduce the concept of each mechanism, and describe their roles. The details of each mechanism and our implementation approach are discussed in detail throughout the thesis.

We refer to the different video or audio streams as *'traffic sources'* and to the actual data as *'traffic'* from here on. This is because each video and audio stream could be considered as a data generating source.

Figure 1.4 illustrates the basic building blocks of the proposed transmission system. As shown, this system consists of the following units: admission control, traffic characterization, scalable coder, service classes, and data multiplexer.

1.4.1 Admission Control

Before an incidental stream is added to a TV program, and actually starts submitting data to the transmitter for transmission, it is necessary to determine the rate R and the waiting-time t_w for this stream. We refer to this mechanism as 'admission control.' The admission control mechanism determines whether or not a certain incidental stream is allowed to be transmitted. The admission control relies on some bandwidth provisioning mechanisms, which forecast the free bandwidth in the system in the future.

The admission control mechanism is initiated by sending a *connection request* from the TV production studio to the transmitter. The connection request is sent in advance of the actual data, and conveys to the transmitter that an incidental stream is going to be added to the program in the near future. The connection request also conveys a set of minimum service parameters for the incidental streams, which are the minimum bitrate R^{min} and the largest waiting-time t_w^{max} that can be selected for this incidental stream. Then, the admission control must determine whether or not it can assign a bitrate R and a waiting-time t_w to the incidental stream, such that $R \geq R^{min}$ and $T_w \leq t_w \leq t_w^{max}$. If the admission control can find such an (R, t_w) parameter pair, then the incidental stream is *accepted* by the admission control. If not, then the incidental stream is *rejected*; that is, the incidental stream will not be added to the program. The above procedure is referred to as the 'admission

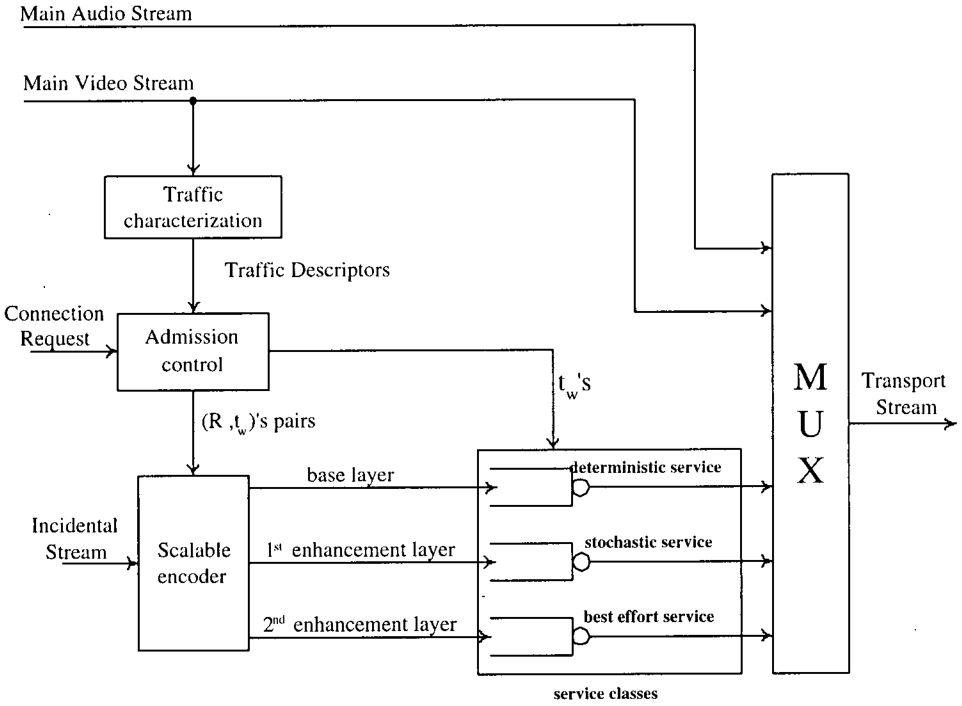


Figure 1.4: Structure of the proposed Transmission System.

test.'

Once the transmitter receives the actual data of an accepted incidental stream, it re-encodes the incidental stream with rate R; and makes the encoded data units available for transmission at exactly t_w seconds before their decoding time. This will be described in more details in the next sections.

1.4.2 Traffic Characterization

The traffic characterization unit assigns a *traffic descriptor* to each main video source, and conveys them to the admission control unit. The traffic descriptor is used in forecasting the bandwidth required by the main streams, from which we can deduce the bandwidth available for the incidental streams. Therefore, the admission control unit uses the traffic descriptors in its bandwidth provisioning mechanism to determine how much bandwidth will be available to the incidental streams. A traffic descriptor is composed of a set of parameters which contain useful and important characteristic information about the traffic shape of the source. Specifically, a traffic descriptor carries information about the traffic 'burstiness' of the sources. A '*traffic burst*' refers to a state where a traffic source generates traffic at a rate higher than its average for a long period of time. We say a traffic source is *bursty* if it frequently generates traffic bursts.

As it will be discussed later, extracting the traffic descriptors directly from traffic is not a straight forward process. For this reason, we employ a modelling approach, where we use a *parameterized model* for modelling the traffic; we then find the traffic descriptors from the model parameters. We refer to these parameterized models as *Traffic Models.* For pre-recorded TV programs, traffic models and traffic descriptors are obtained by using off-line algorithms. For live TV programs, the traffic models and traffic descriptors are obtained by monitoring traffic, and by using online methods.

1.4.3 Service Classes

An incidental stream can be transmitted using one of three different *service classes.* These service classes are defined below.

Deterministic Service Class: When an incidental stream is transmitted using the deterministic service class, the transmitter *guarantees* to send all the incidental data units on time and without any deadline violation or loss. The advantage of this approach is that the incidental stream experiences no data loss; hence the playback of incidental streams at the receivers will have no undesirable visual artifacts such as blocks or picture freezing. The disadvantage of this approach is that T_w (or R) is determined by the admission control process based on the most *pessimistic* bandwidth provisioning for incidental streams. This results in very large T_w (or small R), which is not desirable.

Stochastic Service Class: When an incidental stream is transmitted using the stochastic service class, some of the incidental data units may be dropped (i.e., not transmitted); however, the data loss probability is guaranteed to be less than a certain threshold, say $p\%$. The advantage of this approach is that T_w (or R) is determined using a more relaxed bandwidth provisioning for incidental streams. This results in much smaller

T_w (or larger R) than that of the deterministic service class. However, this approach has the disadvantage that data loss is inescapable, and hence, playback of incidental streams will have some visual discrepancies.

Best Effort Service Class: In the best effort service class, the transmitter does not provide any guarantee of sending the incidental data. As the name 'best effort' implies, the transmitter uses any free bandwidth in the transmission line for sending the incidental data. Since no service guarantee is given, no admission control is necessary for this service class.

As noted, each service class defined above has its own pros and cons. More precisely, if an incidental stream is transmitted with a deterministic service class, then T_w should be selected large enough (or alternately, R is selected small enough) such that we *always* have enough bandwidth to send all the incidental data on time. This means that T_w is selected such that even during the *worst-case conditions*, we find sufficient transmission opportunities for incidental data. This worst-case condition happens when the bandwidth available to incidental streams is at its minimum. In this case, the admission control is performed based on a *pessimistic* bandwidth provisioning. Therefore, the bandwidth provisioning mechanism deviates far from a *usual* state of system. This means that on average, incidental stream data units will wait much less than T_w seconds in the transmitter buffer. This results in poor bandwidth utilization. Conversely, the bandwidth provisioning in the stochastic service class is based on a more relaxed approach. This results in much higher bandwidth utilization. However, data loss is probable with this approach, which in some instances results in unattractive visual discrepancies in TV picture such as green blocks, picture freezing and so on. A solution, which offers a trade-off between visual quality and bandwidth utilization, is discussed below.

1.4.4 Scalable Coding

In order to achieve a compromise between high bandwidth utilization and the smooth playback of incidental streams, we use the *scalable coding* technique for encoding the incidental streams [3–5]. In this technique, a video or audio sequence is encoded to more than one bitstream. The first stream is called the *base layer* stream and usually has a low bitrate. The other bitstreams are called the *enhancement layer* streams, and carry a better picture or sound quality than that of the base layer alone. Unlike the base layer stream, the decoding of enhancement layer streams is not stand-alone, and requires the base layer stream during the decoding process. We propose to send the base layer stream using the deterministic service class, and the first enhancement layer stream using the stochastic service class. The second enhancement layer is transmitted using the best effort service class. This approach guarantees that the base layer stream data are delivered to receivers without any data loss. Therefore, the incidental stream is guaranteed to play back with the minimum quality offered by the base layer stream. Meanwhile, the first enhancement layer data will be transmitted by taking advantage of the bandwidth that is not utilized by the main and base layer incidental streams. Since the data loss is bounded, most of the enhancement layer stream data are expected to be transmitted on time. The second enhancement layer data are transmitted using the best effort service. Therefore, any bandwidth left over by other streams is utilized by the second enhancement layer. This approach results in an incidental stream playback with minimum picture or sound quality determined by the base layer stream, an average quality determined by the first enhancement layer, and a best quality determined by the second enhancement layer.

Therefore, for an incidental stream that is to be re-encoded using scalable coding and transmitted with different service classes, two admission control processes must be performed. The first admission control process determines the rate R_{base} and waiting-time t_w^{base} for the base layer stream. The second admission control process determines the rate R_{enh} and the waiting-time t_w^{enh} at a given data loss rate, say $p\%$, for the first enhancement layer stream. Since no service guarantee is offered in the best effort service class, no admission control process is necessary for the second enhancement layer stream.

1.4.5 Data Multiplexing

The multiplexing unit is responsible for multiplexing the main and incidental streams together. This unit handles the data units of each traffic source according to their priority. The main streams have the highest priority, followed by the incidental streams with deterministic service class. This, in turn, is followed by the incidental streams that use the stochastic service class. The incidental streams with the best effort service class have the lowest priority.

1.5 Thesis Contribution and Structure

This thesis consists of 6 chapters and 3 appendixes. Chapters 2-4 describe how the deterministic and stochastic service classes are designed. This includes the selection of the traffic model, the design of efficient model fitting methods, the selection of a traffic descriptor, and finally the admission control mechanism.

In Chapter 2, we address the admission control problem for the deterministic service class, and develop methods for the deterministic service class. We use the so called $(\vec{\sigma}, \vec{\rho})$ model to model the traffic of main streams, and the so called 'traffic constraint function' as the traffic descriptor. This approach is based on the 'Network Calculus' theory, which studies the deterministic service guarantees in a communication network. We develop efficient algorithms for fitting the $(\vec{\sigma}, \vec{\rho})$ model to a traffic source. These algorithms are useful in any application that employs the $(\vec{\sigma}, \vec{\rho})$ model, and are part of the novel contributions of this chapter. Finally, we design an admission control scheme for the deterministic service class. This admission control scheme is the most important novel contribution of this chapter.

In Chapter 3, we design the stochastic service class. The approach taken is based on the recently introduced theory of 'Effective Bandwidth.' We develop a new physical interpretation of the effective bandwidth concept based on a data buffering model and the large deviation principles concept. Then, we design an efficient algorithm for admission control of the stochastic service class using the effective bandwidth concept. The two important contributions of this chapter are the physical interpretation of the effective bandwidth concept, and the admission control scheme for the stochastic service class.

In Chapter 4, we address the traffic modelling problem for the stochastic service class. We examine different traffic modelling approaches for stochastic modelling of main streams, and select the family of Markovian signal models for modelling the data traffic in the proposed ITV application. We show that the 'Hidden Semi-Markov Models' capture the characteristics of digital TV traffic better than other Markovian models; hence, we employ this model. This line of development is one of the contributions of this chapter. Then, we present a new signal model for hidden semi-Markov models, and present novel methods for parameter estimation of this new signal model for both the off-line and online cases. This new signal model and its parameter identification algorithms are the most important contributions of this chapter, and are useful in other applications which employ hidden semi-Markov models. Finally, we show how the effective bandwidth of a source is obtained from the parameters of a hidden semi-Markov model. This line of development is also a part of contributions of this chapter.

We design a conceptual transmitting system for the proposed interactive TV system in Chapter 5. We discuss the role and importance of 'packet scheduling policy,' and present a scheduling algorithm for multiplexing of main and incidental streams data. Even though this chapter does not include any major contributions, however, it shows how the deterministic and stochastic service class concepts are implemented and integrated together in an actual system.

Finally, in Chapter 6, we present the thesis conclusion. We highlight the thesis contributions, and discuss the future research direction in this field as well.

Chapter 2

Deterministic Service Class

Things which matter most should never be at the mercy of things which matter least.

Goethe

Overview

A scheme needed for the deterministic service class is presented. This includes traffic modelling for main streams, traffic model fitting, traffic descriptor and an admission control scheme. Using the methods presented in this chapter, one can determine the maximum waiting time for an incidental stream with a given rate.

2.1 Introduction

In this chapter, we present the scheme needed for implementing the deterministic service class. For that, we need to forecast the *minimum* amount of bandwidth not occupied by the main streams. The approach taken is based on using a model to forecast the *maximum* data flow of each main video source. This model is referred to as 'traffic model', and the parameters which describe the maximum data flow of the source are referred to as 'traffic descriptors.' These traffic descriptors are then used in the admission control mechanism to obtain the rate and maximum waiting time of incidental streams.

Our approach to this service class studies the transmission system in a *worst-case condition* scenario. In this scenario, all the main sources send the maximum possible traffic to the transmission system for *long periods of time*, thus, during these times the bandwidth available for incidental streams is minimal. Using this minimum bandwidth knowledge, the rate and the maximum waiting time of incidental streams are then determined by the admission control mechanism. This ensures that even in worst-case conditions, there exists sufficient bandwidth to transmit the incidental data.

The traffic descriptor of the main streams is used to specify the *maximum* traffic that the main sources can generate in any time period of length t. This traffic descriptor is known as the 'traffic constraint function' in the literature, and is defined in Section 2.2. This approach also requires that the traffic model captures the *worst-case* burstiness of a traffic source. That is, traffic models which capture the *statistical properties* of traffic are not needed here.

The rest of this chapter is organized as follows. In Section 2.2, we discuss the current deterministic traffic models, and select the most suitable model for our application. The model we employ is called the $(\vec{\sigma}, \vec{\rho})$ model. We also show how the traffic descriptor is obtained from this model. In Section 2.3, we address the issues that arise in fitting the $(\vec{\sigma}, \vec{\rho})$ model to empirical traffic traces, and present efficient solutions for these issues. In Section 2.4, we present an admission control scheme. In Section 2.5, we present numerical results of applying the methods presented in this chapter to actual traffic sources.

2.2 Traffic Characterization

In this section we find a mathematical model for characterizing the traffic of TV video sources (i.e., the main streams). As described earlier, the term 'traffic of a video source' refers to the amount of data bits generated by the video encoder. The mathematical model that we seek should provide a deterministic bound on the amount of traffic a video source generates in any time interval. The important features of a good traffic characterization model are thus 1) accuracy in characterizing the traffic, 2) simplicity in implementation, and 3) ability to capture the useful characteristics of traffic in different time scales. For example, though the peak-to-average ratio of the bitrate of a source can roughly show how the burstiness of the source looks like in a large time-scale, it does not incorporate any information about the burstiness of the source on short time-scales. Therefore, it cannot be used in the design of an efficient admission test.

Although there is a great deal of related work on traffic characterization, much of this work cannot be applied to our problem. Most of these methods characterize the video sources using sophisticated stochastic models, such as Markov models [6–8], autoregressive [9, 10], self-similar [11, 12] and S-BIND [13, 14] models. These approaches are all stochastic and do not provide a deterministic bound on the traffic.

The problem of deterministic characterization of video has been studied for other applications using different approaches [6, 15–18]. These approaches study video bitrate variability at the frame level. Since none of these approaches address the rate variability of full screen video of TV programs at the scene level, the results of these studies are not valid for the proposed ITV application.

In this section, we show how a deterministic traffic characterization is defined. Then we discuss the current parameterized deterministic traffic models, evaluate these models, and select a suitable model for the proposed ITV application.

2.2.1 Deterministic Traffic Characterization

We describe the traffic by means of a cumulative function defined as the amount of data (e.g., number of bits or packets) generated by the source in the time interval $[0, t]$. This functions is called the *cumulative traffic function*, and is defined as

$$A(t) = \int_0^t y(\tau) d\tau \tag{2.1}$$

where $y(\tau)$ is the bitrate of the source at time τ. We use discrete time, where the time parameter t corresponds to an integer number representing the GOP[1] number in the video sequence. Therefore, in our application $y(\tau)$ denotes the number of bits generated by a video source during the τ^{th} GOP (i.e., $y(\tau)$ is the size of the τ^{th} GOP). In this case, 2.1 becomes

$$A(t) = \sum_{\tau=0}^{t} y(\tau), \qquad \forall t \geq 0 \tag{2.2}$$

[1] Group Of Pictures (GOP) in MPEG terminology is the group of frames between two consecutive I frames. The GOP length in most NTSC TV programs is 15. Therefore, with a frame rate of 30 fps, each GOP corresponds to $15/30 = .5$ seconds.

The traffic characterization of a source is obtained by defining the *traffic constraint function* $A^*(t)$, that defines an upper bound on the amount of traffic generated over *any time interval of length t*. Thus

$$A^*(t) \geq A(s+t) - A(s) \qquad \forall s \geq 0 \tag{2.3}$$

Note that the traffic constraint function $A^*(t)$ does not depend on s and hence provides a *time-invariant* bound for the function A. The traffic constraint function is always wide-sense increasing (i.e., $A^*(t) \leq A^*(t+\tau)$ for $\tau > 0$). As discussed in [19] and [20], $A^*(t)$ defines a meaningful constraint only if it is subadditive, which means that $A^*(t+s) \leq A^*(t) + A^*(s)$ for all $s, t \geq 0$. If $A^*(t)$ is not subadditive, it can be replaced by its 'subadditive closure' [21]. The subadditive closure of a function $f(t)$ is the function $f'(t)$ defined with the following recursive equation

$$\begin{aligned} &f'(0) = 0, \\ &f'(t) = \min\left[f(t), \min_{0<s<t}[f'(s) + f'(t-s)]\right], \quad t > 0. \end{aligned} \tag{2.4}$$

Empirical Envelope

The empirical envelope is the tightest traffic constraint function of a source and is defined as

$$E(t) = \max_s\{A(s+t) - A(s)\} \qquad \forall s \geq 0, \forall t \geq 0 \tag{2.5}$$

The empirical envelope indicates the maximum burst size that a source generates in any time interval of length t. The shape of the empirical envelope function carries important information about the burstiness of the source in the worst-case conditions. For example, Figure 2.1 shows $E(t)$ for a constant bitrate (CBR) and a typical variable bitrate (VBR) source. For the CBR source with rate R, the empirical envelope is a linear function of t with slope R, i.e., $E(t) = Rt$. For a VBR source with maximum bitrate R, $E(t)$ is a concave non-decreasing function. For a small t, $E(t)$ of a VBR source is typically very close to Rt.

Traffic characterization of multiplexed sources

Consider an ideal multiplexer with N input video sources. An ideal multiplexer does not delay the incoming traffic and generates a multiplexed stream of all

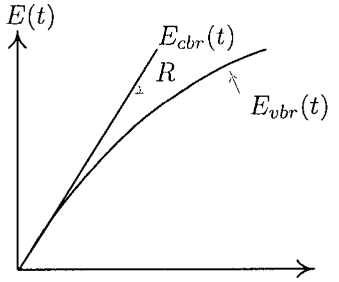


Figure 2.1: $E(t)$ for a CBR and a typical VBR source.

the input streams. The instantaneous rate of the multiplexed stream is the aggregate instantaneous rate of the input streams. If the N input streams are each characterized by the traffic constraint functions $A_i^*(t)$, $i = 1, 2, ..., N$, then the multiplexed stream has the traffic constraint function A_{mux}^* such that

$$A_{mux}^*(t) = \sum_{i=1}^{N} A_i^*(t) \tag{2.6}$$

2.2.2 Parameterized Traffic Characterization Models

In order to use the concept of the traffic constraint function in a practical system, it is necessary to represent the function $A^*(t)$ with a parameterized model. Such a parameterized model would significantly facilitate the design of the admission tests of the incidental streams. Besides, by using a parameterized model, the sources can efficiently specify their traffic characteristics to the system, as only a few parameters need to be conveyed.

As discussed above, the criteria used to evaluate a traffic characterization model are accuracy, simplicity and efficiency of the model in capturing meaningful information about the burstiness of the sources. From the perspective of bandwidth provisioning, the model should be accurate. This means that $A^*(t)$ should be as tight as possible, so that we do not overestimate the traffic of the source. Since the empirical envelope is the tightest bound for the traffic of a source, it is used as a benchmark for the accuracy of a traffic constraint function. While in general a model with more parameters can achieve a more accurate traffic constraint function, the additional parameterizations cause an increase in the complexity of the traffic model. Thus, the selection of an appropriate model must involve a compromise between accuracy and simplicity.

There are five important parameterized deterministic traffic models studied in the literature, known as the peak rate model, the (r, T) model, the the (σ, ρ) model, the $(\vec{\sigma}, \vec{\rho})$ model and D-BIND model.

The peak rate method The peak rate model is the simplest and the most widely used model of all traffic models. In this model, the traffic is described with only one parameter, the peak rate R_{max}. The traffic constraint function for this model is given by $A^*(t) = R_{max} \times t$ for all t. The model is usually enforced for video sources by the rate control section of the source encoder. Note that the peak-rate model is appropriate for specifying CBR traffic, but will overestimate the traffic of VBR sources. This is illustrated in Figure 2.2-a, where the empirical envelope, $E(t)$, and the peak-rate model traffic constraint function, $A^*(t)$, are shown for a VBR source. As shown, the $A^*(t)$ is not a good model for $E(t)$ and overestimates the traffic for long time-intervals (i.e., for large t's).

The (r, T) model The (r, T) model describes the traffic with a rate parameter r and a framing interval T. In this model, time is partitioned into frames of length T and the traffic generated during each frame interval is limited to rT bits. Thus, this model enforces an average rate r, while allowing for moderate bursts. The traffic constraint function for this model is given by: $A^*(t) = (\lceil t/T \rceil + 1)r, \forall t \geq 0$, which is illustrated in Figure 2.2-b.

This model is most suitable for VBR sources with small fluctuations in their bitrate. If a VBR source has large fluctuations in its bitrate, then in order to capture all changes in the traffic a large framing interval T must be selected. However, selecting a large T usually results in overestimation of the traffic.

The (σ, ρ) model The (σ, ρ) model describes the traffic with a burst parameter σ and a rate parameter ρ [15,16]. In this model, the traffic constraint function is $A^*(t) = \sigma + \rho t$. Hence, this model enforces a rate ρ, while allowing some burstiness up to σ. Figure 2.2-c shows the traffic constraint function $A^*(t)$ for this model. Though this model is very simple, it has been successfully used in efficient characterizing of a large class of traffic sources. This model can easily be implemented by using a *leaky bucket* traffic shaper[2]. Because of these attractive features, this model has been widely used in many traffic engineering applications.

The $(\vec{\sigma}, \vec{\rho})$ model The $(\vec{\sigma}, \vec{\rho})$ model is a generalization of the (σ, ρ) model. A $(\vec{\sigma}, \vec{\rho})$ model consists of a set of m (σ_i, ρ_i) pairs, $1 \leq i \leq m$. The traffic is limited by each (σ_i, ρ_i) pair, i.e.,:

$$A^*(t) = \min_i(\sigma_i + \rho_i t), \qquad \forall t \geq 0 \qquad (2.7)$$

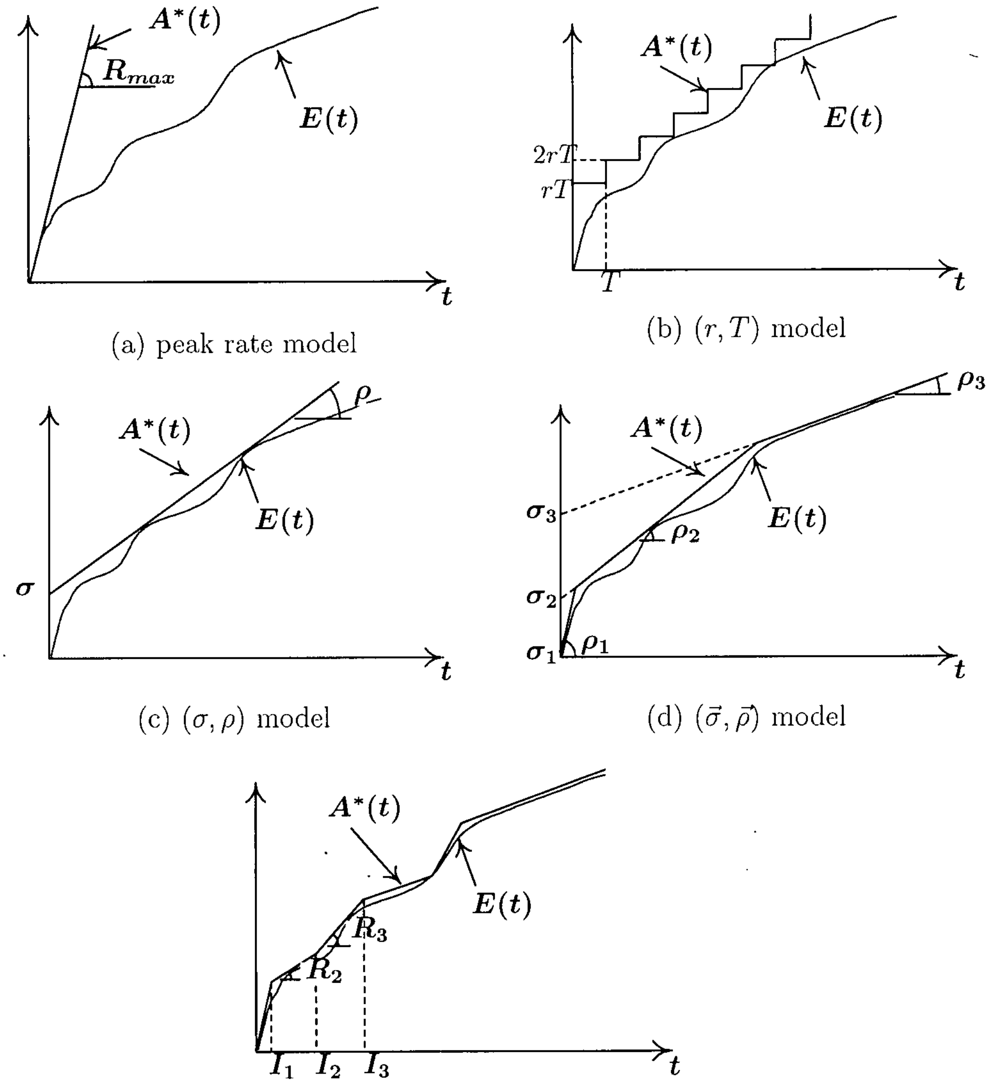
Figure 2.2-d illustrates the $A^*(t)$ function for this model with $m = 3$ piece-wise linear segments. As shown, the traffic constraint function for this model consists of m piecewise-linear segments. By increasing the number of (σ, ρ) pairs m, the model results in a tighter and more accurate constraint function for the traffic. This is illustrated in Figure 2.3, where the $(\vec{\sigma}, \vec{\rho})$ model for a source is plotted for $m = 1, 2$ and 3. As shown by increasing m, the traffic constraint function $A^*(t)$ gets closer to the empirical envelope of the source. However, practical considerations, such as implementation complexity, limit the size of the model, m. Therefore, there is a tradeoff between the accuracy of the model (which usually requires large m) and the simplicity of the model.

The D-BIND model The D-BIND traffic model is a general traffic model that uses a number of rate-interval pairs $\{(R_i, I_i)|i = 1, \ldots, n\}$ [24, 25] The maximum rate over any interval of length I_i is restricted to R_i for all pairs i. The traffic constraint function is given as follows:

$$A^*(t) = \frac{R_i I_i - R_{i-1} I_{i-1}}{I_i - I_{i-1}}(t - I_i) + R_i I_i \quad \text{for all } I_{i-1} \leq t \leq I_i \qquad (2.8)$$

The traffic constraint function of the D-BIND model thus consists of n piece-wise linear segments as shown in Figure 2.2-e. Note that the $(\vec{\sigma}, \vec{\rho})$ model can be viewed as a special case of the D-BIND model since the $(\vec{\sigma}, \vec{\rho})$ model defines an n segment *concave* piece-wise linear constraint function. It should be noted that the traffic constraint function of the D-BIND model in some instances may not be subadditive.

[2] Efficient implementation of the leaky bucket mechanism is discussed in [22, 23].



(a) peak rate model

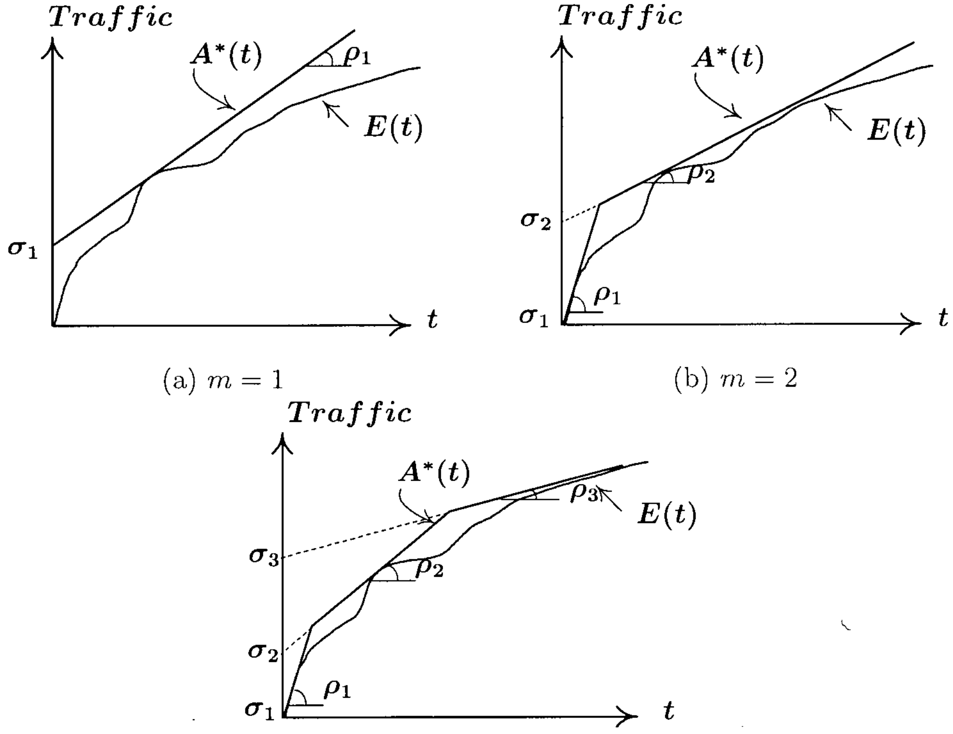
(b) (r, T) model

(c) (σ, ρ) model

(d) $(\vec{\sigma}, \vec{\rho})$ model

(e) D-BIND model

Figure 2.2: Traffic constraint function for different traffic models.



(a) $m = 1$

(b) $m = 2$

(c) $m = 3$

Figure 2.3: $A^*(t)$ for $(\vec{\sigma}, \vec{\rho})$ model for $m = 1, 2$ and 3.

Several studies have evaluated these deterministic traffic models for modelling the video traffic[3] with respect to accuracy and simplicity criteria (see [26, 27] and [17]). In these studies, the simplicity of the models are evaluated based on the complexity of implementation of the admission control tests and the traffic monitoring and policing[4] mechanisms. Meanwhile, it is shown in [27] how the parameters of each model can be expressed in terms of the other models. This enables a direct comparison of these models. All

[3] In all of these studies, the traffic video source is considered at frame level or ATM cell level.

[4] In *traffic monitoring*, the traffic of the source is monitored in real time to make sure that it complies with its traffic characterization model. If the real traffic does not comply with the model, the traffic shape is enforced to follow the model by a mechanism called *traffic policing.*

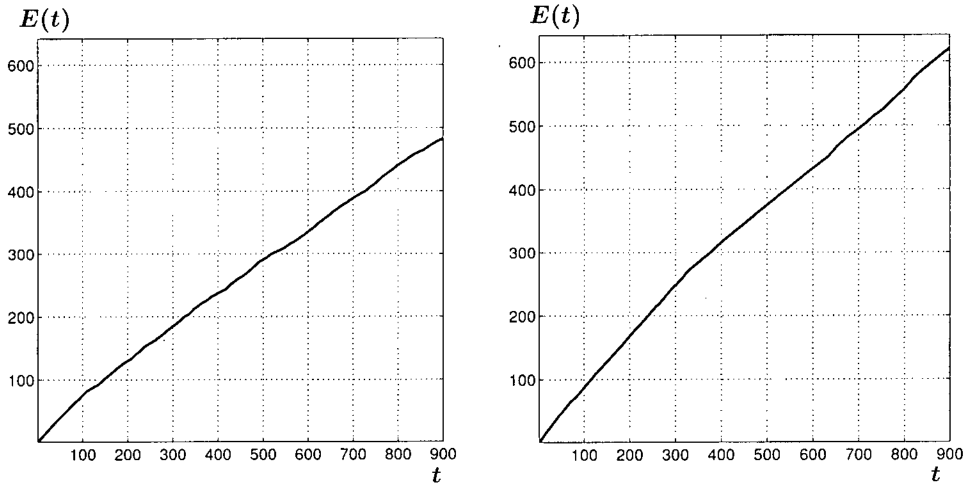
these studies indicate that the (σ, ρ) model is superior to the peak-rate and (r, T) models. However, they also show that the use of a single (σ, ρ) model cannot usually achieve an acceptable accuracy for most of the applications. It is shown in [17] that the $(\vec{\sigma}, \vec{\rho})$ model which employs multiple (σ, ρ) models can accurately characterize the VBR video.

We employ the $(\vec{\sigma}, \vec{\rho})$ model, as it is known to be simple and accurate for modelling VBR video [17, 26, 27]. In order to evaluate the $(\vec{\sigma}, \vec{\rho})$ model with respect to its ability to capture useful characteristics of traffic for our application, we studied the empirical envelopes of several typical TV programs. Figure 2.4 shows $E(t)$ for two typical TV programs, a movie and a news program. The results show that $E(t)$ is a concave increasing function, with two expected characteristics. 1) For very small t's, $E(t)$ is almost linear, that is $E(t) \approx R_{max}t$, where R_{max} is the maximum rate of the source. 2) For very large t's, $E(t)$ is also almost linear with the slope $dE(t)/dt \approx R_{avg}$, where R_{avg} is the average rate of the source. For other t's, $E(t)$ is a concave decreasing function. In order to capture these two important characteristics of the $E(t)$, we select the first (σ, ρ) pair of the model as $\sigma_1 = 0$ and $\rho_1 = R_{max}$. In addition, the rate parameter of the last (σ, ρ) pair is set to $\rho_m = R_{avg}$. This selection captures the two important characteristics of $E(t)$ in our model. The other (σ, ρ) pairs' parameters should be found so that $A^*(t)$ models the concave section of $E(t)$.

2.3 Fitting the $(\vec{\sigma}, \vec{\rho})$ Model to a Source

We here address how to construct an *accurate* $(\vec{\sigma}, \vec{\rho})$ model for a traffic source, specifically a video source, using *a few* (σ, ρ) pairs and a *reasonable* amount of *computational effort*. Finding an efficient and practical way of constructing a model, which offers the right trade off between accuracy, size and computational effort, is a real challenge. Such a model should be accurate in order to achieve high bandwidth utilization, and should include as few (σ, ρ) pairs as possible, so that it can be used in practical admission control schemes [27]. Computation time is very important for *online* traffic sources, where the $(\vec{\sigma}, \vec{\rho})$ model should be constructed by monitoring samples from an online traffic source.

There are many methods for selecting the $(\vec{\sigma}, \vec{\rho})$ model parameters whose design objective is not to strive for high bandwidth utilization, but



(a) News (b) "Mission Impossible" movie

Figure 2.4: $E(t)$ for two typical TV programs. The traffic is normalized to its maximum rate, i.e., the maximum rate of these sources is 1.

instead to select the traffic parameters according to bandwidth availability. These methods include Dual leaky bucket, fixed burst [28], concave hull [17], product [29] and maximum distance [17].

Characterization of a VBR traffic source with the goal of achieving high bandwidth utilization and use of the $(\vec{\sigma}, \vec{\rho})$ model is studied in [1,17,30]. The benchmark in these methods for evaluating the accuracy of a traffic source is the empirical envelope of the source. Since $E(t)$ is the most accurate traffic constraint function for a source, the approach of these methods is to first construct $E(t)$, and then construct $A^*(t)$ as an approximation of $E(t)$.

However, construction of $E(t)$ for all t is extremely time consuming and is not practical in real time applications. Existing methods use extrapolation techniques to reduce the computational load in finding $E(t)$ for large time intervals (i.e., large t's), a process that in turn results in rough estimates of $E(t)$. Moreover, finding the model parameters from the empirical envelope is also a challenge. Existing methods are based on a 'brute force' search approach, which is extremely time consuming.

In this section, we present a new algorithm for constructing the empirical envelope of a source. We show that our method results in a better approximation of $E(t)$, when compared to existing methods. Moreover, due to its speed and iterative design, our method can easily be employed for on-line traffic sources, where the source traffic is not known a-priori and the speed of the traffic characterization algorithm is important.

We also present a unique and robust algorithm for obtaining the optimum model parameters from $E(t)$. Since $E(t)$ is the most accurate constraint function for a source, we select the $(\vec{\sigma}, \vec{\rho})$ model parameters so that $A^*(t) \geq E(t)$ for all t, where $A^*(t) = min_i(\sigma_i + \rho_i t)$ and $A^*(t)$ is *as close to* $E(t)$ *as possible.* We use the 'divide-and-conquer' approach in our algorithm and set up the problem such that a powerful optimization method, called Sequential Quadratic Programming, can be applied to the problem. Unlike other methods in the literature, our method finds the model parameters directly from the true or sub-sampled $E(t)$. In addition, our method is faster and more robust than the current methods and results in a near optimum model.

In Section 2.3.1, we review previous works related to fitting a $(\vec{\sigma}, \vec{\rho})$ model to a traffic source. We present our method for constructing the empirical envelope in Section 2.3.2. In Section 2.3.3, we present our method for obtaining the $(\vec{\sigma}, \vec{\rho})$ model parameters from the empirical envelope.

2.3.1 Overview of Current Methods for Fitting the $(\vec{\sigma}, \vec{\rho})$ Model to a Traffic Source

As mentioned above, all current methods are based on the two counterparts. First, constructing the empirical envelope, $E(t)$, and second, finding the $(\vec{\sigma}, \vec{\rho})$ parameters from $E(t)$. We now discuss current methods for each part.

Constructing $E(t)$ from the traffic

In [17], the empirical envelope $E(t)$ is obtained by running an exhaustive search and finding the maximum burst size in the entire stream. More precisely, if the instantaneous traffic rate of a source is given by $y(i)$ and the total length of the source is N, then $E(t)$ is constructed by calculating:

$$E(t) = \max_{1 \leq k \leq N-t+1} \sum_{i=k}^{k+t-1} y(i) \tag{2.9}$$

The drawback of this approach is its extensive computational complexity. In order to compute $E(t)$ for $1 \leq t \leq N$, $\mathcal{O}(N^2)$ operations is required. In addition, N is generally very large, specially for video traffic sources (e.g., $N \approx 10^5$ for a 2 hour movie when the traffic is considered at the frame level). Hence, for the majority of video sources, it is not practical to compute $E(t)$ using equation 2.9. There are methods in the literature that strive to reduce this computational complexity [1,30]. In [1,30] $E(t)$ is *approximated* using extrapolation. That is, $E(t)$ for $1 \leq t \leq \tau$ is computed using equation 2.9, and $E(t)$ for $t > \tau$, denoted by $E_\tau(t)$, is extrapolated from $E(t)$, $1 \leq t \leq \tau$. The extrapolation method used is either the "largest subadditive closure" with the computational complexity of $\mathcal{O}(N^2)$ (which is the same complexity if $E(t)$ was constructed using equation 2.9), where

$$E_\tau(t) = \min_{1 \leq k \leq t} \{E(k) + E(t-k)\} \qquad \text{for } t > \tau \tag{2.10}$$

or the "repetition extrapolation" with the computational complexity of $\mathcal{O}(\tau N)$, where $E(t)$ for $t > \tau$ is obtained by simply repeating the first τ values in the envelope (i.e., $E(t)$, $1 \leq t \leq \tau$)

$$E_\tau(t) = \lfloor \frac{t}{\tau} \rfloor E(\tau) + E(t - \lfloor \frac{t}{\tau} \rfloor \tau) \qquad \text{for } t > \tau \tag{2.11}$$

The parameter τ is experimentally selected for each application. The disadvantage of this extrapolation approach is that it results in high utilization *only* for the small maximum waiting times T_w. For large T_w's, the traffic characterization based on $E_\tau(t)$ results in a poor network utilization. This is because for large t's, $E_\tau(t)$ is not a good approximation of $E(t)$. This is illustrated in Figure 2.5, where $E(t)$ and $E_\tau(t)$ for a typical video source are shown. We observe that $E_\tau(t)$ is not a good approximation for $E(t)$ for large t's.

Finding the model parameters from $E(t)$

Once $E(t)$ (or $E_\tau(t)$) is found, $A^*(t)$ is constructed as a piece-wise linear approximation of $E(t)$. The current approach includes two steps. First, since $E(t)$ is not necessarily concave and sub-additive, it is replaced with the concave hull of $E(t)$, denoted by $\mathcal{H}(E)$ [17,18]. $\mathcal{H}(E)$ is the smallest piece-wise linear concave function larger than E [31]. Theoretically, $\mathcal{H}(E)$ can be used as A^*. However, the number of piece-wise linear segments in $\mathcal{H}(E)$ is *usually very large*, resulting in an impractical large model size. For this reason, $\mathcal{H}(E)$ is replaced with another piece-wise linear function that has only a few linear segments. The idea behind this approach is to use a cost function to measure the

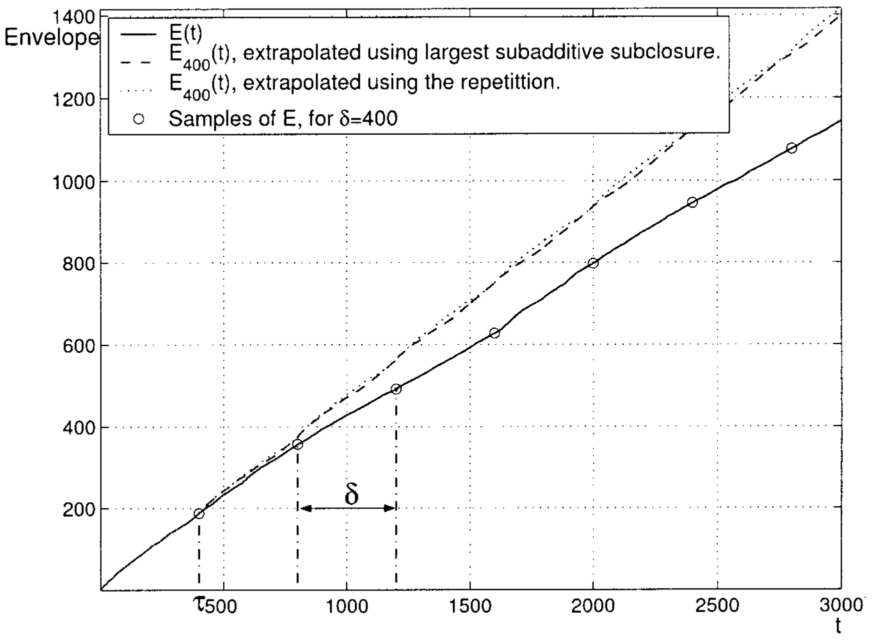


Figure 2.5: $E(t)$ obtained from the direct approach, largest subadditive closure extrapolation, repetition extrapolation for $\tau = 400$ and sampling with $\delta = 400$. The source is a VBR MPEG-2 sequence, selected from the motion picture 'Mission Impossible' with 720×480 resolution, frame rate 30 and maximum bitrate of 5 Mbps.

difference between $\mathcal{H}(E)$ and the traffic constraint function of the new model (say A^*_n).where the model size n is small enough. Then an algorithm with a *heuristic* approach is used to find the (σ, ρ) parameters of the A^*_n model. In this algorithm, each (σ_i, ρ_i) pair is updated in each iteration through *an exhaustive search* through all the possible values for each (σ_i, ρ_i) pair to find the one that minimizes the cost function (see [1,30]). The major drawbacks of this method are: 1) the computation of the convex hull of E and the heuristic approach to reduce the model size are both computationally expensive, 2) the heuristic method to reduce the model size is not always guaranteed to converge to a result, and 3) this method might converge to the local maximum of the cost function, thus it does not guarantee that the optimum parameters are found.

2.3.2 Our Approach to Constructing the Empirical Envelope

Here, we present a method that finds the exact values of $E(t)$ for all $1 \leq t \leq \tau$. For $t > \tau$, we find $E(t)$ at equally δ spaced samples $t = \delta, 2\delta, ..., n\delta$ (see Figure 2.5). The sampling interval δ is a positive integer. If $\delta < \tau$, then the samples at $t = i\delta$ for which $i\delta < \tau$ are repeating and need not to be computed again. For simplicity, we present our algorithm for constructing $E(t)$ for $t < \tau$ and for samples of $E(t)$ separately in sections 2.3.2 and 2.3.2, respectively. The number of operations to construct $E(t)$ for $1 < t < \tau$ and the equally spaced samples of $E(t)$ are $\mathcal{O}(\tau N)$ and $\mathcal{O}(nN)$ respectively.

Construction of $E(t)$ for $t < \tau$

Assume that the total number of bits generated by the source, $y(i)$ for $i = 1, 2, \ldots, N$ are given, where N is the total length of the source. In our case, N is the total number of GOP's in the whole video sequence. We like to find $E(t)$ for $t = 1, 2, \ldots, \tau$, where $\tau \leq N$. $E(t)$ is given by [17]

$$E(t) = \max_{1 \leq k \leq N-t+1} \sum_{i=k}^{k+t-1} y(i) \tag{2.12}$$

We define vector $\mathbf{s}_k$ of size $1 \times \tau$ as

$$\mathbf{s}_k = \left[y(k) \quad \cdots \quad \sum_{i=k-1}^{k} y(i) \quad \sum_{i=k-\tau+1}^{k} y(i) \right]^T \tag{2.13}$$

The objective is to construct $\mathbf{s}_k$ for $k = 1, 2, \ldots, N$. The empirical envelope $E(i)$ is computed as the $\max_k(\mathbf{s}_k(i))$. $\mathbf{s}_k$ is easily constructed from $\mathbf{s}_{k-1}$ by shifting elements of $\mathbf{s}_{k-1}$ down and adding $y(k)$ to the result. Our algorithm consists of the following steps:

1. Let $k = 1$ and initialize $\mathbf{s}_1$ and e as follows

$$\mathbf{s}_1 = \begin{bmatrix} y(1) & 0 & \cdots & 0 \end{bmatrix}^T_{\tau \times 1} \tag{2.14}$$

$$e = \mathbf{s}_1 \tag{2.15}$$

2. Let $k = k + 1$. Find $\mathbf{s}_k$ and e using

$$\begin{aligned} \mathbf{s}_k &= (\text{shift elements of } \mathbf{s}_{k-1} \text{ down by one}) \\ &\quad + \begin{bmatrix} y(k) & 0 & \cdots & 0 \end{bmatrix}^T \qquad (2.16) \\ e &= max(e, \mathbf{s}_k) \qquad (2.17) \end{aligned}$$

3. Repeat step 2 until the last input is reached, i.e, $k = N$.

When the algorithm finishes, e is the empirical envelope of the source $e(i) = E(i)$ for $1 \leq i \leq \tau$.

The computational complexity of our algorithm is only $\mathcal{O}(\tau N)$, which is considerably less than $\mathcal{O}(\tau N^2)$ for a brute force approach using equation 2.12. Due to the iterative structure of our method, it can easily be adopted in on-line applications, where the source traffic $y(i)$ is not known a priori.

Construction of samples of $E(t)$ for $t = \delta, 2\delta, \ldots, n\delta$

Given the traffic source $y(i)$, we like to compute $E(t)$ for $t = \delta, 2\delta, \ldots, n\delta$, where n is the number of samples to be computed. Let

$$\mathbf{\Delta}_k(i) = \sum_{l=k-i\delta+1}^{k} y(l), \quad 1 \leq k \leq N,\ 1 \leq i \leq n \qquad (2.18)$$

Then using equation 2.12, we have $E(i\delta) = \max_k \mathbf{\Delta}_k(i)$. Our goal here is to construct $\mathbf{\Delta}_k(i)$ in an efficient way rather than computing $\sum_{l=k-i\delta+1}^{k} y(l)$ for all $1 \leq k \leq N$ and $1 \leq i \leq n$. Our algorithm iteratively constructs $\mathbf{\Delta}_k$, $k = 1, 2, \ldots, N$ and $1 \leq i \leq n$. The key idea of our algorithm is that for each new k, we efficiently *re-use some pre-computed* terms to construct $\mathbf{\Delta}_k(i)$. By doing so, our algorithm reduces drastically the number of operations required to construct each $\mathbf{\Delta}_k(i)$.

First, we define the vector $\mathbf{\Delta}_k$ of size $n \times 1$, where $\mathbf{\Delta}_k(i)$ is the sum of $i\delta$ consecutive input ending by $y(k)$ as defined in equation 2.18. It can be easily shown that

$$\begin{aligned} \mathbf{\Delta}_k(i) &= \mathbf{\Delta}_k(i-1) + \sum_{l=k-i\delta+1}^{k-(i-1)\delta} y(l) \\ &= \mathbf{\Delta}_k(i-1) + \mathbf{\Delta}_{k-(i-1)\delta}(1) \qquad \text{for } i > 1 \qquad (2.19) \end{aligned}$$

Our algorithm relies on equation 2.19 to iteratively construct $\mathbf{\Delta}_k$ for $k = 1, 2, \ldots, N$. Equation 2.19 requires $\mathbf{\Delta}_k(1)$, which is already computed in previous iterations. Hence, we save the first element of $\mathbf{\Delta}_k$ in each iteration for future use. For this purpose, we use a vector $\mathbf{A}$ of size $((n-1)\delta + 1) \times 1$, where $\mathbf{\Delta}_k(1)$ is pushed into $\mathbf{A}$ in each iteration. In k^{th} iteration, we have $\mathbf{A}(i) = \mathbf{\Delta}_{k-i+1}(1)$ and equation 2.19 becomes

$$\mathbf{\Delta}_k(i) = \mathbf{\Delta}_k(i-1) + \mathbf{A}((i-1)\delta + 1) \qquad \text{for } i > 1 \tag{2.20}$$

The empirical envelope samples, i.e., $E(t)$ at $t = i\delta$, are easily obtained as $\max_k \mathbf{\Delta}_k(i)$. The algorithm is summarized as follows.

1. Let $k = 1$. Initialize $n \times 1$ vector $\mathbf{\Delta}$, $n \times 1$ vector E_δ and $((n-1)\delta + 1) \times 1$ vector $\mathbf{A}$ as :

$$E_\delta = \mathbf{\Delta} = \begin{bmatrix} y(1) & y(1) & \ldots & y(1) \end{bmatrix}^T \tag{2.21}$$

$$\mathbf{A}(i) = \begin{cases} \mathbf{\Delta}(1) & \text{for } i = 1 \\ 0 & \text{for } 1 < i \leq (n-1)\delta + 1 \end{cases} \tag{2.22}$$

2. Let $k = k + 1$. Update $\mathbf{\Delta}(1)$ as

$$\mathbf{\Delta}(1) = \mathbf{\Delta}(1) + y(k) - y(k - \delta) \tag{2.23}$$

Note that if this algorithm is executed in parallel with the algorithm presented in 2.3.2 and $\delta < \tau$, then we have $\mathbf{\Delta}(1) = \mathrm{s}_k(\delta)$. Hence, this step of the algorithm can be ignored.

3. Update vector $\mathbf{A}$

$$\mathbf{A}(i) = \begin{cases} \mathbf{\Delta}(1) & \text{for } i = 1 \\ \mathbf{A}(i-1) & \text{for } 1 < i < (n-1)\delta + 1 \end{cases} \tag{2.24}$$

4. Update $\mathbf{\Delta}(i)$ for $i > 1$ using equation 2.20.

5. Update $E_\delta = \max(E_\delta, \mathbf{\Delta})$.

Steps 2 to 5 of the algorithm are repeated for $k = 2, 3, \ldots, N$. When the algorithm finishes, we have $E(i\delta) = E_\delta(i)$, $1 \leq i \leq n$.

Algorithm 1 Find the $(\vec{\sigma}, \vec{\rho})$ model parameters.

INPUTS: $E(t)$ for $t = 1, 2, \ldots, N$; a criteria to end the algorithm (i.e., the size of the $(\vec{\sigma}, \vec{\rho})$ model n, or the maximum acceptable error in the model).

OUTPUT: σ_i and ρ_i for $1 \leq i \leq n$.

```
Fit a single $(\sigma, \rho)$ to $E(t)$ for $t = 1, 2, \ldots, N$
$A^*(t) = \sigma + \rho t$
while $A^*(t)$ does not satisfy the accuracy criteria or the model has
not reached its maximum size do
    Find $[T_1, T_2]$ such that for all $t \in [T_1, T_2]$, $A^*(t)$ overestimates
    $E(t)$ more than a threshold
    Fit a single $(\sigma, \rho)$ to $E(t)$ for $t \in [T1, T2]$
    Add the new $(\sigma, \rho)$ to the model
    $A^*(t) = \min_i(\sigma_i + \rho_i t)$
end while
```

Figure 2.6: Pseudo-code of our algorithm for obtaining the $(\vec{\sigma}, \vec{\rho})$ model parameters from $E(t)$.

2.3.3 Obtaining the $(\vec{\sigma}, \vec{\rho})$ Parameters from the Empirical Envelope

Our method for finding the $(\vec{\sigma}, \vec{\rho})$ model parameters from $E(t)$ follows a *divide-and-conquer* approach [31]. In this approach, the problem is broken into subproblems which are similar to the original problem but smaller in size, the subproblems are solved, and then the results are combined to create the solution to the original problem. Following this technique, we divide our problem to the subproblems of fitting a single (σ, ρ) to the $E(t)$ for a specific range of t, let us say $t \in [T_1, T_2]$. This subproblem has a smaller size than the original problem and is easier to solve. In Section 2.3.3 we describe how we solve this subproblem, and how all the results are combined to obtain the final $(\vec{\sigma}, \vec{\rho})$.

Suppose $E(t)$, for $t = 1, 2, \ldots, N$, is given. Our algorithm first fits a single (σ, ρ) model to the whole input data. That is, we find σ and ρ so that $\sigma + \rho t \geq E(t)$, and $\sigma + \rho t$ is a good approximation for $E(t)$, for $t = 1, 2, \ldots, N$. Then we proceed by reducing our problem to a subproblem of smaller size. For this, we first select the interval $[T_1, T_2]$ such that $A^*(t)$ is not a satisfactory estimate for $E(t)$ for all $t \in [T_1, T_2]$. For example, we find a $[T_1, T_2]$ such that

$(A^*(t) - E(t))/E(t)$ is greater than a threshold for all $t \in [T_1, T_2]$. If there are more than one such interval, we select the one that $A^*(t)$ is the worst estimate for $E(t)$. The same approach is then used, and a single (σ, ρ) is fitted to $E(t)$ for $t \in [T_1, T_2]$. This will add a new (σ, ρ) pair to our model. This procedure of adding a single (σ, ρ) pair to the model is repeated until a criteria for ending the algorithm is met. This criteria depends on the application and is either the maximum number of (σ, ρ) pairs in the model or an accuracy criteria. For example, in some applications the practical considerations may require that the model size does not exceed a certain size, say n. In this case, the algorithm ends when n (σ, ρ) pairs are added to the constructed model. On the other hand, some applications may require that a certain level of accuracy is preserved in the model, e.g., $A^*(t)$ does not overestimate $E(t)$ more than a threshold, say $p\%$. In this case, we add (σ, ρ) pairs to the model until $A^*(t)$ satisfies this accuracy criteria.

Our sub-problem: fitting a single (σ, ρ) to $E(t)$

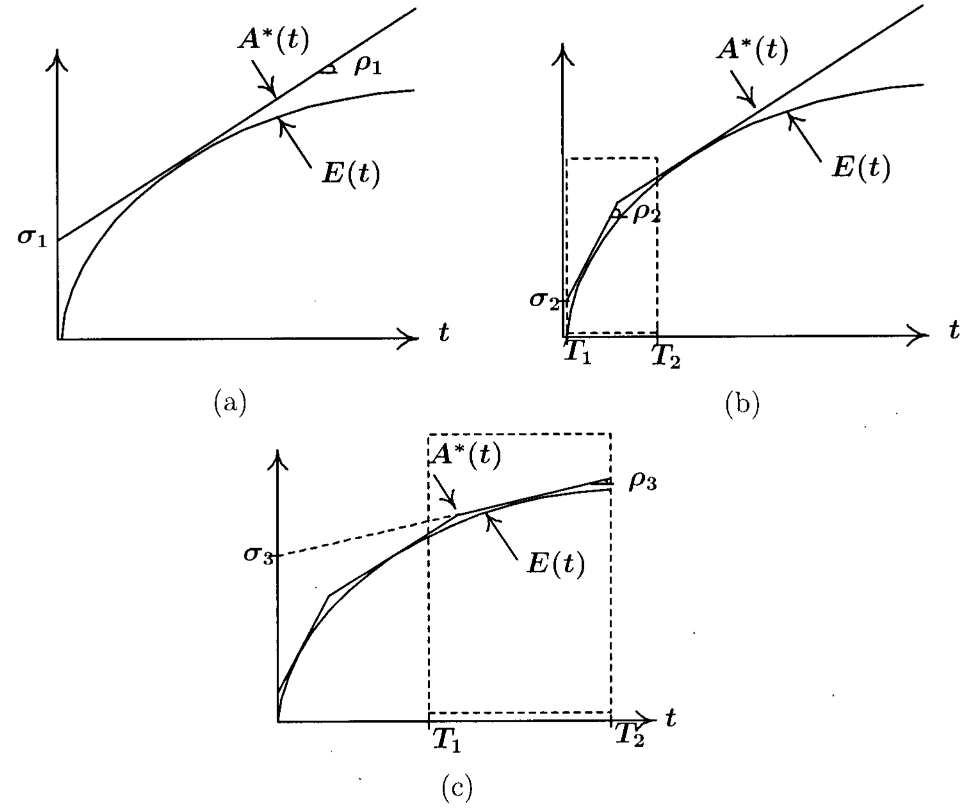
In each iteration of our algorithm, we need to solve the sub-problem of fitting a single (σ, ρ) to a part of $E(t)$. That is, given $E(t)$, T_1, and T_2, we should find σ and ρ such that: 1) $\sigma + \rho t \geq E(t)$ for all $1 \leq t \leq N$; this constraint ensures that the constructed model is *concave* and does not underestimate $E(t)$ for any t, and 2) $\sigma + \rho t$ is an optimum approximation of $E(t)$ for $t \in [T_1, T_2]$. In order to measure the closeness between $A^*(t)$ and $E(t)$ for $t \in [T_1, T_2]$ we use the error function defined as:

$$\begin{aligned} error(\sigma, \rho) &= \textstyle\sum_{t=T_1}^{T_2} (\sigma + \rho t - E(t)) \\ &= (T_2 - T_1 + 1)(\sigma + \tfrac{(T_2+T_1)}{2}\rho) - \textstyle\sum_{t=T_1}^{T_2} E(t) \end{aligned} \tag{2.25}$$

The terms $\sum_{t=T_1}^{T_2} E(t)$ and $T_2 - T_1 + 1$ do not depend on σ or ρ. Hence, we only need to minimize the function

$$error(\sigma, \rho) = \sigma + \frac{(T_2 + T_1)}{2}\rho \tag{2.26}$$

with the constraint $\sigma + \rho t \geq E(t)$ for all $1 \leq t \leq N$. This is a classic optimization problem and there are many standard approaches available in literature to solve such a problem [32]. We choose to employ the *Sequential Quadratic Programming (SQP)* method to solve this problem in our application. SQP



(a)

(b)

(c)

Figure 2.7: Divide and Conquer approach in our algorithm. (a) First step: a single (σ_1, ρ_1) is fitted into the $E(t)$. (b) Second Step: since $A^*(t)$ in step one was not a satisfactory estimate of $E(t)$ for $[T_1, T_2]$, the (σ_2, ρ_2) is fitted to $E(t)$ for $t \in [T_1, T_2]$. (c) Third Step: the (σ_3, ρ_3) is fitted to $E(t)$ for $t \in [T_1, T_2]$.

is a robust and state of the art technique for solving optimization problems. For implementation details of this technique see [32,33]. Using the SQP technique, we easily minimize the function defined in equation 2.26, and find the optimum σ and ρ to our problem within a few iterations.

Note that there are other approaches to finding the parameters of a *single* (σ, ρ) model like the 'product method' [29], 'maximum distance' [34] and 'fixed burst' [28]. These methods use different optimality criterion for selecting the parameters. We should point out that instead of using SQP, any of the above mentioned approaches can be used to solve the subproblem of

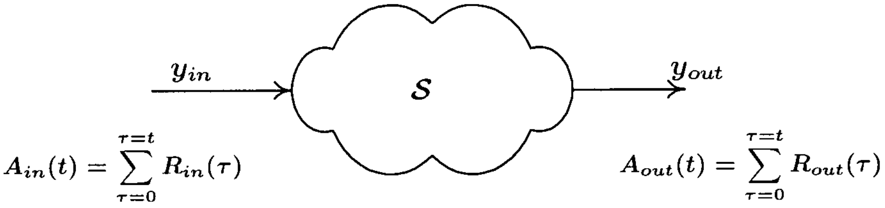

y_{in}
$\mathcal{S}$
y_{out}
$A_{in}(t) = \sum_{\tau=0}^{\tau=t} R_{in}(\tau)$
$A_{out}(t) = \sum_{\tau=0}^{\tau=t} R_{out}(\tau)$

Figure 2.8: The General system $\mathcal{S}$.

fitting a single (σ, ρ).

2.4 Admission Control

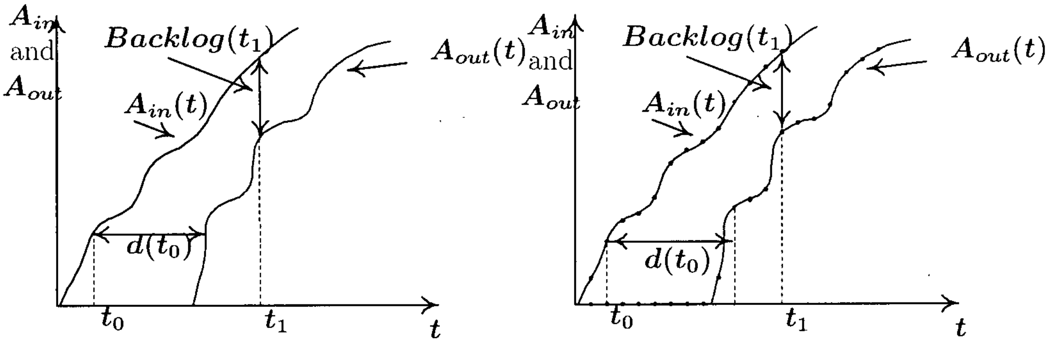
In this section we present a method which finds a bound on the waiting time of the incidental streams data. This method is based on the "Network Calculus" theory [19, 35, 36]. In Section 2.4.1 we discuss the basics of the Network Calculus and describe how the bound on the waiting time is obtained. Then, in Section 2.4.2 we present an algorithm that employs this method and finds T_w.

2.4.1 Network Calculus Basics

Consider a general system $\mathcal{S}$ which is viewed as a black box; $\mathcal{S}$ receives the input data at the variable rate $y_{in}(t)$ and delivers the data after a variable waiting time at the variable rate $y_{out}(t)$. In the proposed ITV application, $\mathcal{S}$ is the multiplexer on the transmitter side, that multiplexes the main and incidental streams. We define the cumulative function of the amount of data input and output to $\mathcal{S}$ as

$$\begin{aligned} A_{in}(t) &= \sum_{\tau=0}^{\tau=t} y_{in}(\tau), \qquad \forall\tau \geq 0, \forall t \geq 0 \\ A_{out}(t) &= \sum_{\tau=0}^{\tau=t} y_{out}(\tau), \qquad \forall\tau \geq 0, \forall t \geq 0 \end{aligned} \tag{2.27}$$

$A_{in}(t)$ and $A_{out}(t)$ are called the 'arrival' and 'departure' functions, respectively [35,36]. The arrival and departure functions for a sample system are



(a) Continuous time.

(b) Discrete time.

Figure 2.9: The arrival and departure functions: (a) Continuous time, where A_{in} and A_{out} are defined for all $t \geq 0$, (b) Discrete time, where A_{in} and A_{out} are only defined at discrete times denoted by dots.

illustrated in Figure 2.9. From the arrival and departure functions we derive the following two quantities:

Backlog: The backlog at time t is the amount of data waiting in the system $\mathcal{S}$ at time t and is given by $A_{in}(t) - A_{out}(t)$. As shown in Figure 2.9, the backlog at t_1 is simply the vertical distance between $A_{in}(t_1)$ and $A_{out}(t_2)$.

Waiting time (or delay): The waiting time at time t is the time that the incoming data at time t will wait in the system $\mathcal{S}$ before being served. The waiting time for the data that is input to the system at time t is given by:

$$d(t) = \min\{\tau \geq 0 : A_{in}(t) \leq A_{out}(t+\tau)\} \tag{2.28}$$

The waiting time at t_0 is illustrated by $d(t_0)$ in Figure 2.9. If the traffic is continuous, then $A_{in}(t_0) = A_{out}(t_0 + d(t_0))$, which means that all the input data to the system up to the time t_0 are served by the time $t_0 + d(t_0)$. As shown, for continuous traffic the waiting time is simply the *horizontal distance* between A_{in} and A_{out}.

Bounds on Waiting Time and Backlog

Network Calculus gives computational rules for bounding the waiting time and backlog. Before discussing how these bounds are obtained, we need to define the *service curve* and the *horizontal deviation* concepts:

Service Curve: Assume $A_{in}(t)$ and $A_{out}(t)$ are given functions. We say that the system $\mathcal{S}$ offers the input a service curve β if and only if for all $t \geq 0$, there exists some s, $0 \leq s \leq t$ such that

$$A_{out}(t) - A_{in}(t-s) \geq \beta(s) \tag{2.29}$$

where $\beta(t) \geq 0$ for all $t \geq 0$.

The service curve is an abstract concept, and indicates the capacity of system in accommodating traffic during a time interval of length s. Roughly speaking, $\beta(s)$ is a lower bound on the amount of traffic that can depart from the system during any time interval of length s, that is $A_{out}(t) - A_{out}(t-s) \geq \beta(s)$. To better understand the physical meaning of $\beta(s)$ we write the equation 2.29 as

$$A_{out}(t) \geq A_{in}(s) + \beta(t-s) \tag{2.30}$$

Then, a more precise physical interpretation of β is that if s is the beginning of a busy period, that is the backlog at s is zero ($A_{out}(s) - A_{in}(s) = 0$) and there are always some data waiting in the system in $[s,t]$, then the system will send at least $\beta(t-s)$ data units in $[s,t]$.

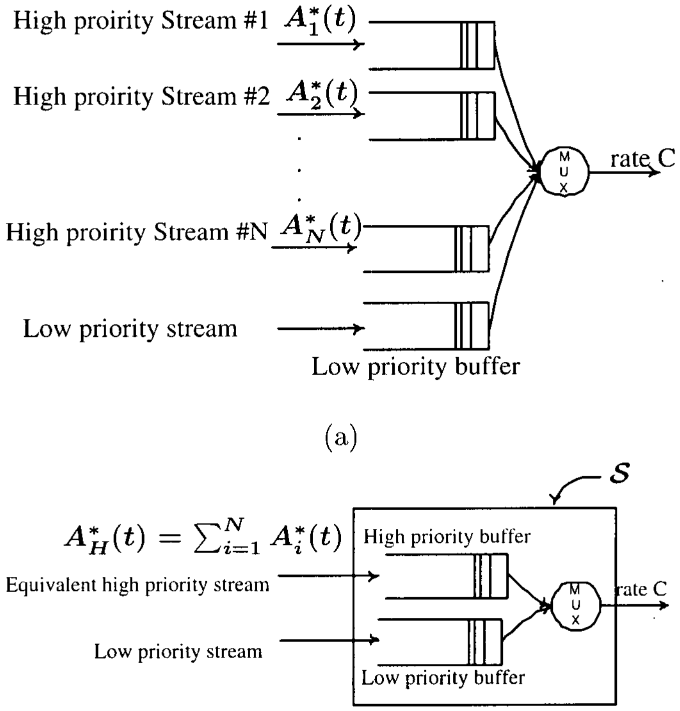
Horizontal deviation: The horizontal deviation between the arrival and departure functions denoted by $h(A_{in}, A_{out})$ is defined as the maximum of all the waiting time values $d(t)$, and mathematically is defined as

$$\begin{aligned} h(A_{in}, A_{out}) &= \max_t d(t) \\ &= \max_t \{\min\{\tau \geq 0 : A_{in}(t) \leq A_{out}(t+\tau)\}\} \end{aligned} \tag{2.31}$$

Now assume the input traffic to the system is characterized by the traffic constraint function $A^*_{in}(t)$. This means that for all $t \geq s \geq 0$ (see 2.2)

$$A_{in}(t) - A_{in}(s) \leq A^*_{in}(t-s) \tag{2.32}$$

Two theorems in the Network Calculus state that the backlog and waiting time in a system are bounded respectively by the vertical and horizontal deviations between the traffic constraint function of the input $A^*_{in}(t)$ and the service curve of the system $\beta(t)$ (see [19, 20, 35, 36]). Since we are interested in the bound on the waiting time, we only state the theorem that defines a bound on the waiting time:


High proirity Stream #1 $A_1^*(t)$
High proirity Stream #2 $A_2^*(t)$
High proirity Stream #N $A_N^*(t)$
Low priority stream
Low priority buffer
M U X
rate C
$\mathcal{S}$
$A_H^*(t) = \sum_{i=1}^N A_i^*(t)$
High priority buffer
Equivalent high priority stream
Low priority stream
Low priority buffer
M U X
rate C

(a)

(b)

Figure 2.10: a) The transmitter model with N high priority (main) streams. b) The equivalent model, where the N high priority input streams are replaced with one equivalent stream.

Theorem 1 *Assume a traffic source constrained by $A_{in}^*(t)$ traverses a system $\mathcal{S}$ that offers the service curve $\beta(t)$. The waiting time $d(t)$ for all t satisfies: $d(t) \leq h(A_{in}^*, \beta)$ [35, 36].*

For a proof of this theorem, see Appendix 1.

2.4.2 Waiting-Time Bound in the proposed ITV application.

We model the multiplexer in our interactive TV application with the model shown in Figure 2.10-a. In this model, the inputs to the system are N main streams and one incidental stream. The main streams have transmission priority over the incidental stream, i.e., the multiplexer serves the incidental stream only when the main stream's buffers are empty. The rate of the outgoing channel is constant, equal to C packets per second. We assume the traffic of the i^{th} main stream is characterized by the traffic constraint function $A_i^*(t)$. As discussed in 2.2.1, all the main streams can be replaced with one equivalent high-priority stream, which is constrained by $A_H^*(t)$ where

$$A_H^*(t) = \sum_{i=1}^{N} A_i^*(t) \tag{2.33}$$

This is shown in Figure 2.10-b. We denote the arrival and departure functions for the low-priority input (i.e., the incidental stream) by $A_{in,L}(t)$ and $A_{out,L}(t)$.

Our motivation here is to first find the service curve for the low-priority input. Then, using theorem 1 we will find T_w. Consider an arbitrary time t. Call $s < t$ the beginning of a busy period for the low-priority input, i.e., the backlog for low-priority input at s is zero ($A_{in,L}(s) = A_{out,L}(s)$) and there is always some low-priority data waiting in the system during $[s, t]$. During $[s, t]$ the high priority inputs can send up to $A_H^*(t-s)$ packets to the system. Hence the system will send at least $C(t-s) - A_H^*(t-s)$ packets of the low-priority input in $[s, t]$:

$$A_{out,L}(t) - A_{out,L}(s) \geq C(t-s) - A_H^*(t-s) \tag{2.34}$$

Since the backlog at s is zero then we have

$$A_{out,L}(t) - A_{in,L}(s) \geq C(t-s) - A_H^*(t-s) \tag{2.35}$$

It follows from this equation that the service curve for the low-priority input is $\beta_L(t) = Ct - A_H^*(t)$.

The traffic constraint function for a constant bitrate incidental stream is given by

$$A_{in,L}^*(t) = Rt$$

where R is the rate of the stream. We use theorem 1to find the maximum waiting time T_w in the proposed ITV application. That is, we consider a hypothetical system where the arrival function is $A_{in,L}^*(t)$ and the departure function is $\beta_L(t)$. Theorem 1 states that T_w, defined as the maximum of $d(t)$, is given by the horizontal deviation between $A_{in,L}^*$ and β_L, i.e., $T_w = h(A_{in,L}^*, \beta_L)$, where $\beta_L(t)$ and $A_{in,L}^*(t)$ are given in equations 2.35 and 2.36.

Algorithm 1 Find T_W

INPUTS: (σ_i, ρ_i) pairs for $i = 1, 2, \ldots, m$; the incidental stream rate R; the incidental stream duration T and the channel rate C.

OUTPUT: T_W

```
for $i = 1$ to $m$ do
    $t_i = \frac{\sigma_{i-1}-\sigma_i}{\rho_i-\rho_{i-1}}$
    $t_i$ is the abscissa of the intersection of the $i^{th}$ and $i-1^{th}$ line segments of $\beta_L$.
    $d_i = \frac{(R-C+\rho_i)\times t+\sigma_i}{R}$
    $d_i$ is the horizontal distance between the $R \times t$ and $\beta_L(t_i)$.
end for
if $T \neq \infty$ then
    Find $T'$ such that $\beta_L(T') = R \times T$
    $d_T = T' - T$
else
    if $\max_i\{C - \rho_i\} \leq R$ then
        $d_T = \infty$
    else
        $d_T = 0$
    end if
end if
$T_W = \max\{d_T, \max_i\{d_i\}\}$
RETURN
```

Figure 2.11: Pseudo-code of our algorithm to find T_w.

2.4.3 Our Algorithm to Find T_w

Our algorithm is presented in Figure 2.11, which uses the method presented in the previous section and finds T_w. The inputs to the algorithm are the m parameter pairs (σ_i, ρ_i) of the main stream constraint function $A^*_H(t)$, the channel output rate C, the duration of the incidental stream T and the rate of the incidental stream R. We have

$$\beta_L(t) = Ct - A^*_H(t) = Ct - \min_i\{\sigma_i + \rho_i t\} \tag{2.36}$$

$$\beta_L(t) = \max_i\{-\sigma_i + (C - \rho_i)t\} \tag{2.37}$$

As shown in Figure 2.12, $\beta_L(t)$ is a convex piece-wise linear and non-decreasing function. First, the algorithm finds t_i for $i = 1, 2, .., m$, where t_i is the abscissa

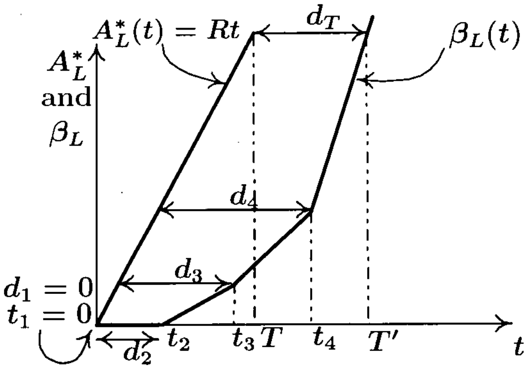


Figure 2.12: $\beta_L(t)$ and $A^*_L(t)$.

of the intersection of the i^{th} and $i-1^{th}$ line segments of β_L. In the next step, the algorithm computes the horizontal distance d_i between A^*_L and $\beta_L(t)$ for $t = t_i$, $i = 1, 2, .., m$ (see Figure 2.12):

$$d_i = t_i - \frac{\beta_L(t_i)}{R} = \frac{(R - C + \rho_i)t_i + \sigma_i}{R} \tag{2.38}$$

If the incidental stream is a video or audio sequence of length T, then the horizontal deviation at $t = T$ is computed and denoted by d_T. Otherwise, that is if the incidental stream is a video or audio sequence with an unlimited duration, the algorithm checks if $\beta_L(t) \leq Rt$ for very large time intervals (i.e., for $t \longrightarrow \infty$). If this condition is not met, it means that the system cannot guarantee any maximum waiting time and d_T is set to ∞. Finally, T_w is found as the maximum of all d_i's and d_T.

2.5 Results

In this section, we present the numerical results of implementing the methods presented in this chapter. First, we will evaluate the performance and accuracy of our model fitting methods presented in sections 2.3.2 and 2.3.3 by comparing them with the current methods in literature [1, 17]. Then, we will present the results of implementing our admission control scheme using empirical traffic traces from video sequences of typical TV programs. These results demonstrate how the rate R and the maximum waiting-time T_w depend on each other in a typical digital TV system.

2.5.1 Numerical Results of 'Model Fitting' methods

In our first experiment, we evaluated the performance and accuracy of our model fitting methods presented in sections 2.3.2 and 2.3.3 by comparing them with the current methods in literature [1, 17]. The traffic source used in our experiment is an MPEG-2 video stream from the motion picture "Mission Impossible". This video stream was encoded with constant picture quality, with picture resolution 720 × 480, frame rate 30 and maximum bitrate of 4.5 Mbps. The length of this video was about one hour, which corresponds to $N \approx 10^5$ frames.

In our simulation, the time parameter is an integer number $t \in \mathbb{N} = \{0, 1, 2, \dots\}$ that represents the GOP number in the MPEG video stream. Since the frame rate of the video stream is 30 fps and the GOP size is 15, each GOP is 0.5 seconds. Thus, the 25 minutes sequence represents 3000 GOPs. The traffic in our simulation is also discrete and represents the number of packets. We use constant size packets of 184 bytes. This conforms to the digital TV and MPEG-2 standards[5]. For example, if a source generates 2 MBits in the time interval $[0, .5]$ (i.e., in the first GOP), then the discrete traffic is represented by $y(1) = ceil[(2 \times 10^6)/(184 \times 8)] = 1359$ packets. In order to make the results transparent from the maximum bitrate of the source, we normalize the traffic to 1 by dividing y by its maximum. For example, in the previous example, if the maximum bitrate of the source is 4.5 Mbps and GOP-time = .5 seconds, then we have $y(1) = \frac{ceil[(2 \times 10^6)/(184 \times 8)]}{ceil[(4.5 \times 10^6 * .5)/(184 \times 8)]} = 1359/1529 \simeq .88$.

In our first experiment, we construct the empirical envelope $E(t)$, and compare the speed of our algorithm with other methods in the literature. We construct $E(t)$ using the direct method [17], the largest sub-additive extrapolation method [1], the repetition extrapolation method [1], and our methods presented in Section 2.3.2. Table 2.1 summarizes the computation time of each method. As shown, our algorithm speed is almost the same as that of the repetition extrapolation approach, and both are considerably faster than the direct method. However, the extrapolation approaches do not estimate $E(t)$ for $t > \tau$ closely, while our method finds the exact samples of $E(t)$ for $t > \tau$ (see Figure 2.5).

In our second experiment, we fit a $(\vec{\sigma}, \vec{\rho})$ model to the whole empirical

[5] The American and the European digital TV standards employ the MPEG-2 transport stream (TS) syntax for the transmission stream. Each packet in MPEG-2 TS syntax contains 184 bytes of data payload plus 4 bytes header.

Computation Method	**Execution time (seconds)**
Direct approach	161.27
Repetition extrapolation $\tau = 200$	17.28
Sub-additive extrapolation $\tau = 200$	150.48
Our sampling method $\tau = 200$, $delta = 50$	23.60
Our sampling method $\tau = 200$, $delta = 100$	16.24
Our sampling method $\tau = 200$, $delta = 200$	7.89

Table 2.1: Execution time in seconds for calculating $E(t)$ for $1 \leq t \leq 10000$. Simulations were run on a PC with pentium IV processor at 1.7 GHz, using Matlab implementation.

envelope $E(t)$ using our method presented in Section 2.3.3, and the heuristic method presented in [1]. We computed the error function $\sum_{t=1}^{N} \frac{A^*(t)-E(t)}{E(t)}$ as a metric for the accuracy of each model. Table 2.2 summarizes the results. As shown, our method results in a more accurate model for the source than the method in [1].

In order to evaluate the effect of our sampling approach on the accuracy of the constructed $(\vec{\sigma}, \vec{\rho})$ model, we fit a $(\vec{\sigma}, \vec{\rho})$ model to the empirical envelope constructed by our sampling approach. Table 2.3 summarizes the results. As shown, the model parameters and the error function for the models constructed from the samples of $E(t)$ are fairly close to the models constructed from the whole $E(t)$.

In our next experiment, we evaluate the accuracy of our method with respect to achieving high bandwidth utilizations. For this purpose, we use a metric that determines how closely a particular model $A^*(t)$ approximates $E(t)$ with respect to bandwidth utilization [1]. We consider a single FCFS multiplexer with a switch that operates at $r = 155$ Mbps. We assume that all the input traffic sources connected to this switch are from the same source, which are all characterized by $A^*(t)$. We also assume that all the sources have an identical delay bound d. Assuming n sources are connected to this

	our method	method in [1]
$(\vec{\sigma}, \vec{\rho})$	$\begin{bmatrix} 0 & 1.0000 \\ 53.4927 & 0.3805 \\ 11.4585 & 0.4622 \\ 2.9396 & 0.5837 \\ 1.7966 & 0.6573 \end{bmatrix}$	$\begin{bmatrix} 0 & 1.0000 \\ 50.1352 & 0.3063 \\ 13.4325 & 0.3293 \\ 3.5398 & 0.5644 \\ 1.2903 & 0.6888 \end{bmatrix}$
Error	65.47	93.96

Table 2.2: Comparison of our method with the method in [1] for finding the $(\vec{\sigma}, \vec{\rho})$ parameters. $E(t)$ constructed for all $1 \leq t \leq N$.

	entire envelope	$\delta = 100$	$\delta = 400$
$(\vec{\sigma}, \vec{\rho})$	$\begin{bmatrix} 0 & 1.0000 \\ 53.4927 & 0.3805 \\ 11.4585 & 0.4622 \\ 2.9396 & 0.5837 \\ 1.7966 & 0.6573 \end{bmatrix}$	$\begin{bmatrix} 0 & 1.0000 \\ 53.4511 & 0.3805 \\ 12.6243 & 0.4546 \\ 4.4318 & 0.5515 \\ 2.3172 & 0.6117 \end{bmatrix}$	$\begin{bmatrix} 0 & 1.0000 \\ 90.5083 & 0.3567 \\ 40.9453 & 0.3961 \\ 13.9549 & 0.4475 \\ 6.6479 & 0.5189 \end{bmatrix}$
Error	65.47	76.74	84.12

Table 2.3: The model parameters constructed from the entire envelope (second column) and from $E(t)$ for $1 \leq t \leq \tau$ and samples of $E(t)$ at $t = i\delta$ for $1 \leq i \leq n$. $\tau = 200$, $n\delta = N$.

switch. Then, as discussed in [15], these sources are supported by this FCFS multiplexer without maximum waiting time violation if and only if

$$d \geq \sum_{i=1}^{n} A^*(t) - t \times r, \qquad \forall t \geq 0 \tag{2.39}$$

We define the 'utilization ratio', $U(A^*, d)$, as the number of admissible sources using $A^*(t)$ to the number of admissible sources using the empirical envelope $E(t)$ at maximum waiting time d. Particularly, $U(A^*, d)$ is the maximum n that satisfies equation 2.39, divided by the maximum m that satisfies $d \geq \sum_{i=1}^{m} E(t) - t \times r$ for $\forall t \geq 0$. $U(A^*, d)$ shows how closely $A^*(t)$ approximates $E(t)$ with respect to bandwidth utilization. An ideal model, which admits the same number of streams as the empirical envelope, results in the constant $U(A^*, d) = 1$. Figure 2.13 shows $U(A^*, d)$ for a $(\vec{\sigma}, \vec{\rho})$ model of size 4, constructed using our method and the method presented in [1]. As shown, our method results in a higher utilization ratio than that of the method in [1].

Figure 2.13 also shows the utilization ratio for a $(\vec{\sigma}, \vec{\rho})$ model constructed from samples of $E(t)$. As shown, the utilization ratio for the model constructed

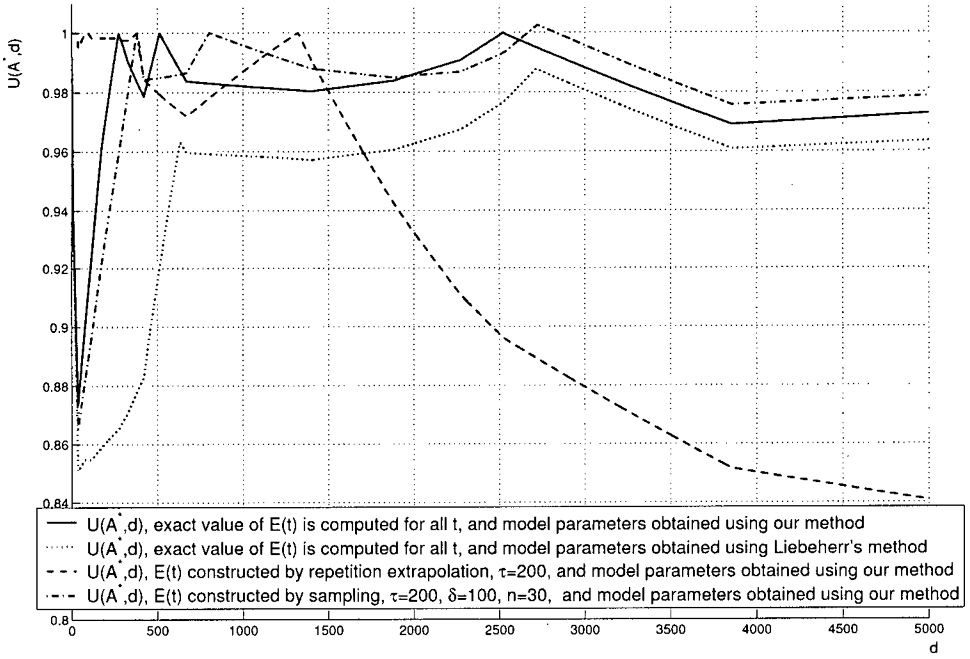


Figure 2.13: Utilization ratio, $U(A^*, d)$.

from samples of the envelope is very close to the utilization ratio for the model constructed from the entire envelope. This means that using some samples of $E(t)$ are sufficient to construct an accurate $(\vec{\sigma}, \vec{\rho})$ model and the computation of the entire envelope is not required.

We also study the utilization ratio curve for different parameters δ and τ. In practical applications, δ and τ should be selected such that the utilization ratio is close to one for the selected delay bound d, and the computation time is reasonable for the application. Based on this experiment, we suggest $100 \leq \tau \leq 300$, and $\delta = \tau$. This selection results in a reasonable utilization ratio for almost all maximum waiting times d.

2.5.2 Numerical Results of 'Admission Control Mechanism'

In this section, we present the results of applying our admission control method to a typical digital TV programs. All the video sequences used as the main

	Sequence Name	*Source type*
1	Mission Impossible	Action movie
2	Muppets	Children TV show
3	News	News show
4	Talk Show	Oprah Winfrey Show
4	Documentary	Documentary
5	Court Show	Judge Judy show
6	Muppets show	Sesame St. show
7	Soap opera	Days of our lives
8	Cartoon	Tigger movie

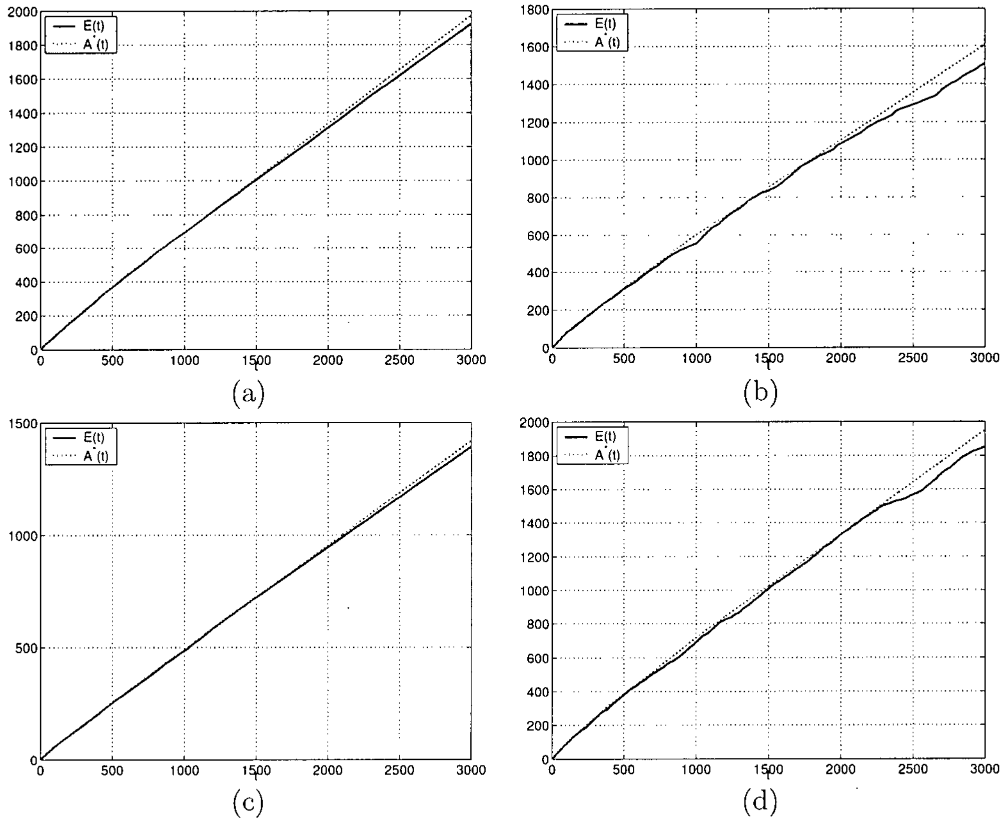
Table 2.4: Video sequences used in our study.

Compression Standard	MPEG-2
Resolution	720×480
Rate	4.50 Mbps
Frame rate	30 fps
GOP size	15
Number of P frames in each GOP	4
Number of B frames in each GOP	10

Table 2.5: Encoding parameters for the video streams used in our study.

video streams (see table 2.4) were selected from typical TV programs. These video sequences were encoded with constant picture quality, and with a maximum bitrate of 4.5 Mbps. The picture quality of these videos was subjectively selected to be at a satisfactory level for TV applications. The length of every sequence used was 25 minutes. Table 2.5 summarizes the encoding parameters of the video sequences used in our simulation.

By studying the empirical envelopes of the main video sequences, we observed that with $m = 5$ (σ, ρ) pairs one can accurately model the video of most TV programs. Therefore, we select $m = 5$ in our application. Figures 2.14 and 2.15 show the empirical envelopes and the fitted $(\vec{\sigma}, \vec{\rho})$ models for the traffic sources. Table 2.6 shows the numerical value of the $(\vec{\sigma}, \vec{\rho})$ model parameters for each source. As shown, $E(t)$ is an increasing function where $E(t)/t \approx 1$ for small t's, and $dE(t)/dt$ is approximately the average rate of the source for large t's. $E(t)$ drops faster for large t's in video sources with simple content than video sources with active and complex content. The fitted $(\vec{\sigma}, \vec{\rho})$



(a)

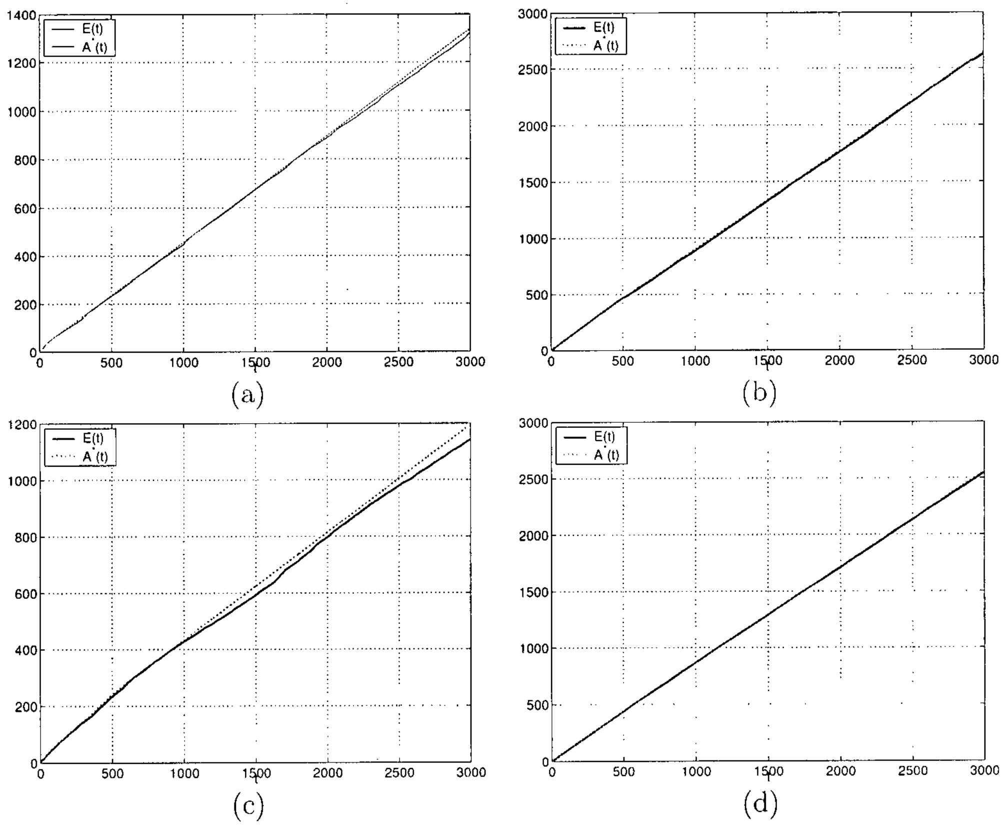
(b)

(c)

(d)

Figure 2.14: $A^*(t)$ and $E(t)$ of video sequences' of typical TV programs, a) Mission Impossible, b) News, c) Talk show, d) Documentary.

model with five (σ, ρ) pairs is also shown in these figures. The (σ, ρ) pairs are selected such that $(\sigma_1, \rho_1) = (0, 1)$ and the rate parameter of the last (σ, ρ) pair (i.e., ρ_5) is the average rate of the source. As shown, the model can approximate the $E(t)$ very well with only a few (σ, ρ) pairs in the $(\vec{\sigma}, \vec{\rho})$ model.



(a)

(b)

(c)

(d)

Figure 2.15: $A^*(t)$ and $E(t)$ of video sequences' of typical TV programs, a) Court show, b) Muppets, c) Soap Opera, d) Cartoon.

	Sequence Name	$(\vec{\sigma}, \vec{\rho})$ parameters
1	Mission Impossible	$\begin{bmatrix} 60.5602 & 0.6376 \\ 20.1277 & 0.7025 \\ 11.3654 & 0.7332 \\ 7.8706 & 0.7615 \\ 0 & 1.0000 \end{bmatrix}$
2	News	$\begin{bmatrix} 100.9816 & 0.5018 \\ 30.4283 & 0.5727 \\ 12.9072 & 0.6938 \\ 2.1920 & 0.8130 \\ 0 & 1.0000 \end{bmatrix}$
3	Talk Show	$\begin{bmatrix} 27.7782 & 0.4642 \\ 11.0175 & 0.4860 \\ 4.9481 & 0.5433 \\ 1.7709 & 0.6584 \\ 0 & 1.0000 \end{bmatrix}$
4	Documentary	$\begin{bmatrix} 103.1472 & 0.6157 \\ 53.8328 & 0.6629 \\ 20.7147 & 0.7503 \\ 4.5189 & 0.8782 \\ 0 & 1.0000 \end{bmatrix}$
5	Court Show	$\begin{bmatrix} 16.7626 & 0.4403 \\ 5.8024 & 0.6030 \\ 2.3032 & 0.7112 \\ 0.6417 & 0.8298 \\ 0 & 1.0000 \end{bmatrix}$
6	Muppets show	$\begin{bmatrix} 40.4100 & 0.8688 \\ 36.5059 & 0.8703 \\ 34.3194 & 0.8740 \\ 3.8070 & 0.9445 \\ 0 & 1.0000 \end{bmatrix}$
7	Soap opera	$\begin{bmatrix} 53.4927 & 0.3805 \\ 11.4585 & 0.4622 \\ 2.9396 & 0.5837 \\ 1.7966 & 0.6573 \\ 0 & 1.0000 \end{bmatrix}$
8	Cartoon	$\begin{bmatrix} 23.4084 & 0.8482 \\ 8.3675 & 0.8680 \\ 4.6805 & 0.8811 \\ 0.6067 & 0.9556 \\ 0 & 1.0000 \end{bmatrix}$

Table 2.6: Numerical values of $(\vec{\sigma}, \vec{\rho})$ model parameters for the main video sequences used in our simulation.

	Simulation parameter set I (cable medium)	Simulation parameter set II (terrestrial medium)
Transmission rate	19.8 Mbps	39.8 Mbps
Number of TV programs sharing the channel, N	4	8
Maximum bitrate assigned to each main video stream	4.5 Mbps	4.5 Mbps
Transmission capacity reserved for video streams	18 Mbps	36 Mbps
Transmission capacity reserved for audio streams and other ancillary data	1.8 Mbps	3.8 Mbps
Main video stream sources	1. Mission Impossible 2. News 3. Talk Show 4. Documentary	1. Mission Impossible 2. News 3. Talk Show 4. Documentary 5. Court Show 6. Muppets show 7. Soap opera 8. Cartoon

Table 2.7: Simulation parameters.

After fitting a $(\vec{\sigma}, \vec{\rho})$ model to each video sequence, we conducted another set of experiments where we considered a system similar to Figure 2.10, which consisted of N main (high-priority) streams and 1 incidental (low-priority) stream. This system simulates the head-end of a digital TV transmission system. We conducted two experiments using two different simulation parameters, as shown in Table 2.7. The first set of parameters are selected to simulate cable transmission medium, while the second set simulates a terrestrial medium[6]. As noted, no portion of the transmission capacity is reserved for incidental streams.

In order to illustrate the relation between R and T_w for the incidental stream, we plotted R versus T_w as shown in Figure 2.16. This graph is interesting as it provides exemplary numerical values for the rate of an incidental stream in a typical digital TV transmission system. As we expected, R is an

[6] A 6 MHz channel in the cable medium is capable of delivering digital data at the 19.8 Mbps rate. This capacity is usually shared by 4 or 5 TV programs. In terrestrial medium, a 6 MHz channel is capable of delivering at the 39.8 Mbps rate, which is usually shared by 8 or 9 TV programs.

increasing function of T_w. This means that by allowing a larger waiting time in the multiplexing system, the system can accept higher rate incidental streams. However, for very large T_w, R becomes constant and equal to $C - R_{avg}$, where C is the transmission rate reserved for the main video streams and R_{avg} is the total average rate of all the main streams. This is due to the fact that even by increasing T_w, R cannot become larger than the channel rate minus the main streams average rate.

In next experiment, we tested the accuracy of our admission control scheme via simulation by observing the waiting time of the incidental streams data units during multiplexing. The results showed that if an incidental stream with rate R and maximum waiting time T_w is accepted by the admission test, then the waiting time of its data units in the system is always less than T_w. However, for the incidental streams which were rejected by the admission test, the waiting time of some data units was more than T_w seconds.

2.6 Conclusion

In this chapter, we presented methods for implementing the deterministic service class. We employed a model for the traffic of main video sources. We used the concept of traffic constraint function, and the empirical envelope as the tightest traffic constraint function. After discussing the current approaches to deterministic traffic modelling, we selected the $(\vec{\sigma}, \vec{\rho})$ model as the traffic model for our application. We showed that the $(\vec{\sigma}, \vec{\rho})$ model can accurately model the empirical envelope of main video sources.

Then, we presented efficient methods for fitting the $(\vec{\sigma}, \vec{\rho})$ model to a traffic source. We showed that our model fitting methods result in a more efficient and more accurate model parameters than other methods in the literature.

Next, we adapted the newly developed 'Network Calculus' theory, and designed an admission control mechanism for the deterministic service class of the proposed ITV application. Our admission control mechanism finds the maximum waiting-time T_w for an incidental stream with rate R.

Our simulation results provided some exemplary numerical values for the maximum waiting-time T_w and the rate R of an incidental stream in a typical digital TV system.

The deterministic admission control scheme presented in this chapter

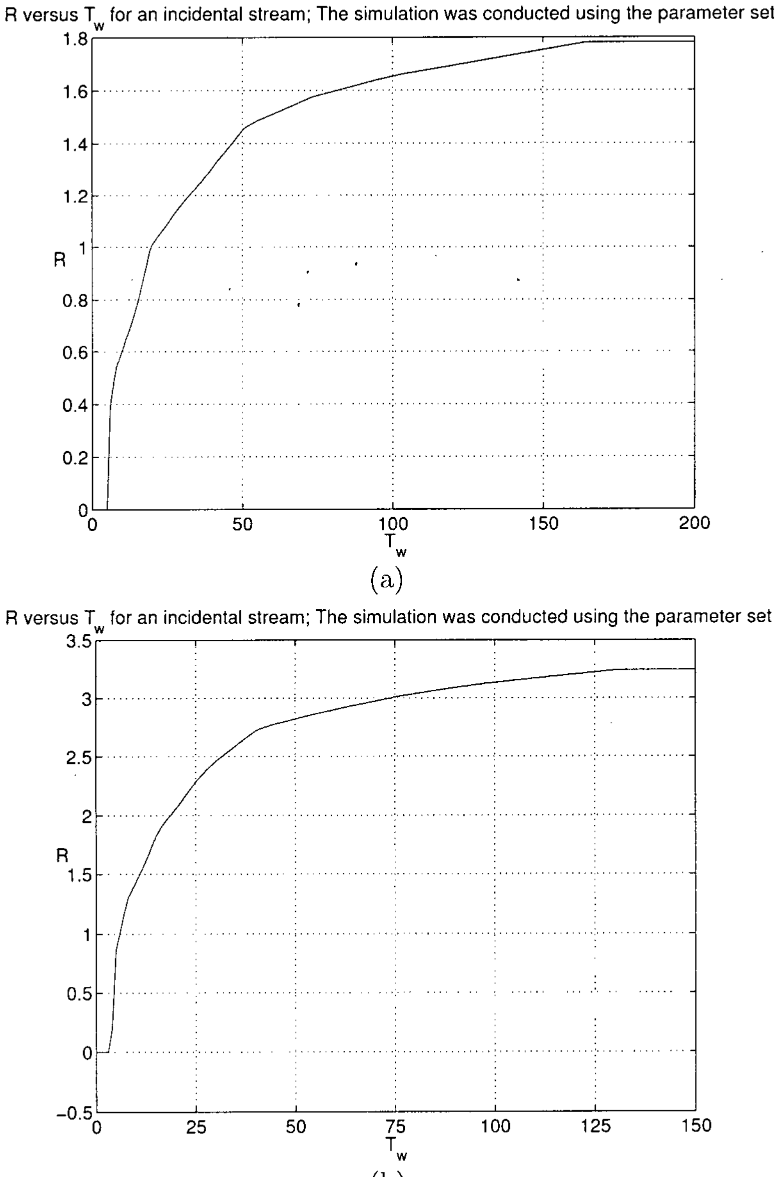


Figure 2.16: R versus T_w for an incidental stream. T_w is obtained for each R using the admission control algorithm presented in Section 2.4.3.

relies on the traffic constraint function of the main video streams, which is a *worst-case* estimate of the traffic a main video source can generate. Therefore, the obtained T_w is based on the most *pessimistic* forecast of the system. This approach is attractive since it ensures that no incidental data packet will be lost. However, it does not result in high utilization of available bandwidth. In the next chapter, we will discuss the *stochastic* service class, where T_w is found such that some data loss is possible, however, this data loss is limited. Our design of the stochastic service class is fundamentally different from the deterministic service class, and is based on a different type of traffic models.

Chapter 3

Stochastic Service Class

Chance favors only the prepared mind.

-Louis Pasteur, A New Kind of Country, 1978.

Overview

This Chapter presents a scheme for implementing the stochastic service class based on the 'effective bandwidth' theory. The effective bandwidth characteristics are exploited. We also show how the effective bandwidth is used to design an admission control scheme for the stochastic service class of the proposed ITV application. Using the methods presented in this chapter, one can find T_w for an incidental stream with given rate R and data loss probability p.

3.1 Introduction

In this chapter, we develop a method for implementing the 'Stochastic Service Class.' As discussed in Chapter 1, when an incidental stream is added to a TV program using the stochastic service class, the transmitter does not guarantee to send all the incidental data units on time (i.e., before their transmission deadline). The data units which are not transmitted on time are considered lost data. Therefore, some incidental data loss is probable in the stochastic service class. However, the rate and waiting time of an incidental stream should be selected so that the data loss probability is less than a threshold. Therefore, an incidental stream that is to be transmitted using the stochastic service class should be first accepted by an admission control mechanism. This admission control mechanism verifies that the transmitter can send this incidental stream with rate R and maximum waiting time t_w and with a data loss probability less than a given threshold, say $p\%$. Our ultimate goal in this chapter is to present a scheme, by which the admission control finds the maximum waiting-time T_w, given the rate R and the loss probability $p\%$ for an incidental stream.

A key issue in implementing the stochastic service class is to find a suitable traffic descriptor, which encapsulates the stochastic properties of the main streams traffic. Then an accurate admission control using the selected traffic descriptor should be designed. Our approach here is based on the theory of *'effective bandwidth'*. The main motivation behind this theory is to provide a measure of bandwidth usage by a traffic source in a communication network, which can adequately represent the statistical characteristics of the source. In this theory, each traffic source is described with a traffic descriptor called 'effective bandwidth curve'. This theory then provides mechanisms to find a level of statistical service guarantee for usual network operations, such as multiplexing, buffering, etc. We use this theory to design the admission control mechanisms of our application problem.

The rest of this chapter is organized as follows. In Section 3.2, the effective bandwidth is defined and its characteristics are described. In Section 3.3, we show how the effective bandwidth is used to bound the data loss in general network operations. Based on this theory, we design an admission control mechanism for the stochastic service class in Section 3.4. In Section 3.5, we discuss the current approaches to the numerical estimation of the effective bandwidth curve.

3.2 Effective Bandwidth

The theory of effective bandwidth was first introduced in the early 1990's by Gibbens and Hunt [37], Kelly [38], and Guerin *et al* [39]. Since then, this theory has attracted much attention from both the mathematics and engineering communities, and emerged as a powerful but complicated mathematical theory. Currently a great effort is in progress to expand the effective bandwidth theory and its applications.

The associated mathematical theory of the effective bandwidth concept is built upon the theory of *Large Deviation Principle*, LDP, which studies the tail properties of probability distributions. The effective bandwidth of a source is closely related to the moment generating function of the arrival process of the source. The moment generating function of a random variable contains more information about the stochastic characteristics of a process than its mean. Hence, traffic characterization methods based on the effective bandwidth function are more accurate than the widely used traffic characterization methods based on 'Poisson processes', which rely on the average rate.

A useful interpretation of the effective bandwidth concept is that the effective bandwidth theory gives the probability of a traffic source generating traffic at a rate higher than its average for a long period of time. More precisely, let us denote the instantaneous rate of a traffic source by $y(t)$ and the average rate of the source by μ. Then we expect $\sum_{\tau=1}^{t} y(t)$ to be close to μt for large t's. Effective bandwidth theory bounds the probability that $\sum_{\tau=1}^{t} y(t) \approx ta$, where $a > \mu$. Hence, under some mild conditions, the effective bandwidth theory gives the *probability* that a variable rate source generates traffic that is equal to a constant rate source with rate a during a long period. The probability that the source follows the effective bandwidth model is incorporated into the model through a parameter named '*scale factor*', θ. Therefore, the effective bandwidth is a function of the scale factor, usually denoted as $\alpha(\theta)$. The effective bandwidth value lies *always between the average and the peak rate of the source.* Higher levels of certainty result in a larger θ and an effective bandwidth that is closer to the peak rate, e.g., a certainty value equal to one corresponds to $\theta = \infty$ and $\alpha(\infty)$ is the peak rate of the source.

The effective bandwidth concept can be viewed as a compromise between two alternative bandwidth allocation schemes, a pessimistic outlook and an optimistic one. In the pessimistic case, one uses a strict approach to bandwidth allocation, where the bandwidth allocation is based on the sources peak rate. This approach seeks to eliminate data loss. In the optimistic case, one uses a lenient approach to bandwidth allocation, where the bandwidth is allocated based on the source's average rate. This approach seeks to gain high bandwidth utilization. The effective bandwidth $\alpha(\theta)$ gives a spectrum between these two approaches, where the scale factor $0 \leq \theta \leq \infty$ determines how lenient or strict this approach is.

In next Section, we first briefly review the large deviation principle concept. Then, we present a precise definition of the effective bandwidth in Section 3.2.2.

3.2.1 Large Deviation Principle

As mentioned, the theory of effective bandwidth relies on the 'Large Deviation Principle', LDP. Large deviation principle is a theory that studies the tail properties of probability distributions. This theory refers to a collection of techniques used for estimating properties of *rare events*, such as the frequency of their occurrence, or the most likely manner of their occurrence.

Large deviations do not apply to any event that has a very low probability of occurrence. Roughly speaking, a large deviation event is caused by a large number of unlikely things occurring together, rather than a single event of small probabilities. For example, winning a lottery cannot be studied with large deviations, since it is a single event composed of a single trial and cannot be broken into more than one sub-event. However, the probability that the average grade of a class in an easy exam becomes very low can be considered a rare event, since it can be decomposed to the improbable sub-events that each individual student gets a very low mark. In the proposed ITV application, a large burst of traffic is generated by a source when the source starts sending traffic at a rate higher than its average, and continues to do so for a long period of time. Therefore, the occurrence of a large burst of traffic can be broken into many low-probability sub-events.

One can consider LDP as a tool to turn the probability problems into deterministic optimization problems. Loosely speaking, to calculate the probability of a rare event, one assigns a cost to each sample path that would cause an event to occur. In the example of having a very low average grade for a big class in an easy exam, a path is that all the students get a low mark, and an alternate path is that many students get zero and only a few get very good marks. Then one finds the cheapest (or the most probable) path in that set of sample paths. The probability of event is then estimated by:

$$\mathbb{P}(event) = e^{-n \times const} \tag{3.1}$$

where n is an asymptotic parameter, usually the length of time over which we observe the process. Therefore, one can think about the rare events in terms of sample paths and costs, and to find the probability of a rare event, one can simply consider the cheapest way that the event can happen.

General Definition Of LDP.

Let $y(1), y(2), \ldots$ be a sequence of a random processes. Let $Y(t) = \sum_{\tau=1}^{t} y(\tau)$. We say that $Y(1), Y(2), \ldots$ satisfies the large deviation principle with the rate function I if (see [40])

1. For every closed set $C \subset \mathbb{R}$ we have
$$\limsup_{n \to \infty} \frac{1}{n} \log \mathbb{P}(Y(n) \in C) \leq - \inf_{a \in C} I(a) \tag{3.2}$$

2. For every open set $G \subset \mathbb{R}$, we have
$$\liminf_{n \to \infty} \frac{1}{n} \log \mathbb{P}(Y(n) \in G) \geq - \inf_{a \in G} I(a) \tag{3.3}$$

It is shown in the LDP theory that if the random variable Y has a finite moment generating function $\mathbb{E}(e^{\theta Y})$ for all θ, where $\mathbb{E}$ denotes expected value, then Y satisfies the large deviation principle with the rate function $I = \Lambda^*$ where:

$$\Lambda(\theta) = \lim_{t \to \infty} \frac{1}{t} \log \mathbb{E}(e^{\theta Y(t)}) \tag{3.4}$$
$$\Lambda^*(a) = \sup_{\theta} [\theta a - \Lambda(\theta)] \tag{3.5}$$

$\Lambda(\theta)$ is called the 'Gartner-Ellis' limit. $\Lambda^*(a)$ is called the 'Legendre Transform' of $\Lambda(\theta)$. Also, it is shown that $\Lambda^*(a)$ has the following characteristics:

1. $\Lambda^*(a)$ is non-negative, i.e., $\Lambda^*(a) \geq 0$

2. $\Lambda^*(a)$ is strictly increasing in a.

3. $\Lambda^*(a)$ attains its minimum at $a = \mathbb{E}(y)$, i.e., at the average of y.

4. $\Lambda^*(\mathbb{E}(y)) = 0$

5. $\Lambda^*(a)$ is convex, i.e., $\Lambda^*(\alpha a_1 + (1-\alpha)a_2) \leq \alpha \Lambda^*(a_1) + (1-\alpha)\Lambda^*(a_2)$

6. If $\Lambda^*(a)$ is the Legendre transform of $\Lambda(\theta)$, then $\Lambda(\theta)$ is the Legendre transform of $\Lambda^*(a)$ as well, i.e., $(\Lambda^*)^*(\theta) = \Lambda(\theta)$. For this reason, the two functions Λ^* and Λ are called *Legendre transform pairs*.

3.2.2 Effective Bandwidth Definition

Let $y(t)$ denote the instantaneous rate of a traffic source. The arrival process is defined as $Y(t) = \sum_{\tau=0}^{t} y(\tau)$, which is the total traffic generated in $[0,t]$. Then the effective bandwidth for this source is defined as (see [41])

$$\alpha(\theta) = \lim_{t\to\infty} \frac{1}{\theta t} \log \mathbb{E}(e^{\theta Y(t)}), \quad \forall\theta \in \mathbb{R}^+ \tag{3.6}$$

Parameter θ is called *scale factor*. By definition, $\varphi_y(\theta) = \mathbb{E}(e^{\theta Y(t)})$ is the moment generating function of the random process $Y(t)$. Therefore, the effective bandwidth of a traffic source is closely related to the moment generating function of the arrival process of the source. Figure 3.1 shows a typical effective bandwidth curve for a traffic source with bounded peak rate. It has been shown that [41]

1. $\alpha(\theta)$ is a non-decreasing function of θ.

2. $\alpha(\theta)$ lies between the average and peak rate of the source, i.e., $\mathbb{E}(y) \leq \alpha(\theta) \leq \overline{y}$, where $\overline{y}$ denotes the peak rate of the source y.

3. If we have N sources, each with arrival process $Y_i(t)$, and if $Y(t)$ is the multiplexed stream of these sources, i.e., $Y(t) = \sum_i Y_i(t)$, then:

$$\alpha(\theta) = \sum_i \alpha_i(\theta) \tag{3.7}$$

4. It is shown that the shape of $\alpha(\theta)$ around $\theta = 0$ primarily depends on the mean, the variance and the higher moments of $Y(t)$, while the shape of $\alpha(\theta)$ for large θ's is primarily influenced by the tail distribution of $Y(t)$ around it's maximum. This can be justified using the Taylor expansion of $\alpha(\theta)$ at $\theta = 0$ and $1/\theta = 0$.

$$\alpha(\theta) = \eta_1 + \eta_2\theta + O(\theta^2) \tag{3.8}$$

$$\alpha(\theta) = \eta_3 - \eta_4\frac{1}{\theta} + O(\frac{1}{\theta^2}) \tag{3.9}$$

It is shown that $\eta_1 = \mathbb{E}(y)$, $\eta_2 = \frac{1}{2}\mathtt{var}(y)$, $\eta_3 = \overline{y}$ where $\overline{y}$ is the peak rate of y and η_4 is the average length of the periods that y is equal to its maximum, that is $\overline{y}$ is the average length of burst periods [41].

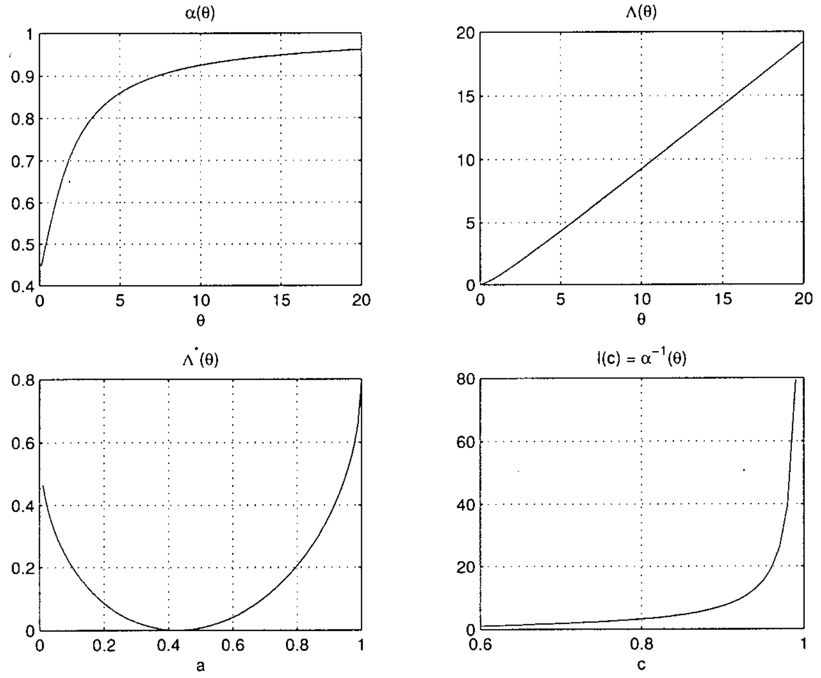


Figure 3.1: A sample effective bandwidth curve $\alpha(\theta)$, and $\Lambda(\theta)$, $\Lambda^*(\theta)$ and inverse of the effective bandwidth function $I(c) = \alpha^{-1}(\theta)$. The average rate of the source is .448.

3.2.3 Physical Interpretation of Effective Bandwidth

One useful physical interpretation of $\alpha(\theta)$ is that of using the effective bandwidth concept to approximate the behavior of a VBR source with a constant rate. Suppose $y(t)$ has a finite mean value $\mathbb{E}(y) = \mu$. We know from the law of large numbers that with probability 1, $Y(t)/t \rightarrow \mu$ as $t \rightarrow \infty$. Thus the probability that $Y(t)/t$ is away from μ goes to 0 as t increases. In the theory of large deviation principal, it is shown that this convergence to 0 occurs exponentially fast, that is, for $a \geq \mu$,

$$\lim_{t \to \infty} \frac{1}{t} \log \mathbb{P}(Y(t) \geq ta) = -\Lambda^*(a) \qquad (3.10)$$

where $\Lambda^*(a)$ is known as the *rate function.* Roughly speaking, equation 3.10 states that

$$\mathbb{P}(Y(t)/t \approx a) \approx e^{-t\Lambda^*(a)} \qquad (3.11)$$

Therefore, the value of $\Lambda^*(a)$ indicates how difficult it is for $Y(t)/t$ to be close to the constant rate a. $\Lambda^*(\theta)$ is related to $\alpha(\theta)$ with $\Lambda^*(a) = \sup_\theta[\theta a - \theta\alpha(\theta)]$. That is $\Lambda^*(a)$ is the Legendre transform of $\theta\alpha(\theta)$.

3.3 Quantifying Data Loss in Communication Networks Using Effective Bandwidth

In this Section, we describe how the effective bandwidth concept is used for quantifying the probability of loss in communication networks. Let us first consider a simple buffering model, where a single buffer of size B is filled by data from a variable rate source with rate $y(t)$, and is emptied at constant rate c. (see Figure 3.2). Let $W(t)$ denote the buffer workload at t. c is selected to be greater than the average input rate, but less than the maximum input rate. Therefore, we expect some data workload to build up in the buffer occasionally, and also that the buffer becomes empty regularly. It is shown that the probability of the buffer overflow, i.e., the probability that starting with an empty buffer the workload exceeds the buffer size before the buffer becomes empty again, is bounded as

$$\mathbb{P}(W(t) \geq B) \leq p \tag{3.12}$$

where $p = e^{-B\theta^*}$. θ^* is selected such that $\alpha(\theta^*) = c$ and t is large[1].

Equation 3.12 also gives a bound on the buffering delay, defined as $d(t) = W(t)/c$,

$$\mathbb{P}(d(t) \geq D) \leq p = e^{-B\theta^*} \tag{3.13}$$

where $D = B/c$ is the maximum acceptable buffering delay.

Validity of equations 3.12 and 3.13 have been proven [41–43]. An intuitional proof of these equations can be given based on equation 3.11 as follows. Note that a buffer overflow in this simple buffering model happens if the input has the rate $a > c$ for at least $B/(a-c)$ time units. According to equation 3.11, the probability that the input behaves in that way is approximately

$$\exp\{-\frac{B}{a-c}\Lambda^*(a)\} \tag{3.14}$$

[1] Note that the buffer workload random process, $W(t)$, converges to a marginal distribution as t becomes large enough, usually denoted with W_∞. Therefore, $\mathbb{P}(W(t) \geq B)$ actually denotes the tail properties of the W_∞ distribution.

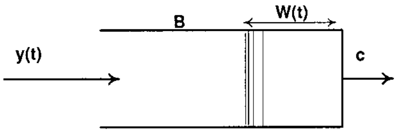


Figure 3.2: A simple buffer of size B, filled at variable rate y_t and emptied at constant rate c.

where the parameter a can be any value larger than c. Let θ^* be selected such that $\alpha(\theta^*) = c$. Then the probability that buffer occupancy reaches B is approximately given by:

$$p = \sum_{a>\alpha(\theta^*)} \exp\{-\frac{B}{a-\alpha(\theta^*)}\Lambda^*(a)\} \tag{3.15}$$

Now we approximate this sum of exponentials with the exponential with the largest exponent, that is, by

$$p \approx \exp\{-\frac{B}{a^*-\alpha(\theta^*)}\Lambda^*(a^*)\} \tag{3.16}$$

where a^* is such that

$$\frac{\Lambda^*(a^*)}{a^*-\alpha(\theta^*)} = \min_{a>\alpha(\theta^*)} \frac{\Lambda^*(a)}{a-\alpha(\theta^*)} \tag{3.17}$$

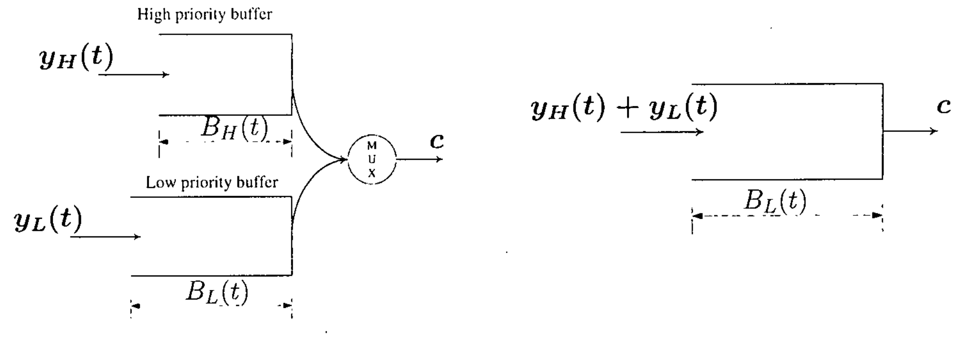
However, from the definition of Λ^* in equation 3.5, it can be noted that $\Lambda^*(a) \geq \theta a - \theta\alpha(\theta)$ for all θ. Hence we have $\min_{a>\alpha(\theta^*)} \frac{\Lambda^*(a)}{a-\alpha(\theta^*)} = \theta^*$ and

$$p \approx e^{-B\theta^*} \tag{3.18}$$

Therefore, one can say that $\alpha(\theta)$ is the rate at which the buffer must be emptied so that the buffer overflow probability decays exponentially with rate θ.

3.3.1 Data Loss Probability in a Queuing Model with Priority

We consider a simple system with two prioritized inputs as shown in Figure 3.3. The inputs are buffered at two different buffers of size B_H and B_L, and are



(a)
A Multiplexer with prioritized inputs

(b)
The equivalent hypothetical buffer for determining the low-priority buffer overflow probability

Figure 3.3: A simple multiplexer model with prioritized inputs.

then multiplexed to one output with constant rate c. The high-priority input has primitive priority over the low-priority input, which means high-priority packets will never be impeded by low-priority packets.

Let $y_H(t)$ and $y_L(t)$ be the rate of the high-priority and low-priority inputs at t respectively. The cumulative arrival functions for these inputs are denoted by Y_H and Y_L, where we have

$$\begin{aligned} Y_H(t) &= \sum_{\tau=0}^{t} y_H(\tau), \qquad \forall t \geq 0 \\ Y_L(t) &= \sum_{\tau=0}^{t} y_L(\tau), \qquad \forall t \geq 0 \end{aligned} \tag{3.19}$$

Also, let $\overline{y}_H$ and $\overline{y}_L$ denote the maximum rate of the input sources. The output rate c is such that $\overline{y}_H \leq c \leq \overline{y}_H + \overline{y}_L$. Therefore, the high-priority buffer will never overflow, while it is possible that the low-priority buffer overflows. Also, c is greater than the average of $y_H + y_L$, which means that the low-priority buffer becomes empty regularly. Let p denote the probability that the low-priority buffer overflows.

Let $W_L(t)$ be the low-priority buffer workload at t. The low-priority buffer overflows if $W_L(t) > B_L$. During $[0, t]$ the high-priority source sends $Y_H(t)$ bits to the system and the system will transmit $c \times t - Y_H(t)$ bits from the low-priority buffer data. Therefore, we have

$$\begin{aligned} W_L(t) &= Y_L(t) - (c \times t - Y_H(t)) \\ &= (Y_L(t) + Y_H(t)) - c \times t \qquad (3.20) \end{aligned}$$

and thus

$$\mathbb{P}(W_L(t) > B_L) = \mathbb{P}((Y_L(t) + Y_H(t)) - c \times t > B_L) \qquad (3.21)$$

The right hand side of this equation can be interpreted as the probability of buffer overflow in a hypothetical single buffer of size B_L, filled at rate $y_L(t) + y_H(t)$ and emptied at rate c. Therefore, in order to study the data loss probability of the low-priority source, we can replace the prioritized model with another model, which has a single hypothetical buffer of size B_L, filled at rate $y_L(t) + y_H(t)$, and emptied at rate c, as shown in Figure 3.3-b. Then, as discussed in Section 3.3, if θ^* is selected such that $\alpha_H(\theta^*) + \alpha_L(\theta^*) = c$, we will have $p = e^{-B\theta^*}$ or $B = -\log(p)/\theta^*$.

If the low-priority input source has a constant input rate R (i.e., $y_L(t) = R$ for all t), then we can find a bound on the time that the low-priority data units wait in the buffer. Let say that W_L data units of the low-priority input are waiting in the buffer, as shown in Figure 3.4. Let $d_L(t)$ denote the time that data unit #1 has being waiting in the buffer. Then, since the buffer fill up rate is constant, we have $d_L(t) = W_L(t)/R$. Therefore, $\mathbb{P}(d_L(t) > d) = \mathbb{P}(W_L(t) > R \times d)$. Note that if the low-priority source is a variable rate source rather than a constant rate source, then this method will not be applicable.

When the low-priority source has constant bitrate, we have

$$\begin{aligned} \mathbb{P}(W_L(t) > B_L) &= \mathbb{P}((Y_L(t) + Y_H(t) - c \times t) > B_L) \qquad (3.22) \\ &= \mathbb{P}((R \times t + Y_H(t) - c \times t) > B_L) \qquad (3.23) \\ &= \mathbb{P}(Y_H(t) - (c - R) \times t > B_L) \qquad (3.24) \end{aligned}$$

which means that $\mathbb{P}(W_L(t) > B_L)$ is equal to probability of loss in a buffer of size B_L, filled with the high-priority source at rate $y_H(t)$ and emptied at constant rate $c - R$.

Now we extend the simple model in Figure 3.3-a to a system with N high-priority inputs and one low-priority input, as shown in Figure 3.5-a. In this model, N high-priority sources are multiplexed with one low-priority

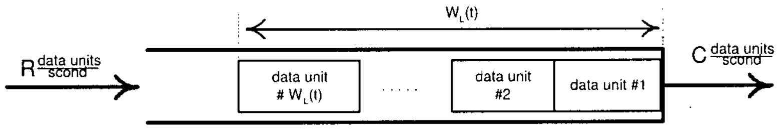
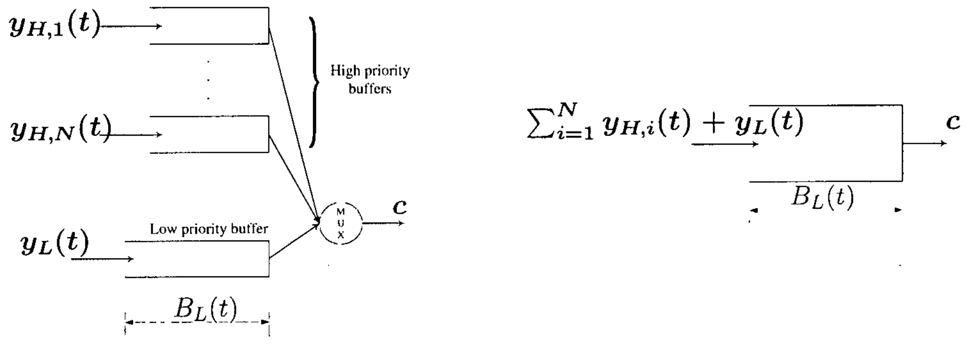


Figure 3.4: A simple buffer filled at constant rate R and emptied at variable rate c. If the buffer workload is W_L, then we know that the first data unit has waited W_L/R seconds in the buffer.

source. The i^{th} high-priority input has the instant rate of $y_{H,i}$, maximum rate of $\overline{y}_{H,i}$, and effective bandwidth of $\alpha_i(\theta)$. The maximum rate of multiplexed stream is c. c is selected such that $\sum_{i=1}^{N} \overline{y}_{H,i} \leq c \leq \overline{y}_{L,i} + \sum_{i=1}^{N} \overline{y}_{H,i}$, therefore the high-priority buffers never overflows, but it is possible for the low-priority buffer to overflow. It is easily noted that the probability of the low-priority buffer overflow is given by $\mathbb{P}(W_L(t) > B_L) = \mathbb{P}((Y_L(t) + \sum_{i=1}^{N} Y_{H,n}(t)) - c \times t > B_L)$, where Y_H^i is the arrival process of the i^{th} high-priority input. As stated in property 4 in Section 3.2.2, the effective bandwidth of a multiplexed source is simply the sum of the effective bandwidths of all the sources. Therefore, the probability of the low-priority buffer overflow is bounded by $e^{-B\theta^*}$, where θ^* is selected such that $\alpha_L + \sum_{i=1}^{N} \alpha_i(\theta^*) = c$. If the low-priority source has a constant rate, then $\alpha_L(\theta) = R$ for all θ; and θ^* is selected such that $\sum_{i=1}^{N} \alpha_i(\theta^*) = c - R$

3.4 Admission Control for the Stochastic Service Class

In this Section, we describe how the effective bandwidth, and the queuing model presented in Section 3.3.1, are adapted to design an admission control test for our interactive TV system. Suppose N television programs are sharing one transmission line with constant capacity c. We assume that each main video stream is characterized by a known effective bandwidth curve, say $\alpha_i(\theta)$ for the i^{th} main video. Now suppose an incidental stream with constant rate R, maximum waiting-time d, and probability of violating the maximum waiting time constraint of p, is requested to be added to the system using the stochastic service class. The multiplexer system can be modelled with the buffering system shown in Figure 3.5. As discussed in Section 3.3.1, the



(a)
A Multiplexer with N high-priority and one low-priority inputs

(b)
The equivalent hypothetical buffer for determining the low-priority buffer overflow probability

Figure 3.5: A simple multiplexer model with N high-priority and one low-priority inputs.

following equations hold

$$
\begin{aligned}
p &= e^{-B\theta^*} \\
B &= R \times d \\
\sum_{i=1}^{N} \alpha_i(\theta^*) &= c - R
\end{aligned}
\tag{3.25}
$$

Hence, given two out of three parameters R, d and p, one can use the equation 3.25 to determine the third un-known parameter. If the triple (R, d, p) conforms to the quality requirements of the connection request, then the connection request is accepted.

3.5 Numerical Estimation of Effective Bandwidth

In this section, we describe the current approaches to estimating the effective bandwidth of a source. Before doing so, it is noted from the definition of effective bandwidth (equation 3.6) that $\alpha(\theta)$ closely depends on the 'moment generating function'[2] of the arrival process of the source. Therefore, to estimate the effective bandwidth, one should estimate the moment generating function or all the generating momentums[3] of the arrival process of the source, which is a very complicated task. Also, note that in many applications such as in the proposed ITV application, the whole effective bandwidth curve, i.e., $\alpha(\theta)$ for all θ, should be estimated. This actually makes the estimation process even harder.

Current approaches to numerical estimation of effective bandwidth are as follows.

1. **Direct Approach** Recall that the effective bandwidth α is defined by

$$\alpha(\theta) = \lim_{t\to\infty} \frac{1}{\theta t} \log \mathbb{E}(e^{\theta Y(t)})$$

Thus, by monitoring the traffic one can use

$$\alpha_{k\times m}(\theta) = \theta^{-1}k^{-1}\log(m^{-1}\sum_{i=1}^{m} e^{\theta(Y(k\times i)-Y(k\times(i-1))})} \tag{3.27}$$

as an estimator for α [43–46]. This approach is attractive since it circumvents modelling the traffic source. However, this approach takes a very long time to converge to an accurate result. According to [43], an accurate estimator of $\alpha(\theta)$ requires that both k and m be large. So, the monitoring time $k \times m$ may in fact be very lengthy.

In [45], a technique called *re-sampling or boot strap* is proposed for this problem. In these approaches some *synthetic* data are generated, which are *similar* to the original data in *'some sense'* [47–49]. These synthetic data are used along the original data, as a data set with a larger sample size, to estimate $\alpha(\theta)$ using equation 3.27. However, this approach introduces two important and problematic issues. First, we should determine

[2] Moment generating function of a random variable X is defined as

$$\psi(\theta) = \mathbb{E}(e^{\theta}X) \tag{3.26}$$

[3] The n^{th} momentum of random variable X is defined as $\mathbb{E}(X^n)$. All the momentums of X can be successively obtained by differentiating $\psi(\theta)$ w.r.t θ and then evaluation at $\theta = 0$, that is $\psi^n(0) = \mathbb{E}(X^n)$, $n \geq 1$.

how the synthetic data are generated. This depends on what kind of *similarity* concept is suitable for our data and application. In the original bootstrap method, which was developed for i.i.d. random variables, the empirical distribution of the original samples $\hat{F}$ is estimated. Then new samples are drawn from this distribution $\hat{F}$. However, this method is not suitable for dependant data. In a method proposed for dependant data, called 'moving block bootstrap', the data is partitioned to blocks of size b, and the same algorithm is then performed on these blocks. However, the disadvantage of this method is that it destroys all the dependency of lags larger than b, hence its is suitable only for short term dependant data. Furthermore, this approach introduces another complication, which is selection of an appropriate block size b. The second issue in using the bootstrap method is to determine how many synthetic data are enough to find an accurate estimate of $\alpha(\theta)$. This question is usually addressed by either using a large sample size, which is thought to be sufficiently large in advance, or by continuing to generate synthetic data sets until the final estimate of $\alpha(\theta)$ does not change by adding a new set of synthetic data. For example, in each iteration we generate a new set of synthetic data and estimate the $\alpha(\theta)$. If this $\alpha(\theta)$ is close to the $\alpha(\theta)$ estimated in the previous iteration by less than a threshold, then this estimate of $\alpha(\theta)$ is accepted. Otherwise, more synthetic data are generated.

In general, this approach is not appropriate for the proposed ITV application, since the sample sizes from the traffic of a typical TV program are not large enough to result in an accurate estimate of $\alpha(\theta)$. Besides, using the bootstrap method is not appropriate, since the current approaches to generate synthetic data will destroy many important characteristic features in our original data.

2. **Virtual Buffer Approach** In this approach, a virtual buffer of size b is considered, which is filled with the source and is emptied at a constant rate c, such that c is between the average rate and peak rate of the source (see [42, 44, 50, 51]). Let $\alpha^{-1}(c)$ be the inverse of the $\alpha(\theta)$ function (i.e., $\alpha^{-1}(c) = \theta \Leftrightarrow \alpha(\theta) = c$). As mentioned in Section 3.3, the probability of buffer overflow can be estimated as $\exp(-b\alpha^{-1}(c))$. Therefore, the buffer workload is monitored in simulation and the empirical distribution of

data loss, $\pi(b)$, is obtained. Then, $\alpha^{-1}(c)$ is chosen so that the distance between $\pi(b)$ and $p(b) = \exp(-b\alpha^{-1}(c))$ is minimized [44]. The distance measure employed is usually the "Kullback-Leibler distance"[4] measure [43, 52]. The estimate for $\alpha^{-1}(c)$ is given by:

$$\hat{\alpha}^{-1}(c) = \log(1 + \frac{\sum_{b=1}^{B_1} \pi(b)}{\sum_{b=B_1}^{\infty} b\pi(b) - B_1 \sum_{b=1}^{B_1} \pi(b)})$$

Alternately, if a sufficient size of data is available, one can plot the logarithm of the loss probability versus b. As shown in [44], this plot has a straight part with slope $\alpha^{-1}(c)$. Hence the slope of this part of this plot can be used as an estimate for $\alpha^{-1}(c)$.

The disadvantage of this method is that the buffer occupancy should be simulated for a long time to obtain an accurate estimate for α. This means that a large sample size is required to simulate the buffer occupancy. Furthermore, this procedure should be repeated for each c to obtain the effective bandwidth curve, i.e., $\alpha(\theta)$ for all θ. On the other hand, this approach is attractive in that it does not assume any signal model for the traffic, which circumvents modelling the traffic.

3. **Model Fitting Approach** This is a two step approach. First, an appropriate parametric model for the source is selected, and its parameters are estimated. Second, the effective bandwidth is numerically computed or obtained from the model parameters. This approach has been carried out successfully in many different applications. An important advantage of this approach is that one can estimate the model parameters in real-time, and use the current estimate of model parameters to update the effective bandwidth estimate online.

   Model fitting itself consists of several steps, including: 1) Model selection: a mathematical parameterized signal model should be selected for the traffic. 2) Model order selection: depending on the selected model, it is usually necessary to select the model size. There are some systematic

[4] Kullback-Leibler distance or the relative entropy of two discrete distribution p and q is defined by

$$d = \sum_k p_k \log_2(\frac{p_k}{q_k}) \qquad (3.28)$$

approaches to this problem, like Akaike information criterion. Selecting a larger model size usually increases the complexity of the model. 3) Estimating the model parameters. For off-line traffic sources, traffic is known a priori and off-line parameter estimation methods can be used. This is usually easier than the online case, where the traffic itself is not know a priori. In online parameter estimation methods, the parameter estimate is usually updated after observing each traffic sample, such that our estimate converges to the actual value of parameters gradually. The online methods are usually able to track the changes in the model parameters.

We employ the traffic modelling approach in the proposed ITV application to find the numerical value of the effective bandwidth curve of a traffic source. This modelling approach comprises three important issues. First, we should select a suitable traffic model, which can capture the important characteristics of TV video traffic. Second, we should find model parameter identification (i.e., model fitting) methods, which can estimate the model parameters from the actual traffic. Finally, we should design a method for obtaining the effective bandwidth from the model parameters. We address these three issues in the next chapter.

3.6 Conclusion

In this chapter, we presented a scheme for the admission control of stochastic service call. Our approach is based on the effective bandwidth theory. We defined the effective bandwidth, and exploited the important characteristics of this concept. Then, we showed how the effective bandwidth concept is used to design an admission control scheme for the stochastic service class. Using the methods presented in this chapter, one can find the maximum-waiting time for an incidental stream whose bitrate is R and data loss probability is $p\%$.

We also discussed current approaches for estimating the numerical value of the effective bandwidth curve from traffic samples of a source. We selected a modelling approach, where a traffic source is modelled using a stochastic model, and effective bandwidth is then obtained from the model parameters. However, we did not indicate which stochastic model shall be used for the video sources in the proposed ITV application. In the next chapter, we address this important problem. That is, we select a stochastic model for modelling the traffic generated by full-screen video sequences of TV programs. We justify our model selection by some evidence from traffic samples of video sequences of actual TV programs. We present methods for estimating model parameters from the traffic samples. We also show how the effective bandwidth curve is obtained from the parameters of the selected stochastic model. Using the methods presented in the next chapter, one can estimate the numerical value of the effective bandwidth curve for a main video source. Hence, the methods presented in this chapter, in conjunction with the methods presented in next chapter, complete the big picture of the admission control mechanism for the stochastic service class.

Chapter 4

Video Traffic Modelling

> *All Models are wrong, but some of them are useful. Modelling should never be an end to itself, but an aid to understand our complex world.*
>
> -George Box

Overview

Stochastic modelling of main video streams traffic is considered, and General hidden Markovian models are discussed for these streams. Specifically, Hidden Semi-Markov Models, HSMMs, are selected for modelling the main video streams traffic. A new formulation for HSMMs is presented, which significantly improves the computational efficiency of HSMM parameter identification algorithms. Based on our new formulation of HSMMs, efficient algorithms for off-line and online identification of an HSMM parameters are presented. Then, it is shown how effective bandwidth curve is obtained from HSMM parameters. Using the methods presented in this chapter, one can estimate the numerical value of the effective bandwidth curve of a video source. The estimated effective bandwidth curve is then utilized in the admission control of incidental streams in the stochastic service class, as discussed in Chapter 3.

4.1 Introduction

In this chapter we seek a stochastic model to characterize the traffic of the main video streams in a TV network. This model should accurately capture the important stochastic characteristics of the traffic. Our motivation is to use this traffic model to find the numerical value of the effective bandwidth curve of the source. Hence, our objectives in this chapter are (1) to select an appropriate stochastic model for video traffic sources, (2) to present a method for estimating the model parameters from traffic (parameter identification), and (3) to obtain the numerical value of the effective bandwidth from the model parameters. As mentioned earlier, this traffic model is then used by the admission control unit to determine how much bandwidth will be available to the incidental streams that use the stochastic service class.

Ideally, a good traffic model for our application should be (a) accurate enough to characterize the important statistical properties of the traffic (b) computationally efficient, and (c) could be used for obtaining the effective bandwidth curve. Note that video traffic demonstrates different characteristics at different time scales. Many of the current modelling efforts strive to capture the rate variability of traffic at the frame level. In those approaches, periodic pattern in frame sizes plays an important role in the traffic shape. This periodic pattern is caused by the frame coding pattern of DCT based video compression standards (e.g., $IBBPBB$ frame coding pattern in each GOP of the MPEG-2 standard). For the proposed ITV application, the traffic model should be able to capture the video rate variability at the *GOP level* (or scene level) rather than at the frame level. This is because the waiting-times in buffers in the proposed ITV application are much larger than the video frame rate, and hence, rate variabilities due to frame patterns are ruled out by buffering.

The rest of this chapter is organized as follows. In Section 4.2, we review current approaches to video traffic modelling. In Section 4.3, we present our modelling approach, which is based on 'Hidden Markov Models', HMMs. In so doing, we discuss the concept of 'state-duration modelling' using the notion of *Semi-Markov* signal models. We show that Hidden Semi-Markov Models, HSMMs, are a better model choice for the proposed ITV application rather than those based on the previously used HMMs. In Section 4.4, we present the theoretical background of HSMMs. As will be discussed, the challenging issue involved in employing HSMMs is that of 'parameter identification'. We address this problem in sections 4.5-4.7. For that effect, we first present a novel signal model for HSMMs in Section 4.5. Based on our new model, we present novel methods for parameter identification of HSMMs for the off-line and online cases in sections 4.6 and 4.7. In Section 4.8, we show how the numerical value of the effective bandwidth curve is obtained from the parameters of an HSMM model.

Finally, we present experimental results of applying the methods developed in this chapter to empirical traffic samples of typical TV programs in Section 4.9.

4.2 Existing Stochastic Video Traffic Models

Current modelling approaches for video traffic modelling (also known as 'source modelling' in literature) can be divided into five main classes:

Renewal Traffic Models Renewal models are mathematically simple and have a long history. These models describe the packets generation at certain points in time. Let A_n be the time between generation of n^{th} and $(n+1)^{th}$ packets, then A_n in a renewal model is identically distributed, but its distribution function is allowed to be general. Poisson and Bernoulli processes are the most popular cases of renewal models in continuous and discrete-time cases respectively. In a Poisson process, A_n is described by an exponential distribution $\mathbb{P}(A_n \leq \tau) = 1 - e^{-\lambda\tau}$, where λ is the average number of packets generated per time unit.

These models have the severe drawback that the autocorrelation function of A_n for any non zero lag is equal to zero. Therefore, these models do not capture any dependency among the A_n's in the past nor in the future.

Autoregressive processes Autoregressive (AR) models estimate the next signal value in a stochastic process as a function of previous signal values. The most important class of these models is the linear autoregressive models, with the form

$$y_t = a_0 + \sum_{i=1}^{N} a_i y_{t-i} + \varepsilon_t \tag{4.1}$$

where y_t is the random variable, a_i's are real constants, and ε_t is an i.i.d. random variable with zero mean. More complicated autoregressive models such as MA (Moving Average), ARMA (Auto Regressive Moving Average) and ARIMA (Auto Regressive Integrated Moving Average) have also being considered for modelling video traffic. The drawback of these models is that they cannot successfully model the marginal distribution of the video traffic [53].

TES models The Transform-Expand-Sample (TES) modelling approach strives to *simultaneously* model both the marginal distribution and the autocorrelation function of an empirical sample set. This means that TES models can capture the first and second order statistical characteristics of a stochastic time-series at the same time.

TES models consist of two classes, called TES$^+$ and TES$^-$, where the plus or minus superscript distinguishes between the cases where the model gives rise to processes with positive or negative lag-1 autocorrelations respectively. TES models consist of two stochastic processes called *background* and *foreground* sequences [53–55]. Background sequences have the form:

$$U_n^+ = \begin{cases} U_0, & n = 0 \\ \langle U_{n-1}^+ + V_n \rangle, & n > 0 \end{cases} \tag{4.2}$$

$$U_n^- = \begin{cases} U_n^+, & n \text{ even} \\ 1 - U_n^-, & n \text{ odd} \end{cases} \tag{4.3}$$

where U_0 is uniformly distributed on $[0,1)$, $\{V_n\}_{n=1}^{\infty}$ is a sequence of i.i.d. random variables, called *innovation sequence*, and the operator $\langle\rangle$ denotes fractional part (also known as 'modulo-1 operator'). It is shown that the marginal distributions of U_n^+ and U_n^- are both uniformly distributed over $[0,1)$, regardless of the distribution of V_n. The foreground sequence is of the form

$$X_n^+ = D(U_n^+) \qquad\qquad X_n^- = D(U_n^-) \tag{4.4}$$

where D is a transformation from $[0,1)$ to the real numbers $\Re$, called *distortion.*

Given an empirical data sample set, TES models select the innovation sequence V_n and the distortion function D, so that both the autocorrelation function and the density function of the foreground process match the autocorrelation and density functions of the empirical data.

The most important family of distortion functions consists of a compound distortion function of the from [54,55]

$$D(x) = \hat{H}_Y^{-1}(S_\xi(x)), \quad x \in [0,1) \tag{4.5}$$

where the inner transform, $S_\xi(x)$, is a 'smoothing' transform called the *stitching transform*, and is parameterized by $0 \leq \xi \leq 1$, and is given by

$$S_\xi(y) = \begin{cases} y/\xi, & 0 \leq y < \xi \\ (1-y)/(1-\xi), & \xi \leq y < 1 \end{cases} \tag{4.6}$$

The outer transformation, $\hat{H}_Y^{-1}$, is the inverse of the empirical distribution function (i.e., histogram) computed from empirical samples $\{Y_n\}_{n=1}^N$.

The rationale for TES modelling approach is based on the following facts. First, all TES background processes are stationary, Markovian, and their marginal distributions is uniform in $[0, 1)$, regardless of distributions of innovation process V_n [54]. Second, the *inversion method* from elementary statistics, allows any uniform variate U on $[0, 1)$ to be transformed to a desired marginal distribution F by applying $F^{-1}(U)$. In particular, TES models use $F = \hat{H}_Y$. Third, for $0 \leq \xi \leq 1$ the stitching transform $S_\xi(U_n)$ preserves the uniformity of U_n; that is, $S_\xi(U_n^+)$ is also uniformly distributed in $[0, 1)$. It is shown that the stitching transform has only a 'smoothing' effect. That is, a sequence $\{S_\xi(U_n^+)\}$ is more 'continuous looking' than an underlying sequence $\{U_n^+\}$. It is shown that this transform preserves uniformity, that is $S_\xi(U_n^+)$ is also uniformly distributed in $[0, 1)$. Hence, it follows that any foreground TES process of the form

$$X_n = \hat{H}_Y^{-1}(S_\xi(U_n)) \tag{4.7}$$

obtained from the background process $\{U_n\}$ is always guaranteed to have the histogram distribution $\hat{H}_Y$, regardless of the selected innovation sequence V_n and the stitching parameter ξ. The choice of the density of the innovation sequence, f_V, determines the autocorrelation function of the foreground process. Thus, TES modelling decouples the fitting of the empirical distribution from the fitting of the empirical autocorrelation function. Since the former is guaranteed, one can concentrate on the latter. In practice, model fitting is carried out by a heuristic search for pairs (ξ, f_V). The search is considered successful if the corresponding TES sequence gives rise to an autocorrelation function that adequately approximates its empirical counterpart [56–58].

Though TES models have been recently used for modelling video traffic with acceptable accuracy, their implementation is too computationally complicated for practical applications. In practice, the heuristic search for (ξ, f_V) relies on a brute-force computation and on selecting the best n combinations of (ξ, f_V) pairs, in the sense that the resulting TES model autocorrelation function minimizes the mean square error with respect to the empirical autocorrelation function. Then, another algorithm is employed to select among the n candidate models the one whose sample path bears the 'most resemblance' to the empirical samples. Furthermore, it is still not known how a TES model can be used for obtaining performance guarantees in a communication network. Specifically, there is no method available for obtaining the effective bandwidth of a traffic source from its TES model

Self-similar or Fractal models These models are based on self-similar process models. The self-similarity concept implies that samples of a process demonstrate similar statistical characteristics when considered at different time scales. The key characterizing parameter is the so-called Hurst parameter, H, which captures the degree of self-similarity in a given empirical signal. Recently, many researches have used self-similar process models for modelling video traffic [12, 53, 59–63]. Most of these researches deal only with the statistical analysis of data sets, including the estimation of the Hurst parameter. These studies provide only limited information about traffic characteristics. Furthermore, very little research has been done on the analysis of communication networks (or queues) with self-similar sources.

Markovian Signal Models Markov signal models have been successfully used for modelling video traffic [6, 9, 53, 64]. One of the popular Markovian models is the two state Markov chain (also called ON/OFF model), where one state represents the peak rate and the other represents the minimum rate. Though this model is simple and is useful for modelling video traffic in some cases, it is not accurate enough for modelling full-screen video sources. The other commonly used model is the Markov Modulated Process, including the Markov Modulated Poisson Process.

Most of the previous research on video modelling study the statistical characteristics of 'video conference' sources and not 'broadcast video' sources. Broadcast video traffic , such as TV video, has different characteristics from those found in video conference applications. Usually, video conference sequences are encoded at a very low bitrate and consist of head and shoulder pictures with little or no camera movements. Hence video conference sequences differ from full-screen television video in two ways: first they consist of very rare scene changes and object movements in the picture; and second the bitrate of video conference streams has little fluctuations when compared to bitrate of full-screen video.

4.2.1 Stochastic Models for Full-Screen Broadcast Video

The usual first step in modelling a real world's stochastic process is through analysis of the system generating the signal. The system that generates video traffic in digital TV applications is an MPEG-2 encoder. We are interested in modelling the amount of traffic that this encoder generates. MPEG-2 encoders use some important encoding parameters, such as number of slices, GOP pattern, GOP length, quantization scale and so on. These parameters affect the quality of the coded video, and are selected according to the application needs. Unfortunately, given the input video sequence and the coding parameters, there is no procedure to obtain the statistical parameters of traffic generated by the encoder. Therefore, it is not useful to incorporate the video coding parameters into the traffic model. However, analysis of MPEG-2 encoding techniques can provide valuable insights into the video traffic shape. One important characteristic of MPEG video traffic is the periodic frame pattern in each GOP. However, in the proposed ITV application, buffering delays for incidental streams (i.e., waiting-time in buffers) are much larger than video frame rates. This means that incidental buffer sizes are much larger than an average incidental video frame, and the periodic frame pattern in each GOP is filtered out during the buffering process. Thus, rate variability caused by the periodic frame pattern does not play any role in the proposed ITV application. Since the buffering delay for an incidental stream in the proposed ITV application is usually more than a couple of seconds, we consider the video source rate at the GOP level rather than the frame level. Note that in most video streams in digital TV application, each GOP contains 15 frames, which results in a GOP length of .5 seconds. Therefore, we seek a traffic model which can capture the *correlation between consecutive GOP sizes* in a video stream.

It is well known that each GOP size depends on the visual content of its corresponding scene. Two features of a video scene that affect its corresponding GOP size are 1) visual details in each frame, and 2) video activities, such as object movements, background movements (e.g., camera panning and zooming), scene changes, etc. In general, simple and non-active scenes result in a small GOP size, while complex scenes result in larger GOP sizes. This inspires us to select a traffic model which can capture the trend in video traffic caused by the underlying scene changes in video, while capturing the rate fluctuations caused by other parameters. 'Hidden Markov Models' (HMMs) are known to match this intuitional requirement. These models consist of a hidden layer and an observable layer. The hidden layer is a Markov chain process, which determines the *state* of the signal. The state of the signal is not observable, and is considered '*hidden*'. One can only observe the output of the observation layer. State of the signal at time t determines the spectral characteristics of the observable layer of the model. That is, statistical characteristics of the observed signal at time t depend on the hidden state of the signal at time t.

Other research studies have confirmed that HMMs are relatively successful in capturing the video traffic characteristics of video conference and broadcast video sources [65–70]. HMMs have also been widely used in many other engineering applications such as speech processing, signal estimation, and queuing networks [71, 72]. The underlying mathematical theory for these models is well established, and efficient algorithms are available for their implementation.

However, HMMs have a limitation. Before we exploit this limitation, we will present a few mathematical preliminaries on the Markov models in the next section. We will then discuss the limitation of HMMs, and use a new model, which can alleviate this limitation.

4.3 Mathematical Background of General Markovian Models

4.3.1 Markov Chain

The simplest form of Markovian signal models is a 'Markov chain'. Consider the discrete-time stochastic process $\{s_t\}$, $t = 1, 2, ...$, which takes its values from the set $\{1, 2, ..., N\}$. The space $\{1, 2, ..., N\}$, is called the state space. If $s_t = i$, then the process is said to be in state i at time t. s_t is a Markov chain if whenever the process is in state i, then the probability that it enters state j in next time-unit is constant. This property is the essence of Markovian signal models and is called *Markovian Property*, and simply states that the probability of transition from a state to another state does not depend on the previous states of the process, that is

$$\mathbb{P}(s_t = j | s_{t-1} = i, s_{t-2} = k, \dots) = \mathbb{P}(s_t = j | s_{t-1} = i) \tag{4.8}$$

Let a_{ij} represent the probability of going from state i to state j, that is

$$a_{ij} = \mathbb{P}(s_t = j | s_{t-1} = i) \tag{4.9}$$

Note that for all i and j, a_{ij} is constant and time-invariant. The matrix $A = [a_{ij}]$ is called the state transition probabilities matrix or state transition matrix. Note that since a_{ij}'s are probabilities and since the process s_t should make a transition to one state at each time instance, we have

$$a_{ij} \geq 0, \qquad 1 \leq i, j \leq N; \quad \sum_{j=1}^{N} a_{ij} = 1, \quad i = 1, ..., N \tag{4.10}$$

Similarly, the n-step transition matrix A^n is defined, where A^n_{ij} represents the probability of being in state j after n state transitions starting in state i.

4.3.2 Hidden Markov Models

Hidden Markov Models, HMMs (also called Markov Modulated Processes, MMPs) are a powerful and widely used class of Markovian signal models. These models are 'doubly stochastic', and consist of a hidden layer and an observable layer. The hidden layer is a Markov chain process s_t, which follows equations 4.8-4.8, and determines the *state* of the signal at t. The state of the signal is not observable, and is considered '*hidden*'. We observe the *observation process* y_t. The spectral characteristics of the observed signal at time t is determined by the state of signal at t. A common model choice for modelling the observable layer is a parametric probability density function (pdf), where the pdf parameters are determined by the current state of the signal. This is denoted by

$$\mathbb{P}(y_t | s_t = i) = b_i(y_t) \tag{4.11}$$

where $b_i(x)$ is a parameterized density function. For example, if $b_i(x)$ is a Normal distribution with mean μ_i and variance σ_i^2, then the probability of observing y_t given that the state process is in state i at t, is

$$\mathbb{P}(y_t|\boldsymbol{s}_t = \boldsymbol{e}_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}}\, e^{-\dfrac{(y_t-\mu_i)^2}{2\sigma_i^2}} \tag{4.12}$$

which only depends on i.

Note that rather than probability density functions, other stochastic models such as Poisson process or TES models, have being considered for modelling the observation layer of HMMs [57].

4.3.3 Limitation of HMM Models

One disadvantage of HMMs lies in their limitation in modelling the *'state duration'* densities. State duration is defined as the time that the state process of an HMM spends in each state before making a transition to another state. Note that in a Markov chain, the probability of remaining in a state is constant. That is, $\mathbb{P}(s_{t+1} = i|s_t = i) = a_{ii}$, where a_{ii} is constant. Therefore, the probability of staying exactly d time units in state i is given by

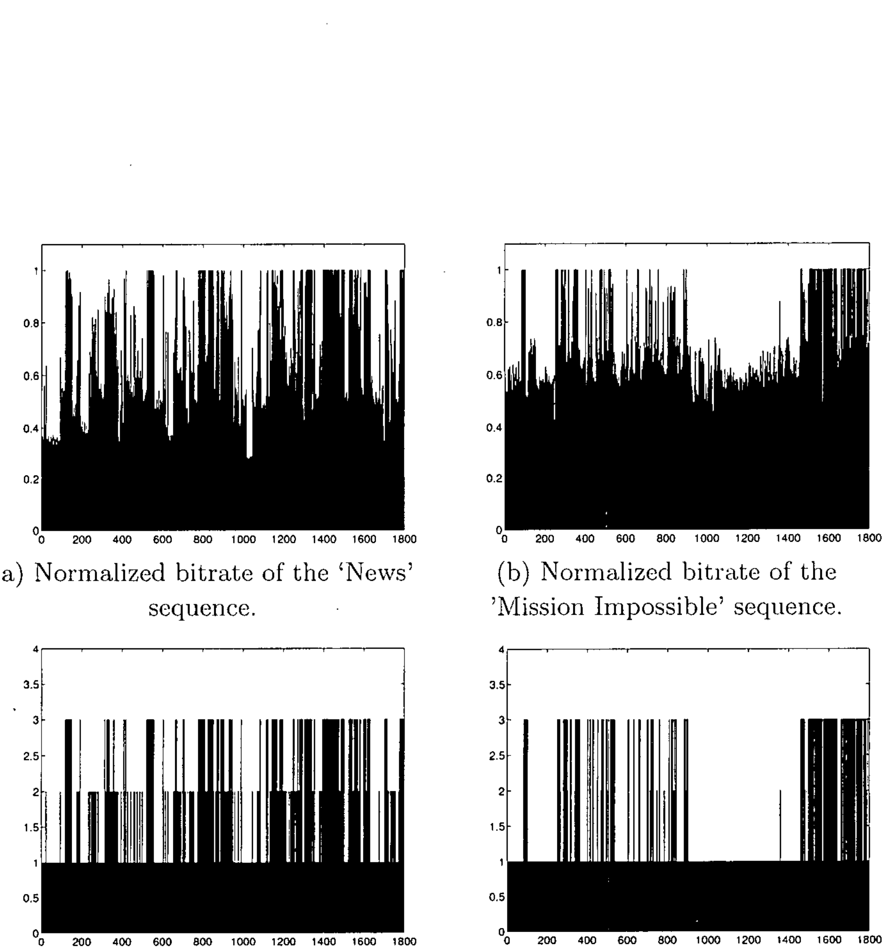
$$p(d) = a_{ii}^{d-1} \cdot (1 - a_{ii}) \tag{4.13}$$

Hence, state duration densities in HMMs are limited to the form given in equation 4.13, which is known as 'Geometric' discrete distribution function in literature. This implies that the probability of staying d time units in each state exponentially goes to zero as d increases. However, the Geometric distribution is not appropriate for the state duration modelling of many physical signals. In order to remove this limitation, a more sophisticated model, called 'Semi-Markov chain,' is used where state duration densities are modelled with some non-Geometric distribution. A hidden Markov model that uses a semi-Markov chain for modelling the hidden state process is called 'Hidden Semi-Markov Model', HSMM. Generally speaking, HSMMs are more powerful in modelling physical signals than HMMs. However, HSMMs are more complicated.

4.3.4 Selecting Between HMM and HSMM Modelling Approach for the Proposed ITV Application

Is HSMM a better model choice for TV video traffic rather than HMM? The answer relies on whether Geometric distribution is a good model choice for state duration of video traffic processes or not. In order to gain an insight to this matter, we conducted two different experiments using empirical bitrate of a few typical TV programs, as discussed below.

The first experiment was based on an empirical approach to estimate the state sequence from the empirical bitrate traces of typical TV programs. In this experiment we first partitioned the video sequences to different scenes. The partitioning method used is similar to that presented in [73], where sudden jumps in the empirical bitrate of video is used to detect scene changes. A manual comparison of detected scene changes and video content showed that this method successfully captures the scene changes from the empirical bitrate record. Then, all scenes were partitioned into three clusters (low, medium and high bitrate) according to the average encoded GOP size of each scene. The clustering algorithm used is presented in [57]. The approach of this clustering algorithm is to partition the given data set, so that elements in each cluster are 'closer' to each other than to elements in all other clusters. We assume that each cluster represents a hidden state of the Markov chain model. According to this classification, we constructed a 'state transition sequence' for each video sequence (see Figure 4.1). Next, we extracted the empirical state-duration traces for each state from the state transition sequence of each video, and calculated the histograms of these state-duration traces. In order to determine if Geometric distribution is a good fit for these empirical state-duration traces, we used both visual histogram comparison and 'Q-Q plot technique' (see appendix B). Figure 4.2-a shows the histogram of one of the state duration for a 'News' video sequence, and 4.2-b shows the typical shape of a Geometric and a Gamma distribution. It is noted that empirical data histogram reveals a curve shape more similar to Gamma distribution than Geometric distribution. Figures 4.2-c and d show the Q-Q plots for Geometric and Gamma distributions. As discussed in appendix B, if the statistical properties of the empirical data match the selected parametric distribution, then the Q-Q plot is expected to be approximately linear with an intercept of 0 and slope 1. It is noted that, the Q-Q plot for Gamma distribution is closer to a line with slope 1 rather than



(a) Normalized bitrate of the 'News' sequence.

(b) Normalized bitrate of the 'Mission Impossible' sequence.

(c) State transition path for 'News' sequence.

(d) State transition path for 'Mission Impossible' sequence.

Figure 4.1: Normalized bitrate and 'State transition path' for two typical TV programs.

the Q-Q plot for Geometric distribution. Hence, *we conclude that a discretized Gamma distribution is a better choice for modelling the state durations of TV video traffic.*

The second experiment was motivated by the results of the first experiment. In this experiment, we used the 'likelihood ratio test' hypothesis testing method (see appendix C) to determine if an HMM model is a better choice for digital TV traffic sources rather than an HSMM model with Gamma state-duration densities. We let $\mathcal{Y}_T = \{y_1, y_2, \cdots, y_T\}$ denote the empirical bitrate trace of the video sequence under study. Then, we test the null hypothesis "$H0$: $\mathcal{Y}_T$ is an HMM" against "$H1$: $\mathcal{Y}_T$ is an HSMM". Details of adopting

Sequence	$-2\ln(\Lambda)$	95% percentile point of χ^2 distribution with $\nu = 3$ degree of freedom	Test result
Mission Impossible II	21.29	7.815	Reject $H0$
News	38.93	7.815	Reject $H0$
Soap Opera	43.17	7.815	Reject $H0$

Table 4.1: Results of likelihood ratio test for different typical TV video sequences. The number of states in HSMM and HMM models is $N = 3$.

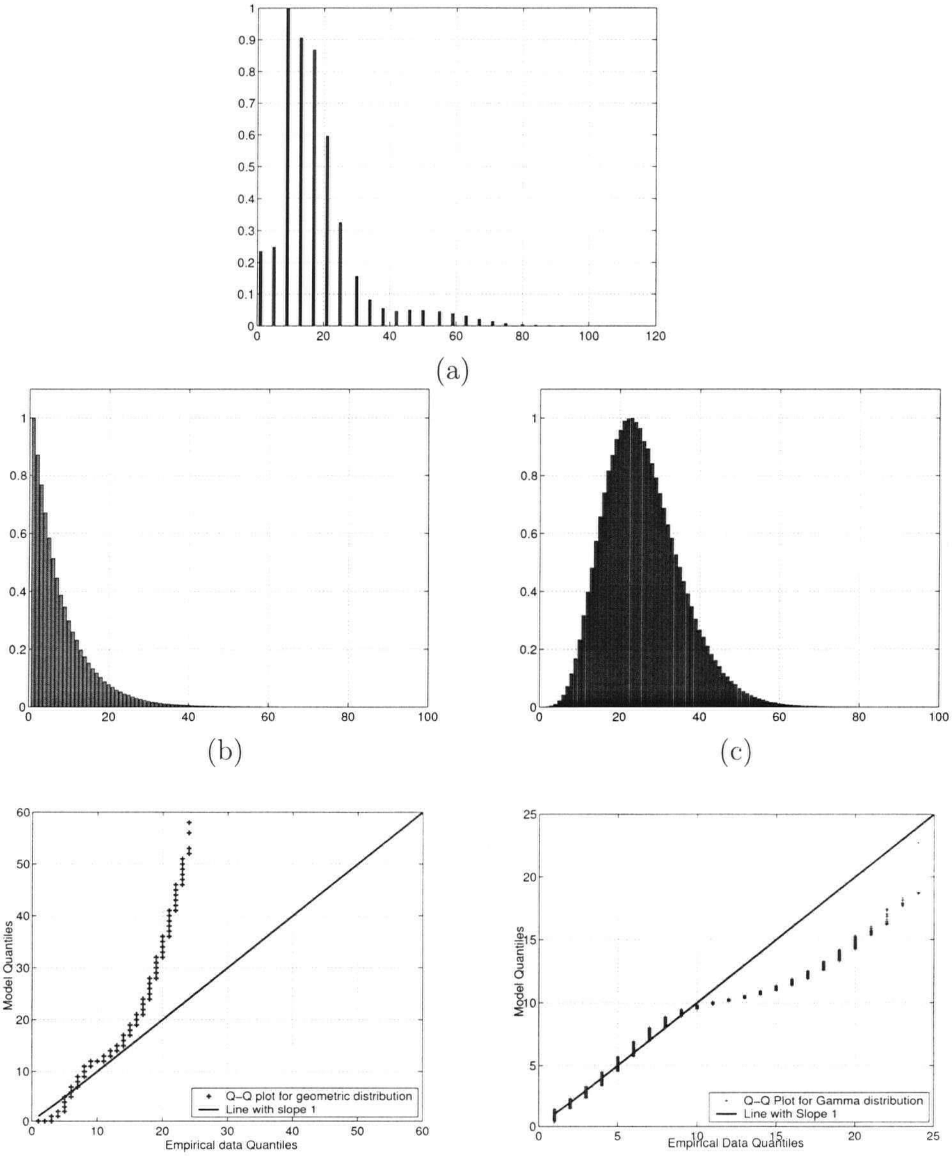
the general likelihood ratio test method for conducting this hypothesis test are given in appendix C. In this test, the likelihood ratio Λ is computed as the likelihood of $\mathcal{Y}_T$ being generated by an HMM model to the likelihood of $\mathcal{Y}_T$ being generated by an HSMM model. It is shown that $-2ln(\Lambda)$ has a χ^2 distribution (see appendix C). Hence, $-2ln(\Lambda)$ is compared to the $100(1-\alpha)$ percentile point of a χ^2 distribution, where α is the significance level of the test. Table 4.1 summarizes the result of this test for the empirical bitrate of a couple of TV programs. The significance level of the test is $\alpha = 5\%$. As shown, for all the sources, $-2ln(\Lambda)$ is greater than the $100(1-\alpha)$ percentile point, which means that the null hypothesis (i.e., '$\mathcal{Y}_T$ is an HMM') should be rejected.

Hence, we choose hidden semi-Markov models for modelling the TV video traffic in the proposed ITV application, where the state durations are modelled with Gamma distribution. In the next section, we present the mathematical formulation of HSMMs, and address the important issues raised in employing HSMMs. In Section 4.9, we present the result of fitting an HSMM model to empirical bitrate traces of typical TV programs.

4.4 General Background on Semi-Markovian Signal Models

4.4.1 Semi-Markov Chains

The discrete-time stochastic process $\{s_t\}$, $t = 1, 2, ...,$ which takes its values from the state space $\{1, 2, ..., N\}$ is a Semi-Markov chain if the next state of signal depends only on the current state and the amount of time that signal



(a)

(b)

(c)

(d)

(e)

Figure 4.2: a) Histogram of state (or cluster) duration for one of the states in the News Sequence. b) Geometrical probability mass function. c) Discretized Gamma probability distribution function. d) Q-Q plot for Geometrical distribution. e) Q-Q plot for Gamma distribution.

has spend in the current state. Formally, this condition is stated as

$$\mathbb{P}(s_t = j | s_{t-1} = s_{t-2} = \cdots = s_{t-d} = i, s_{t-d-1} = k, \cdots) = \mathbb{P}(s_t = j | s_{t-1} = i, d) \tag{4.14}$$

where $i \neq k$. $A = [a_{ij}]$ is known as the state transition probabilities matrix, where a_{ij} is the probably of going to state j from state i, knowing that the signal is leaving state i.

$$a_{ij} = \mathbb{P}(s_t = j | s_{t-1} = i, s_t \neq i) \tag{4.15}$$

Note a_{ij}'s are all constant, and are constrained to

$$a_{ij} \geq 0, \qquad 1 \leq i,j \leq N; \quad \sum_{j=1}^{N} a_{ij} = 1, \quad i = 1, ..., N \tag{4.16}$$

All a_{ii}'s are zero.

In a Semi-Markov chain, the state duration densities are modelled in some non-Geometrical form, unlike that of equation 4.13. This non-Geometrical from is usually denoted by a state duration probability mass function $\varphi_i(d)$, where $\varphi_i(d)$ is the probability that the signal stays exactly d time units in state i during its visits to this state. $\varphi_i(d)$ may be selected to be one of the known parameterized probability mass functions, such as Poisson, Binomial and so on. Alternately, $\varphi_i(d)$ may be selected as a discretized probability distribution function, such as Normal or Gamma pdf.

The signal generation process of a Semi-Markov chain s_t can be summarized as follows.

1. Start with a given initial state, e.g., $s_t = i$ for $t = 1$.

2. Select a duration d according to the state-duration density function of the i^{th} state, $\varphi_i(d)$.

3. For the next d time units, stay at the same state i.

4. Select the next state according to a constant state transition matrix $A = [a_{ij}]$, with the constraint that the signal should leave the current state (i.e., $a_{ii} = 0$).

5. Go back to step 2.

4.4.2 Hidden Semi-Markov Models

A Hidden Semi-Markov Model, HSMM, (also known as Semi-Markov Modulated Process, SMMP) is similar to an HMM except, that the hidden state process is a semi-Markov chain. Generally speaking, HSMMs are a generalization of HMMs and are more powerful in modelling physical signals. However, HSMMs are more complicated than HMMs.

The most important problem that arises in using HSMMs is the *identification* of the model parameters. This identification problem is studied either in off-line or online cases. In off-line case, given a set of observations from an HSMM signal, $\mathcal{Y}_T = \{y_1, y_2, \ldots, y_T\}$, one should find the parameters of the HSMM model, denoted by $\boldsymbol{\theta}$. In online case, one observes a signal sample y_t at time t, and should update the current estimate of the model parameters $\boldsymbol{\theta}_t$ such that $\boldsymbol{\theta}_t$ gradually converges to the actual model parameters. In the rest of this Chapter, we address the identification of HSMMs in both off-line and online cases, as they are both necessary in implementation of the proposed interactive TV system. Off-line identification methods are necessary for estimating the model parameters of pre-recorded video streams. Online identification methods are employed for online model parameter estimation from live video sources. Ultimately, the estimated model parameters are used to find the effective bandwidth curve of sources.

4.4.3 General Background on Identification of HSMMs

Identification of HSMMs is conceptually similar to identification of HMMs; and current approaches to the identification of HSMMs constitute a generalization of the identification methods for HMMs. There are powerful methods available for identification of HMMs. Off-line identification of HMMs is based on an iterative algorithm, known as Baum-Welch[1] or *Expectation Maximization, EM* algorithm. This algorithm finds the maximum likelihood estimate of model parameters $\boldsymbol{\theta}$. That is, $\boldsymbol{\theta}$ is estimated so that $\mathbb{P}(\mathcal{Y}_T|\boldsymbol{\theta})$ is maximized. It is shown that the maximum likelihood estimate converges to the true parameter value as the sample size T increases.

The EM algorithm consists of two steps in each iteration; the **E** or the *Expectation* step and the **M** or the *Maximization* step. The algorithm starts

[1] The EM method method refers to a general class of approaches. The Baum-Welch algorithm is a variant of the EM algorithm for estimating the HMM parameters.

with an initial estimate of model parameters $\boldsymbol{\theta}$. Note that a part of an HMM model is hidden from us, and that is the state transition path of the underlying Markov chain, denoted by $\mathbf{S}_T = \{s_1, s_2, \cdots, s_T\}$. In the **E** step the "optimal" state transition path $\mathbf{S}_T$ is estimated from samples $\mathcal{Y}_T$ and current parameter estimates $\boldsymbol{\theta}$. This is done by estimating a set of so called *forward* and *backward* variables, $\boldsymbol{\alpha}_t$'s and $\boldsymbol{\beta}_t$'s. The forward parameters, $\boldsymbol{\alpha}_t$'s, are computed by induction from $\boldsymbol{\alpha}_{t-1}$. Similarly, the backward parameters, $\boldsymbol{\beta}_t$'s, are computed from $\boldsymbol{\beta}_{t+1}$. In the **M** step, the model parameters are re-estimated by finding the MLE estimate of $\boldsymbol{\theta}$ from the computed forward-backward parameters and $\mathcal{Y}_T$. The **E** and **M** steps are iterated until $\boldsymbol{\theta}$ converges.

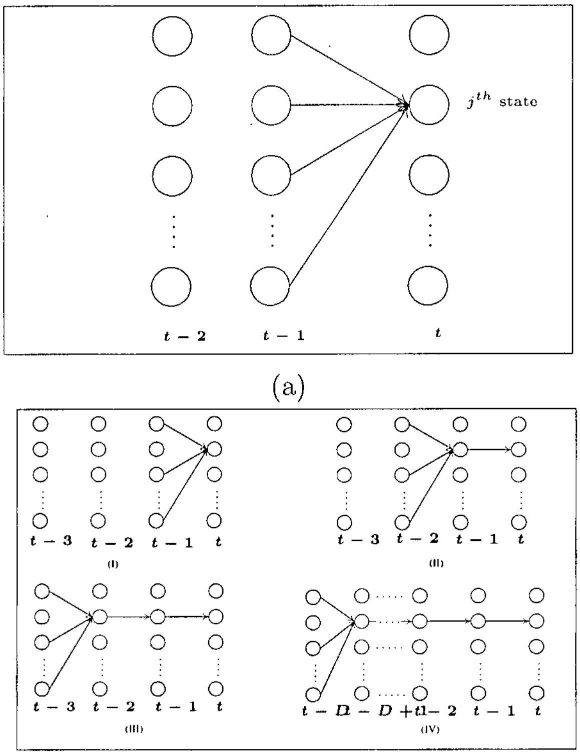
The EM algorithm has been extended to the context of HSMMs by using 'explicit state duration modelling' [72,74–76] or 'parametric state duration modelling' [77]. The first approach estimates the density of the state durations for all possible values explicitly, while the second approach uses a parametric distribution to model the state duration, and only estimates the model parameters. However, current methods for identification of HSMMs, which are based on these two approaches, have the major drawback of greatly increased computational load compared to the HMM case. This increase in the computational load lies in the estimation formulas for forward-backward variables in the **E** step, where $\boldsymbol{\alpha}_t$ ($\boldsymbol{\beta}_t$) should be computed from $\{\boldsymbol{\alpha}_{t-1}, \boldsymbol{\alpha}_{t-2}, \cdots, \boldsymbol{\alpha}_{t-D}\}$ (and from $\{\boldsymbol{\beta}_{t+1}, \boldsymbol{\beta}_{t+2}, \cdots, \boldsymbol{\beta}_{t+D}\}$), instead of $\boldsymbol{\alpha}_{t-1}$ ($\boldsymbol{\beta}_{t+1}$) in HMMs, where D is the maximum allowable state duration. More precisely, in EM algorithm for identification of HMMs, the forward variables $\alpha_t(j)$ are computed by induction from α_{t-1}'s using

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) \cdot a_{ij}\right] b_j(\mathbf{o}_t), \qquad \begin{array}{l} 1 \leq t \leq T \\ 1 \leq j \leq N \end{array} \tag{4.17}$$

In this equation $\alpha_t(j)$ is obtained by adding up the probability that s_t has traversed one of the state transition paths shown in Figure 4.3-a. While in identification of HSMMs, $\alpha_t(j)$ is computed as

$$\sum_{d=1}^{D} \sum_{i=1}^{N} \left(\alpha_{t-d}(i) a_{ij} p_i(d) \prod_{s=t-d+1}^{t} b_i(y_s) \right) \tag{4.18}$$

That is, the probability of being in state j at t is computed by summing the probability of the paths shown for $d = 1, 2, 3, \cdots, D$ in Figure 4.3-b. Similar equations are used for computing β_t's.



(a)

(b)

Figure 4.3: State transition paths that should be considered for computing $\alpha_t(j)$ in the HMM and HSMM cases.

It is shown that current approaches to identification of HSMMs using the EM algorithm increase the memory usage by the factor D times and the computation load by $D^2/2$ times, when compared to the EM algorithm for HMMs. Since D is usually large in many applications(e.g., $D = 25$ in most speech processing applications), these algorithms become impractical.

In [78], an HSMM with N states and maximum state duration D is reformulated as an HMM with ND states, then the standard EM algorithm is used to estimate the model parameters. There are other approaches, which are based on the 'state duration dependant transition probabilities' [79]. In these approaches, the state transition matrix is time-varying or is replaced with a tensor. The drawback of the methods presented in [78, 79] approaches is addition of a large number of parameters into the model. These extra parameters should be estimated in addition to the usual HMM parameters, and this requires a large sample size in order to obtain an accurate estimate.

Online identification of HSMMs is conceptually harder than the off-line case. Currently, there are no available methods for on-line identification of HSMMs in the literature. However, the on-line identification of HMMs have been studied in [71, 80–83]. These approaches are based on either the 'recursive maximum likelihood', (RML), or the 'recursive prediction error', (RPE), techniques. In the RML approach, the current estimate of model parameters vector, $\boldsymbol{\theta}_t$, is updated at each time instance in such a direction that the 'Kullback-Leibler' (KL) information measure is maximized. It is shown that this results in the maximum likelihood estimate of the model parameters as $t \longrightarrow \infty$.

The RPE method is a class of general numerical parameter estimation method. RPE algorithms are, in essence, 'Extended Kalman filters' (EKF). More precisely, RPE methods are a special class of Extended Kalman filters for the case when the unknown constant parameters of the model are viewed, and estimated, as states. In this approach a norm $V(\boldsymbol{\theta})$ that measures the prediction error of the model is defined. The model parameters are updated according to

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda \cdot f^{(t)} \tag{4.19}$$

where $f^{(t)}$ is a search direction based on some information about $V(\boldsymbol{\theta})$ acquired at previous iterations, and λ is a positive step size. There are different algorithms based on equation 4.60 which use different choices for the search direction f. The most important class of these approaches is the *quasi-Newton methods*, where the direction function f is

$$f^{(t)} = -[V''(\boldsymbol{\theta}_t)]^{-1} \cdot V'(\boldsymbol{\theta}_t) \tag{4.20}$$

The mathematical theory for general RPE technique is presented in [84]. In [71, 82], a formulation of HMMs is presented, which allows the general RPE scheme to be applied to on-line identification of the HMMs. The norm function used in this method is a weighted quadratic prediction error criterion:

$$V(\boldsymbol{\theta}) = \gamma_t \sum_{k=1}^{T} \beta(t,k)\epsilon^2(k,\boldsymbol{\theta}) \tag{4.21}$$

where $\epsilon(k, \boldsymbol{\theta}) = \hat{y}(k, \boldsymbol{\theta}) - y(k)$ is the prediction error, γ_t is a normalizing factor and $\beta(t, k)$ is a weighting sequence, introduced to increase the effect of recent observations on the estimate. In [83], an algorithm based on the RPE technique is presented, which finds the MLE of the model parameters, and hence, can be considered as an RML algorithm.

In summary, the existing off-line parameter identification methods for HSMM's are very computationally demanding, and none online parameter identification method have been suggested. In the next section of this chapter, we will present new methods for identification of HSMMs for both off-line and online cases. Our methods are based on a new formulation of HSMMs. In our new signal model, we introduce a new variable to the traditional HSMMs, named the 'state duration' variable, $\boldsymbol{d}_t$. $\boldsymbol{d}_t$ is actually a vector, where $d_t(i)$ denotes the time that the signal has spent in state i, given that the state at time t is i. We then use state duration dependent transition probabilities, where the probability of transition from state i to j is not constant and depends on $d_t(i)$. Hence, in our model, the probability of being in any state at time t depends on the state at $t-1$ as well as $\boldsymbol{d}_{t-1}$. We model the state duration densities with parameterized probability density functions.

We then present a novel version of the EM algorithm for off-line identification of HSMMs based on our model. Our algorithm finds the local maximum likelihood estimate of the model parameters. The major advantage of our method over current ones is that it has almost the same computational complexity as the EM algorithm for the HMMs, and hence is useful for a larger set of practical applications.

We also present a sophisticated method for online identification of HSMMs, which is based on our proposed signal model. Our approach adaptively updates parameter estimates, so that the log-likelihood of model parameters is maximized. We use an estimate of the *Newton-Raphson* direction as the update direction of our model parameters. This choice results in a near optimum convergence rate in our algorithm. In this regard, our method is similar to the online identification algorithm for HMMs in [82].

4.5 New Formulation of HSMMs

4.5.1 Hidden (or Modulating) Layer Modelling

We consider a signal model where the state of the signal at time t, $\boldsymbol{s}_t$, $t \in \mathbb{N}$, is determined by a finite-state discrete-time semi-Markov chain. We assume the initial state $\boldsymbol{s}_1$ is given or its distribution is known. The state space has N distinct states. Without loss of generality, we assume $\boldsymbol{s}_t$ takes its values from the set $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N\}$, where $\boldsymbol{e}_i$ is a $N \times 1$ vector with unity as the i^{th} element and zeros elsewhere. If $\boldsymbol{s}_t = \boldsymbol{e}_i$, we say the signal is in state i at time t. The semi-Markov property of the model implies that the probability of a transition from state $\boldsymbol{e}_j$ to $\boldsymbol{e}_i$, $j \neq i$, at time t depends on the duration spent in state $\boldsymbol{e}_j$ prior to time t. This can be written as

$$\mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_i | \boldsymbol{s}_t = \boldsymbol{e}_j, \boldsymbol{s}_{t-1} = \boldsymbol{e}_k, \cdots, \boldsymbol{s}_1 = \boldsymbol{e}_l) = \mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_i | \boldsymbol{s}_t = \boldsymbol{e}_j, d_t(j)) \tag{4.22}$$

where $d_t(j)$ is the duration spent in the j^{th} state prior to time t, that is

$$d_t(j) = \{d | \boldsymbol{s}_{t-1} = \boldsymbol{e}_j, \cdots, \boldsymbol{s}_{t-d+1} = \boldsymbol{e}_j, \boldsymbol{s}_{t-d} \neq \boldsymbol{e}_j\} \tag{4.23}$$

For each time t, we define the *'state duration'* vector $\boldsymbol{d}_t$ of size $N \times 1$ as

$$\boldsymbol{d}_t = \begin{cases} d_t(j) & \text{if } \boldsymbol{s}_t = \boldsymbol{e}_j \\ 1 & \text{if } \boldsymbol{s}_t \neq \boldsymbol{e}_j \end{cases} \tag{4.24}$$

$d_t(j)$ is easily constructed from $d_{t-1}(j)$ as $d_t(j) = s_t(j) \times d_{t-1}(j) + 1$, which can be written in vector format as

$$\boldsymbol{d}_t = \boldsymbol{s}_t \odot \boldsymbol{d}_{t-1} + 1 \tag{4.25}$$

where $\odot$ denotes element-by-element product.

We model the state duration densities with a parametric *probability mass function*, pmf, $\phi_i(d)$. This means the probability that $\boldsymbol{s}_t$ stays exactly for d time units in state i is given by $\phi_i(d)$. $\phi_i(d)$ should be selected so that it adequately captures the properties of the signal under study. Hence, selection of $\phi_i(d)$ should be justified by some evidence from samples of the signal. Though HMM state durations are inherently discrete, it is noted in many studies that continuous parametric density functions are also suitable for modelling state durations in many applications, including speech processing [76, 77]. In this approach state durations are modelled with the best fitting parametric *probability density function*, pdf, and then the discrete counterpart of this density function is taken as the best pmf. That is, if $\phi_i(x)$ is the continuous pdf for state duration of i^{th} state, then the probability that the signal stays in state i for exactly d time units is given by $\int_{d-1}^{d} \phi_i(x)dx$. Since negative state durations are not physically meaningful, it is usually more appropriate to select $\phi_i(x)$ from the family of exponential distributions [76]. Specifically, the family of Gamma distributions are considered in [77] for speech processing applications. In this paper, we assume that $\phi_i(x)$ is a *Gamma* distribution function with *shape parameter* ν_i and *scale parameter* η_i, that is

$$\phi_i(d) = \frac{\eta_i^{\nu_i}}{\Gamma(\nu_i)} d^{\nu_i - 1} e^{-\eta_i d} \qquad (0 < d < \infty) \tag{4.26}$$

where Γ is the gamma function. The mean and variance of ϕ_i are ν_i/η_i and ν_i/η_i^2 respectively (see [85]). Note that our signal model presented here is applicable with minor changes, if a pdf other than Gamma is selected. Furthermore, let $\Phi_i(x)$ denote the cumulative probability distribution function of $\phi_i(x)$, i.e.,

$$\begin{aligned} \Phi_i(d) &= \mathbb{P}(\boldsymbol{s}_t \text{ stays in state } i \text{ for at most } d \text{ time units}) && (4.27) \\ &= \int_0^d \phi_i(x)dx && (4.28) \end{aligned}$$

We construct our model for HSMMs using state duration dependant transition probabilities. We define the state transition matrix $\boldsymbol{A}_{\boldsymbol{d}_t}$, as $\boldsymbol{A}_{\boldsymbol{d}_t} = [a_{ij}(\boldsymbol{d}_t)]$ where $a_{ij}(\boldsymbol{d}_t) = \mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_j | \boldsymbol{s}_t = \boldsymbol{e}_i, d_t(i))$. Clearly, $a_{ij}(\boldsymbol{d}_t)$'s are not constant and change in time; however, we will denote $a_{ij}(\boldsymbol{d}_t)$ with a_{ij} for notational simplicity. The diagonal elements of $\boldsymbol{A}_{\boldsymbol{d}_t}$, a_{ii}'s, are the probability of staying in state i knowing that $\boldsymbol{s}_t$ has been in state i for $d_t(i)$ time units.

$$\begin{aligned} a_{ii} &= \mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_i | \boldsymbol{s}_t = \boldsymbol{e}_i, d_t(i)) \\ &= \mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_i | \boldsymbol{s}_t = \boldsymbol{e}_i, \boldsymbol{s}_{t-1} = \boldsymbol{e}_i, \ldots \boldsymbol{s}_{t-d_t(i)+1} = \boldsymbol{e}_i, \boldsymbol{s}_{t-d_t(i)} \neq \boldsymbol{e}_i) \\ &= \frac{\mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_i, \boldsymbol{s}_t = \boldsymbol{e}_i, \ldots, \boldsymbol{s}_{t-d_t(i)+2} = \boldsymbol{e}_i | \boldsymbol{s}_{t-d_t(i)+1} = \boldsymbol{e}_i, \boldsymbol{s}_{t-d_t(i)} \neq \boldsymbol{e}_i)}{\mathbb{P}(\boldsymbol{s}_t = \boldsymbol{e}_i, \boldsymbol{s}_{t-1} = \boldsymbol{e}_i, \ldots, \boldsymbol{s}_{t-d_t(i)+1} = \boldsymbol{e}_i | \boldsymbol{s}_{t-d_t(i)+1} = \boldsymbol{e}_i, \boldsymbol{s}_{t-d_t(i)} \neq \boldsymbol{e}_i)} \end{aligned} \tag{4.29}$$

Meanwhile, the denominator of equation 4.29 can be written as $\sum_{k=1}^{\infty} \mathbb{P}(\boldsymbol{s}_{t+k} \neq \boldsymbol{e}_i, \boldsymbol{s}_{t+k-1} = \boldsymbol{e}_i, \ldots, \boldsymbol{s}_{t-d_t(i)+2} = \boldsymbol{e}_i | \boldsymbol{s}_{t-d_t(i)+1} = \boldsymbol{e}_i, \boldsymbol{s}_{t-d_t(i)} \neq \boldsymbol{e}_i)$ or $\sum_{k=1}^{\infty} \mathbb{P}(\boldsymbol{s}_t$ stays in state i for exactly $d_t(i) - 1 + k$ time units), which is $1 - \Phi_i(d_t(i) - 1)$. Therefore, equation 4.29 can be written as

$$a_{ii}(d_t(i)) = \frac{1 - \Phi_i(d_t(i))}{1 - \Phi_i(d_t(i) - 1)} \tag{4.30}$$

The probability that the state process stays in the i^{th} state during its visit to this state for exactly d time units is given by $(1 - a_{ii}(d)) \cdot \prod_{k=1}^{d-1} a_{ii}(k)$. By substituting a_{ii} from 4.30, it is easily shown that the probability density function of the state space durations is actually equal to the selected model $\phi_i(d)$.

For $i \neq j$, a_{ij} is the probability of leaving state i and entering state j, and is given by

$$\begin{aligned} a_{ij} &= \mathbb{P}(\boldsymbol{s}_{t+1} \neq \boldsymbol{e}_i | \boldsymbol{s}_t = \boldsymbol{e}_i, d_t(i)) \times \mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_j | \boldsymbol{s}_t = \boldsymbol{e}_i, i \neq j) \\ &= (1 - a_{ii}) \cdot a_{ij}^o \end{aligned} \tag{4.31}$$

where $a_{ij}^o = \mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_j | \boldsymbol{s}_t = \boldsymbol{e}_i, i \neq j)$ is defined as the probability of transition from state i to state j, knowing that the signal leaves state i. We write the matrix $\boldsymbol{A}_{\boldsymbol{d}_t}$ in terms of a diagonal matrix $\boldsymbol{P}(\boldsymbol{d}_t)$ representing the recurrent state transition probabilities, and a constant matrix $\boldsymbol{A}^o$ representing the non-recurrent state transition probabilities.

$$\boldsymbol{A}_{\boldsymbol{d}_t} = \boldsymbol{P}(d_t) + (\boldsymbol{I} - \boldsymbol{P}(d_t))\boldsymbol{A}^o \tag{4.32}$$

$$p_{ij}(\boldsymbol{d}_t) := \begin{cases} 0 & ,i \neq j \\ \dfrac{1 - \Phi_i(d_t(i))}{1 - \Phi_i(d_t(i) - 1)} & ,i = j \end{cases} \tag{4.33}$$

$$a_{ij}^o := \begin{cases} 0 & ,i = j \\ \mathbb{P}(\boldsymbol{s}_{t+1} = j | \boldsymbol{s}_t = i) & ,i \neq j \end{cases} \tag{4.34}$$

Note that a_{ij}^o are constrained to $\sum_{j=1}^{N} a_{ij}^o = 1$. Since $\boldsymbol{P}(\boldsymbol{d}_t)$ is a diagonal matrix, one can show that $\sum_{j=1}^{N} a_{ij}(\boldsymbol{d}_t) = 1$ for all t.

Hence, the hidden state process $\boldsymbol{s}_t$ evolves in time based on the following equations:

$$\begin{aligned} \boldsymbol{s}_{t+1} &= \boldsymbol{A}_{d_t} \cdot \boldsymbol{s}_t + \boldsymbol{v}_{t+1} \\ \boldsymbol{A}_{d_t} &= \boldsymbol{P}(\boldsymbol{d}_t) + (\boldsymbol{I} - \boldsymbol{P}(\boldsymbol{d}_t)) \cdot \boldsymbol{A}^o \\ \boldsymbol{d}_{t+1} &= \boldsymbol{s}_{t+1} \odot \boldsymbol{d}_t + 1 \end{aligned} \tag{4.35}$$

where $\boldsymbol{v}_{t+1}$ is a martingale increment; that is, $\mathbb{E}(\boldsymbol{v}_{t+1} | \boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_t) = 0$.

REMARK. Our modelling scenario can be encapsulated as a time homogeneous first-order Markov model. Consider the 2-vector process $\boldsymbol{S}_t$ as $\boldsymbol{S}_t = (\boldsymbol{s}_t, \boldsymbol{d}_t)$. $\boldsymbol{s}_t$ takes its values from the finite set $\{\boldsymbol{e}_i | i = 1, 2, \cdots, N\}$, and $\boldsymbol{d}_t$ takes its values from the infinite set $\{\boldsymbol{f}_i^d | i = 1, 2, \cdots, N; d = 1, 2, \cdots\}$, where $\boldsymbol{f}_i^d$ is a $N \times 1$ vector with d as the i^{th} element and unity elsewhere. According to equation 4.25, $\boldsymbol{d}_{t+1}$ depends only on $\boldsymbol{s}_{t+1}$, $\boldsymbol{s}_t$ and $\boldsymbol{d}_t$, and according to equation 4.22, $\boldsymbol{s}_{t+1}$ depends only on $\boldsymbol{s}_t$ and $\boldsymbol{d}_t$. Therefore, $\mathbb{P}(\boldsymbol{S}_{t+1} | \boldsymbol{S}_t)$ is independent of t, and hence $\boldsymbol{S}_t$ is a homogeneous first-order infinite states Markov chain.

4.5.2 Observation Layer Modelling

The state process $\boldsymbol{s}_t$ is hidden and is not observed. We observe the *observation process* y_t, where the probabilistic distribution of y_t is determined by state at time t, i.e., $\boldsymbol{s}_t$. In this paper, we assume that for each state i, y_t has a normal distribution. That is, if $\boldsymbol{s}_t = \boldsymbol{e}_i$ then $\mathbb{P}(y_t | \boldsymbol{s}_t = \boldsymbol{e}_i) = \mathcal{N}(y_t; \mu_i, \sigma_i^2)$, where μ_i and σ_i^2 are the mean and standard deviation of the observation process y_t for state i. We denote the probability of observing y_t in state i with $b_i(y_t)$ throughout this paper, that is

$$b_i(y_t) = \mathbb{P}(y_t | \boldsymbol{s}_t = \boldsymbol{e}_i) \tag{4.36}$$

Therefore, y_t may be written as

$$y_t = \langle \boldsymbol{\mu}, \boldsymbol{s}_t \rangle + \langle \sqrt{\boldsymbol{\sigma}^2}, \boldsymbol{s}_t \rangle w_t \tag{4.37}$$

where $\boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_N]'$, $\boldsymbol{\sigma}^2 = [\sigma_1^2, \sigma_2^2, \cdots, \sigma_N^2]'$, $\langle ., . \rangle$ denotes inner product and w_t is Gaussian white noise with zero mean and variance 1 .

Equations 4.35 and 4.37 define the signal generation process in our signal model. Figure 4.4 summarizes and compares the signal generation process in our model with that of other HSMM signal models.

1. Start with an initial state, e.g., $\boldsymbol{s}_t = e_i$ for $t = 1$.
2. Initialize $\boldsymbol{d}_t = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}'$
3. Generate y_t based on the density function of the current state.
4. Compute $\boldsymbol{A}_{\boldsymbol{d}_t}$ (equation 4.32), and choose a new state s_{t+1} based on $\boldsymbol{A}_{\boldsymbol{d}_t}$.
5. Update the state duration variable, $\boldsymbol{d}_t$, using equation 4.25.
6. Go to step 3.

a) Our HSMM model.

1. Start with an initial state, e.g., $\boldsymbol{s}_t = e_i$ for $t = 1$.
2. Select a duration d according to the i^{th} state duration density function.
3. For the next d time units, stay at state $\boldsymbol{s}_t$, and generate y_t based on the densities of state $\boldsymbol{s}_t$.
4. Select the next state according to a constant state transition matrix $\boldsymbol{A}$, with the constraint that the signal should leave the current state.
5. Go to step 2.

b) Traditional HSMM model.

Figure 4.4: Comparison of the signal generation process in our model with that of existing signal models for HSMMs.

4.5.3 Model Parameterizations

There are $N^2 + 3N$ parameters that define an HSMM signal using our model. These parameters are $N^2 - N$ non-recurrent transition probabilities a^o_{ij}, the mean and variance of the observation process μ_i and σ_i^2 for $1 \leq i \leq N$, and the ν_i and η_i parameters of the gamma distribution of the state-durations for

$1 \leq i \leq N$. We define $\boldsymbol{\theta}$ as a vector containing all the model parameters.

$$\begin{aligned}\boldsymbol{\theta} = \big(&\mu_1, \mu_2, \cdots, \mu_N, \sigma_1^2, \sigma_2^2, \cdots, \sigma_N^2, a_{12}^o, a_{13}^o, \cdots, a_{N-1,N}^o, \\ &\nu_1, \nu_2, \cdots, \nu_N, \eta_1, \eta_2, \cdots, \eta_N\big)'\end{aligned} \tag{4.38}$$

4.6 A New Algorithm for Off-line Identification of HSMMs

In this section, we present a new algorithm for off-line identification of HSMMs. Our algorithm is based on the signal generation model presented in Section 4.5, and requires less computational effort when compared to presently existing methods.

Given a set of observations from an HSMM signal, $\mathcal{Y}_T = \{y_1, y_2, \ldots, y_T\}$, we like to find $\boldsymbol{\theta}$, the parameters of the HSMM model. The algorithm we use is a variant of the EM algorithm. We first initialize $\boldsymbol{\theta}$ to an initial guess. Similar to the EM algorithm for identification of HMMs, in the '**E**' step of our algorithm we define a set of probabilistic measures which describe the evolution of the hidden state variable $\boldsymbol{s}_t$. We define 'forward variables' $\alpha_t(i)$ as the conditional probability of observing the partial sequence $y_1, y_2, \ldots, y_t$ and being in state i at time t, given the model parameters $\boldsymbol{\theta}$. That is,

$$\alpha_t(i) = \mathbb{P}(\boldsymbol{s}_t = \boldsymbol{e}_i, y_1 y_2 \ldots y_t | \boldsymbol{\theta}) \tag{4.39}$$

Let $\hat{\boldsymbol{d}}_t = \begin{bmatrix}\hat{d}_t(1) & \hat{d}_t(2) & \cdots & \hat{d}_t(N)\end{bmatrix}'$, where

$$\hat{d}_t(i) = \mathbb{E}(d_t(i) | \boldsymbol{s}_t = i, \boldsymbol{\theta}, y_1, y_2, \ldots, y_t) \tag{4.40}$$

is our estimate of the state-duration variable for state i at time t. $\hat{\boldsymbol{d}}_t$ is initialized to $\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}'$ for $t = 1$. We reconstruct $\hat{d}_{t+1}(i)$ iteratively as

$$\begin{aligned}\hat{d}_{t+1}(i) &= 1 + \mathbb{E}(\boldsymbol{s}_t(i) | y_1 y_2 \ldots y_t, \boldsymbol{\theta}) \cdot \hat{d}_t(i), && 1 \leq i \leq N \\ &= 1 + \frac{\alpha_t(i)}{\sum\limits_{i=1}^{N} \alpha_t(i)} \cdot \hat{d}_t(i), && 1 \leq i \leq N\end{aligned} \tag{4.41}$$

The state transition matrix $\boldsymbol{A}_{\boldsymbol{d}_t}$ is updated for each t as

$$\boldsymbol{A}_{\boldsymbol{d}_t} = \boldsymbol{P}(\hat{\boldsymbol{d}}_t) + (\boldsymbol{I} - \boldsymbol{P}(\hat{\boldsymbol{d}}_t))\boldsymbol{A}^o \tag{4.42}$$

where $\boldsymbol{P}$ is given in equation 4.33.

The forward variable $\alpha_t(i)$ for $t = 1$ is initialized to the given initial state, that is, $\alpha_1(i) = \boldsymbol{s}_1(i)$ for $1 \leq i \leq N$. The other forward variables are constructed iteratively as

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij}\right] b_j(y_{t+1}), \qquad 1 \leq t \leq T,\, 1 \leq j \leq N \tag{4.43}$$

Similarly, the backward variables $\beta_t(i)$ are defined as the probability of observing the partial sequence $y_{t+1}, y_{t+2}, \ldots, y_T$, given the model parameters $\boldsymbol{\theta}$ and that the state at time t is e_i.

$$\beta_t(i) = \mathbb{P}(y_{t+1}y_{t+2}\ldots y_T|\boldsymbol{s}_t = e_i, \boldsymbol{\theta})$$

β_t's are computed by initializing $\beta_T(i) = 1$, for $1 \leq i \leq N$ and constructing the other variables iteratively using

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) \cdot a_{ij} \cdot b_j(y_{t+1}) \qquad 1 \leq t \leq T,\, 1 \leq j \leq N \tag{4.44}$$

We define $\gamma_t(i)$ as the probability of being in state i at time t, given the observation sequence $\mathcal{Y}_T$ and the model parameters $\boldsymbol{\theta}$.

$$\gamma_t(i) = \mathbb{P}(s_t = i|\mathcal{Y}_T, \boldsymbol{\theta}) \tag{4.45}$$

$\gamma_t(i)$ is expressed in terms of α's and β's as

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)} \tag{4.46}$$

Also, we define $\xi_t(i,j)$ as the conditional probability of being in state i at time t, and state j at time $t+1$

$$\xi_t(i,j) = \mathbb{P}(s_t = i, s_{t+1} = j|\mathcal{Y}_T, \boldsymbol{\theta}) \tag{4.47}$$

$\xi_t(i,j)$ is given in terms of α's and β's as

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)} \tag{4.48}$$

The variables γ and ξ are useful in interpreting the number of transitions between states. That is, $\sum_{t=1}^{T-1}\gamma_t(i)$ is the expected number of transitions from state i to any other state in $\mathcal{Y}_T$, and $\sum_{t=1}^{T-1}\xi_t(i,j)$ is the expected number of transitions from state i to state j in $\mathcal{Y}_T$.

In step **M** of the algorithm, the model parameters $\boldsymbol{\theta}$ are updated to the maximum likelihood estimate of the model parameters computed from the forward-backward variables in step **E**. There are different approaches to obtaining the updating equations, all of which result in the same equations. These approaches are: 1) One can find $\boldsymbol{\theta}$ such that the 'Baum's auxiliary function,' Q, is maximized. Baum's auxiliary function is defined as the expectation of $\log \mathbb{P}(\mathcal{Y}_T, \boldsymbol{S}|\boldsymbol{\theta}))$. It can be shown that maximizing the Baum's auxiliary function is analogous to maximizing the likelihood $\mathbb{P}(\mathcal{Y}_T|\boldsymbol{\theta})$. Hence, one can solve $\partial Q/\partial\boldsymbol{\theta} = 0$ to obtain the update equations for the model parameters. 2) The problem can be set-up as a constrained maximization problem and the Lagrange-multiplier approach can be used to maximize the auxiliary function Q. 3) One can use the filtration variables, $\boldsymbol{\alpha}_t$'s and $\boldsymbol{\beta}_t$'s, to count the expected number of transitions and use the concept of counting the event occurrence to obtain the update equations. We use the latter approach here, though the result is analogous if other approaches were used. Based on the definitions, $\sum_{t=1}^{T-1}\gamma_t(i)$ is the expected number of transitions from state i in $\mathcal{Y}_T$ and $\sum_{t=1}^{T-1}\xi_t(i,j)$ is the expected number of transitions from state i to state j in $\mathcal{Y}_T$. Then, the transition probabilities are estimated as

$$\underset{i\neq j}{a^o_{ij}} = \frac{\text{expected number of transitions from state } i \text{ to state } j \text{ in } \mathcal{Y}_T}{\text{expected number of transitions from state } i \text{ in } \mathcal{Y}_T}$$

$$= \frac{\displaystyle\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{4.49}$$

$$= \frac{\displaystyle\sum_{t=1}^{T-1}\alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)}{\displaystyle\sum_{t=1}^{T-1}\alpha_t(i)\beta_t(i)} \tag{4.50}$$

Similarly, the mean and variance of the observation process are estimated as

$$\mu_i = \frac{\sum_{t=1}^{T-1} \gamma_t(i) y_t}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{4.51}$$

$$\sigma_i^2 = \frac{\sum_{t=1}^{T-1} \gamma_t(i)(y_t - \mu_i)^2}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{4.52}$$

Let $\mu_{i,s}$ and $\sigma_{i,s}^2$ be the mean and variance of the state-duration for state i respectively. $\mu_{i,s}$, is estimated as

$$\mu_{i,s} = \frac{\sum_{t=1}^{T-1} \mathbb{P}(\boldsymbol{s}_{t+1} \neq \boldsymbol{e}_i, \boldsymbol{s}_t = \boldsymbol{e}_i | \mathcal{Y}_T, \boldsymbol{\theta}) \hat{d}_t(i)}{\sum_{t=1}^{T-1} \mathbb{P}(\boldsymbol{s}_{t+1} \neq \boldsymbol{e}_i, \boldsymbol{s}_t = \boldsymbol{e}_i | \mathcal{Y}_T, \boldsymbol{\theta})} \tag{4.53}$$

We have $\mathbb{P}(S_{t+1} \neq i, S_t = i | \mathcal{Y}_T, \boldsymbol{\theta}) = \gamma_t(i) - \xi_t(i,i)$. Hence, $\mu_{i,s}$, in terms of α's and β's is given by

$$\mu_{i,s} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \left(\beta_t(i) - a_{ii} b_i(y_{t+1}) \beta_{t+1}(i)\right) \hat{d}_t(i)}{\sum_{t=1}^{T-1} \alpha_t(i) \left(\beta_t(i) - a_{ii} b_i(y_{t+1}) \beta_{t+1}(i)\right)} \tag{4.54}$$

Similarly, the variance of state-duration distribution is given by

$$\sigma_{i,s}^2 = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \left(\beta_t(i) - a_{ii} b_i(y_{t+1}) \beta_{t+1}(i)\right) \left(\hat{d}_t(i) - \mu_{i,s}\right)^2}{\sum_{t=1}^{T-1} \alpha_t(i) \left(\beta_t(i) - a_{ii} b_i(y_{t+1}) \beta_{t+1}(i)\right)} \tag{4.55}$$

- **E Step:**

  1. *Forward Filtering:* Compute $\alpha_t(i)$ and $\hat{d}_t(i)$ for $1 \leq i \leq N$ and $1 \leq t \leq T-1$ using equations 4.41 and 4.43.
  2. *Backward Filtering:* Compute $\beta_t(i)$ for $1 \leq i \leq N$ and $1 \leq t \leq T$ using equation 4.44.
  3. *Find the State Probabilities:* Compute $\xi_t(i,j)$ and $\gamma_t(i)$ using equations 4.48 and 4.46.

- **M Step:**

  1. *Update the model parameters, $\boldsymbol{\theta}$:* Use equations 4.49, 4.51, 4.52, 4.54 and 4.55 to update the model parameters.

Figure 4.5: '**E**' and '**M**' steps of our algorithm for off-line identification of HSMMs.

The parameters of the state duration distributions, ν_i and η_i, are given in terms of $\mu_{s,i}$ and $\sigma^2_{i,s}$ as

$$\begin{aligned} \nu_i &= \frac{\mu^2_{s,i}}{\sigma^2_{s,i}} \\ \eta_i &= \frac{\mu_{s,i}}{\sigma^2_{s,i}} \end{aligned} \tag{4.56}$$

Figure 4.5 summarizes our algorithm. The algorithm stops when the $\boldsymbol{\theta}$ converges to a constant vector. The forward-backward algorithm has the computational complexity of $\mathcal{O}(N^2T)$ per pass and requires a memory of $3NT$ because all the forward-backward variables and the estimate's of the state duration variables need to be stored.

4.6.1 Implementation Issues

Choice of Initial Estimates Since the EM algorithm finds the local maximum of the likelihood function, it is important to start the algorithm with suitable initial values. Though there is no straightforward method for selecting proper initial values, there are a few techniques in the literature which can assist in selecting the initial values. One of these methods uses segmentation of the observations into states, and averaging the observations between the segmented states. Segmentation can be performed manually, using maximum likelihood segmentation or the 'segmental k-means' method. For more information on these techniques, please refer to [86].

Scaling From equation 4.43 it becomes clear that as t increases, α_t's become small very fast, and can quickly fall out of the numerical range of any computer. The same argument applies to β_t's, when t decreases. To avoid this, we suggest using a scaling scheme similar to the scheme used in [72] and [86] for the HMM case, where α_t's are scaled to sum up to 1 for all t. More precisely, let α_t denote the unscaled variable, $\hat{\alpha}_t(i)$ denote the scaled variable and $\hat{\hat{\alpha}}_t$ the local version of α's before scaling. Let c_t be the scaling factor at time t, where $c_t = 1/\sum_{i=1}^{N} \hat{\hat{\alpha}}_t(i)$. Both α's and β's are scaled using c_t, that is, $\hat{\alpha}_t(i) = c_t\hat{\hat{\alpha}}_t(i)$ and $\hat{\beta}_t(i) = c_t\hat{\hat{\beta}}_t(i)$. It can be easily shown that

$$\hat{\alpha}_t(i) = C_t\alpha_t(i) \qquad\qquad \hat{\beta}_{t+1}(i) = D_{t+1}\beta_{t+1}(i) \tag{4.57}$$

where $C_t = \prod_{s=1}^{T} c_s$ and $D_{t+1} = \prod_{s=t+1}^{T} c_s$. Using these equations, it can be shown that the scaling factors are cancelled out in all of the final update equations, except equations 4.54 and 4.55. In order to cancel the scaling effect on equations 4.54 and 4.55, these equations should be rewritten as

$$\mu_{i,s} = \frac{\displaystyle\sum_{t=1}^{T-1} \alpha_t(i)\left(\beta_t(i)c_t^{-1} - a_{ii}b_i(y_{t+1})\beta_{t+1}(i)\right)\hat{d}_t(i)}{\displaystyle\sum_{t=1}^{T-1} \alpha_t(i)\left(\beta_t(i)c_t^{-1} - a_{ii}b_i(y_{t+1})\beta_{t+1}(i)\right)} \tag{4.58}$$

$$\sigma_{i,s}^2 = \frac{\displaystyle\sum_{t=1}^{T-1} \alpha_t(i)\left(\beta_t(i)c_t^{-1} - a_{ii}b_i(y_{t+1})\beta_{t+1}(i)\right)\left(\hat{d}_t(i) - \mu_{i,s}\right)^2}{\displaystyle\sum_{t=1}^{T-1} \alpha_t(i)\left(\beta_t(i)c_t^{-1} - a_{ii}b_i(y_{t+1})\beta_{t+1}(i)\right)} \tag{4.59}$$

This assures us that the scaling will have no effect on parameter estimates.

4.7 Online Identification of HSMMs

In this section, we present a new algorithm for online identification of HSMMs. Our algorithm is based on the signal generation model presented in Section

4.5. Our approach is to set up the problem of online identification of HSMMs such that the general recursive prediction error (RPE) method can be applied to the problem. First, we will present a general background on general RPE technique in Section 4.7.1. Then, we will present how we adopt this technique for online identification of HSMMs in Section 4.7.2.

4.7.1 General RPE technique

RPE is a class of general numerical parameter estimation, where a norm $V(\theta)$ is defined that measures the prediction error and the estimate of the model parameters is updated according to

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \lambda \cdot f^{(t)} \tag{4.60}$$

where $f^{(t)}$ is a search direction based on some information about $V(\theta)$ acquired at previous iterations, and λ is a positive step size. An important class of these methods is the *Newton* type algorithms, where the correction in 4.60 is chosen in the *Newton Direction*

$$f^{(t)} = -[V''(\hat{\theta}_t)]^{-1} \cdot V'(\hat{\theta}_t) \tag{4.61}$$

There are approaches that use values of function V as well as of its gradients. The most important subclass of these approaches is the *quasi-Newton methods*, which somehow form an estimate of V'' and then use equation 4.61.

Now consider a weighted quadratic prediction error criterion

$$V(\theta) = \gamma_t \sum_{k=1}^{t} \beta(t,k)\epsilon^2(k,\theta) \tag{4.62}$$

where $\epsilon(k,\theta) = \hat{y}(k) - y(k)$ is the prediction error. $\beta(t,k)$ is a weighting sequence with the following property

$$\beta(t,k) = \lambda(t)\beta(t-1,k) \qquad ,0 \leq k \leq t-1 \tag{4.63}$$

$$\beta(t,t) = 1 \tag{4.64}$$

γ is a normalizing factor. Then it is shown that the general search equation is given by

$$\hat{\theta}_t = \hat{\theta}_{t-1} - \lambda_t[R_t]^{-1} \cdot V' \tag{4.65}$$

where V' is the gradient of V, and R_t is a matrix that modifies the search direction. It is shown that this updating formula can be written as

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma_t \cdot [R_t]^{-1} \cdot \psi(t, \hat{\theta}_{t-1}) \cdot \epsilon(t, \hat{\theta}_{t-1}) \tag{4.66}$$

where ψ is the gradient matrix of $\hat{y}$ with respect to θ

The simplest choice for R is $R_t = I$, which results in the gradient or the steepest-descent method. This method becomes fairly inefficient in the region close to the minimum of the norm function. Choosing $R_t = V''(\hat{\theta}_t)$, makes the equation 4.66 a Newton method, which typically perform much better close to the minimum. In this case, V'' is given by

$$V''(\theta) = \frac{1}{N}\sum_{t=1}^{N} \psi(t,\theta) \cdot \psi'(t,\theta) - \frac{1}{N}\sum_{t=1}^{N} \psi'(t,\theta) \cdot \epsilon(t,\theta) \tag{4.67}$$

It may however be quite costly to compute all the terms of ψ'. Then an approximation of V'' is used, where the second sum in equation 4.67 is ignored

$$V''(\theta) \approx \frac{1}{N}\sum_{t=1}^{N} \psi(t,\theta) \cdot \psi'(t,\theta) \tag{4.68}$$

which is by definition the Hessian of $\hat{y}$. If we choose $R_t = V''$ and use the approximation in equation 4.68, then it is shown that R_t is given as

$$R_t = \gamma_t \sum_{k=1}^{t} \beta(t,k) \cdot \psi(k) \cdot \psi'(k) \tag{4.69}$$

It is also shown that R_t can be constructed recursively as

$$\begin{aligned} R_t &= R_{t-1} + \gamma_t[\psi(t)\psi'(t) - R(t-1)] \\ &= (1-\gamma_t) \cdot R(t-1) + \gamma_t\psi\psi' \end{aligned} \tag{4.70}$$

The RPE algorithm is summarized with the recursive scheme presented in Figure 4.6.

4.7.2 Online Identification of HSMMs Using the RPE Method

Let $\boldsymbol{\theta}_t$ denote our estimate of the model parameters at t. We define the 'objective function' $\ell_t(\boldsymbol{\theta}_t) = \log \mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}_t)$, as the log-likelihood of the

1. $\epsilon(t) = y(t) - \hat{y}(t)$
2. $\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma_t \cdot [R_t]^{-1} \cdot \psi(t) \cdot \epsilon(t)$
3. $R_t = R_{t-1} + \gamma_t[\psi(t)\psi'(t) - R(t-1)]$

Figure 4.6: Parameter update equations performed in each step of a general RPE algorithm.

observations up to time t given $\boldsymbol{\theta}_t$. $\ell_t(\boldsymbol{\theta}_t)$ can be rewritten as

$$\begin{aligned} \ell_t(\boldsymbol{\theta}_t) &= \sum_{\tau=1}^{t} \log \mathbb{P}(y_\tau | y_1, y_2, \ldots, y_{\tau-1}; \boldsymbol{\theta}_t) \\ &= \sum_{\tau=1}^{t} u_\tau(\boldsymbol{\theta}_t) \end{aligned} \tag{4.71}$$

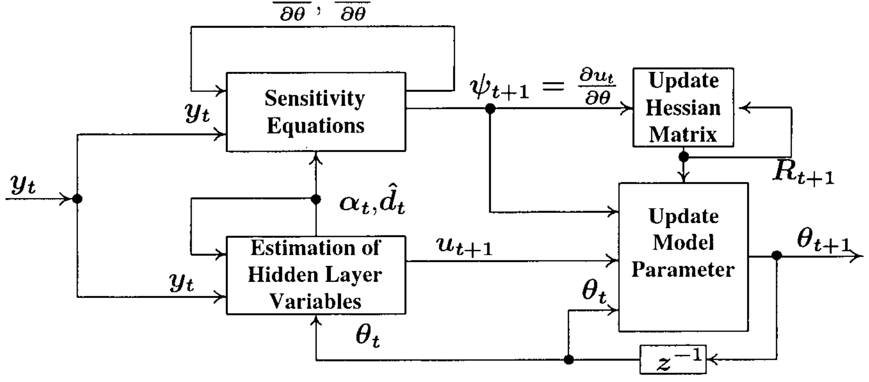
where $u_\tau = \log \mathbb{P}(y_\tau | y_1, y_2, \ldots, y_{\tau-1}; \boldsymbol{\theta}_t)$ is the log-likelihood increment. Our approach here is to update $\boldsymbol{\theta}_t$ in a direction that maximizes the objective function $\ell_t(\boldsymbol{\theta}_t)$. We use the recursive prediction error (RPE) method, where the parameters are updated in the Newton-Raphson direction (see [84] for an extensive discussion of this method). This selection greatly increases the speed of the algorithm. Starting with an initial guess for $\boldsymbol{\theta}_t$ at $t = 1$, $\boldsymbol{\theta}_t$ is updated at each time instance t using

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda_{t+1} \cdot R_{t+1}^{-1} \cdot \psi_{t+1} \tag{4.72}$$

where $R_t = \partial^2 \ell_t(\boldsymbol{\theta})/\partial \boldsymbol{\theta}^2$ is an estimate of the 'Hessian Matrix'. $\psi_t = \partial u_t(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ determines the search direction and is the gradient of u_t with respect to θ_t. λ_t is a step size. R_t and ψ_t are called 'sensitivity parameters' and are estimated recursively in our algorithm.

Figure 4.7 illustrates the basic block diagram of our scheme. The details of how each block performs its function are described in sections 4.7.4 to 4.7.7. Below, we summarize the basic function of each block.

1. *Estimation of hidden layer variables:* Given the observation y_t and model parameters estimate $\boldsymbol{\theta}_t$, this block computes the forward variable $\boldsymbol{\alpha}_t$ and state duration estimate $\hat{\boldsymbol{d}}_t$. These variables are constructed iteratively from $\boldsymbol{\alpha}_{t-1}$ and $\hat{\boldsymbol{d}}_{t-1}$.



$\frac{\partial \alpha_t}{\partial \theta}, \frac{\partial \hat{d}_t}{\partial \theta}$
Sensitivity Equations
y_t
$\psi_{t+1} = \frac{\partial u_t}{\partial \theta}$
Update Hessian Matrix
R_{t+1}
y_t
$\alpha_t, \hat{d}_t$
Update Model Parameter
Estimation of Hidden Layer Variables
u_{t+1}
θ_{t+1}
y_t
θ_t
θ_t
z^{-1}

Figure 4.7: Recursive Prediction Error scheme for estimating HSMM parameters.

2. *Updating the gradient vector* ψ_{t+1}*:* Using the obtained $\boldsymbol{\alpha}_t$ and $\hat{\boldsymbol{d}}_t$, this block computes the gradient vector ψ_{t+1}. For this computation, $\partial\boldsymbol{\alpha}_{t-1}/\partial\boldsymbol{\theta}$ and $\partial\hat{\boldsymbol{d}}_t/\partial\boldsymbol{\theta}$ should be estimated as well. Hence, this block constructs $\partial\boldsymbol{\alpha}_{t-1}/\partial\boldsymbol{\theta}$ and $\partial\hat{\boldsymbol{d}}_t/\partial\boldsymbol{\theta}$ recursively and uses these parameters to update ψ_{t+1}.

3. *Updating the parameters estimate* $\boldsymbol{\theta}$*:* This block updates the model parameters estimate, $\boldsymbol{\theta}_t$, using equation 4.72.

4. *Updating* R_t*:* This block recursively updates the estimate of the Hessian matrix, R_t, from R_{t-1} and ψ_t.

We present the details of each part of our algorithm in the following sub-sections. For simplicity, we use $\boldsymbol{\theta}$ instead of $\boldsymbol{\theta}_t$ in our notations.

4.7.3 Model Parameterizations

Similar to the off-line case, our signal model has N^2+$3N$ parameters. However, as shown in [71], it is more convenient to parameterize the model as

$$\boldsymbol{\theta} = \left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, c_{12}, c_{13}, \cdots, c_{N-1,N}, \boldsymbol{\nu}, \boldsymbol{\eta}\right)' \qquad (4.73)$$

where c_{ij} are simply defined as $c_{ij} = (a^o_{ij})^{1/2}$. This parameterization selection simplifies the development of the update equations. As in the off-line case, $\boldsymbol{\mu}$

and $\boldsymbol{\sigma}^2$ are the vectors of the mean and variance of the observations process, and $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$ are the vectors of the parameters of the gamma probability density functions of the state durations.

4.7.4 Estimation of the Hidden Layer Variables

As with the off-line case, we define the forward variables as $\alpha_t(i) = \mathbb{P}(y_1 y_2 \dots y_t, s_t = i|\boldsymbol{\theta})$. Let $\boldsymbol{\alpha}_t = \begin{bmatrix}\alpha_t(1) & \alpha_t(2) & \cdots & \alpha_t(N)\end{bmatrix}'$. Then the forward filtering recursion equation is given by

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}\boldsymbol{\alpha}_t \tag{4.74}$$

where $\boldsymbol{B}$ is a diagonal matrix, $b_{ii} = b_i(y_{t+1})$ (see equation 4.36), and $\boldsymbol{A}'_{\boldsymbol{d}_t}$ is the transpose of the state transition matrix (equation 4.35). The conditional estimate of the state at time t is given by

$$\begin{aligned}\gamma_t &= \mathbb{E}(\boldsymbol{s}_t|y_1, y_2, \dots, y_t, \boldsymbol{\theta}) \\ &= \boldsymbol{\alpha}_t\langle\boldsymbol{\alpha}_t, \mathbf{1}\rangle^{-1} \end{aligned} \tag{4.75}$$

Given the observations up to time t, the next state and next observation of the signal are estimated as

$$\mathbb{E}(\boldsymbol{s}_{t+1}|y_1, y_2, \dots, y_t, \boldsymbol{\theta}) = \boldsymbol{A}'_{\boldsymbol{d}_t}\gamma_t \tag{4.76}$$

$$\begin{aligned}\hat{y}_{t+1} &= \langle\boldsymbol{\mu}, \mathbb{E}(\boldsymbol{s}_{t+1}|y_1, y_2, \dots, y_t, \boldsymbol{\theta})\rangle \\ &= \langle\boldsymbol{\mu}, \boldsymbol{A}'_{\boldsymbol{d}_t}\boldsymbol{\alpha}_t \cdot \langle\boldsymbol{\alpha}_t, \mathbf{1}\rangle^{-1}\rangle \end{aligned} \tag{4.77}$$

The estimate of the state duration variable is updated similarly to the off-line case (equation 4.41), as

$$\begin{aligned}\hat{\boldsymbol{d}}_{t+1} &= \mathbb{E}(\boldsymbol{s}_t|y_1, y_2, \dots, y_t, \boldsymbol{\theta}) \odot \hat{\boldsymbol{d}}_t + 1 \\ &= \boldsymbol{\alpha}_t\langle\boldsymbol{\alpha}_t, \mathbf{1}\rangle^{-1} \odot \hat{\boldsymbol{d}}_t + 1 \end{aligned} \tag{4.78}$$

The log-likelihood increment, u_{t+1} (equation 4.71), is given by

$$u_{t+1} = \log \mathbb{P}(y_{t+1}|y_1, y_2, \dots, y_t; \boldsymbol{\theta}) \tag{4.79}$$

$$= \log \sum_{i=1}^{N} \mathbb{P}(y_{t+1}|\boldsymbol{s}_{t+1} = \boldsymbol{e}_i, y_1, y_2, \dots, y_t; \boldsymbol{\theta}) \times \mathbb{P}(\boldsymbol{s}_{t+1} = \boldsymbol{e}_i|y_1, y_2, \dots, y_t; \boldsymbol{\theta}) \tag{4.80}$$

$$= \log\langle\mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}\gamma_t\rangle \tag{4.81}$$

4.7.5 Gradient Vector Update Equations

We denote the derivative operator with respect to variable x by D_x, that is, $D_x(.) = \partial(.)/\partial x$. Thus, the gradient vector ψ_t is written as

$$\begin{aligned}
\psi_{t+1} &= \frac{\partial u_{t+1}}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} \\
&= \left(\frac{\partial u_{t+1}}{\partial \mu_i}, \frac{\partial u_{t+1}}{\partial \sigma_i^2}, \frac{\partial u_{t+1}}{\partial c_{ij}}, \frac{\partial u_{t+1}}{\partial \mu_{i,s}}, \frac{\partial u_{t+1}}{\partial \sigma_{i,s}^2}\right)'\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} \\
&= \left(D_{\mu_i}u_{t+1}, D_{\sigma_i^2}u_{t+1}, D_{c_{ij}}u_{t+1}, D_{\mu_{i,s}}u_{t+1}, D_{\sigma_{i,s}^2}u_{t+1}\right)'\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} \qquad (4.82)
\end{aligned}$$

We write the elements of ψ_{t+1} in terms of the derivative of the filtration parameters (i.e.,$\boldsymbol{\alpha}_t$ and $\boldsymbol{d}_t$) with respect to the model parameters. Our estimates of the derivatives of the $\boldsymbol{\alpha}_t$ and $\boldsymbol{d}_t$ are constructed recursively from their estimates at $t-1$. For example, the first element of ψ_{t+1}, $D_{\mu_j}u_{t+1}$, is written in terms of $D_{\mu_j}\boldsymbol{\alpha}_t$ and $D_{\mu_j}\boldsymbol{d}_t$. $D_{\mu_j}\boldsymbol{\alpha}_{t+1}$ and $D_{\mu_j}\boldsymbol{d}_{t+1}$ are also constructed recursively from $D_{\mu_j}\boldsymbol{\alpha}_t$ and $D_{\mu_j}\boldsymbol{d}_t$. In deriving the update equations, it is assumed that the probabilities of non-recurrent transitions (i.e., c_{ij}'s), and the parameters of the state duration probability density functions (i.e., $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$) do not depend on each other.

Thus, to calculate ψ_{t+1}, we use the following equations for the derivatives of u_{t+1} with respect to $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, c_{ij}, $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$.

1-D_{μ_i}

$$D_{\mu_i}u_{t+1} =$$
$$\langle \mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}\gamma_t\rangle^{-1}\left(\langle \mathbf{1}, D_{\mu_i}(\boldsymbol{B})\boldsymbol{A}'_{\boldsymbol{d}_t}\gamma_t\rangle + \langle \mathbf{1}, \boldsymbol{B}D_{\mu_i}(\boldsymbol{A}'_{\boldsymbol{d}_t})\gamma_t\rangle + \langle \mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}D_{\mu_i}(\gamma_t)\rangle\right) \qquad (4.83)$$

$$D_{\mu_i}\gamma_t = D_{\mu_i}(\boldsymbol{\alpha}_t)\langle \mathbf{1}, \boldsymbol{\alpha}_t\rangle^{-1} + \boldsymbol{\alpha}_t\langle \mathbf{1}, D_{\mu_i}(\boldsymbol{\alpha}_t)\rangle^{-1} \qquad (4.84)$$

$$D_{\mu_i}\boldsymbol{\alpha}_t = D_{\mu_i}(\boldsymbol{B})\boldsymbol{A}'_{\boldsymbol{d}_t}\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}D_{\mu_i}(\boldsymbol{A}'_{\boldsymbol{d}_t})\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}D_{\mu_i}(\boldsymbol{\alpha}_{t-1}) \qquad (4.85)$$

$$D_{\mu_i}\boldsymbol{B} = (\frac{y_t - \mu_i}{\sigma^2}) \cdot \boldsymbol{B} \cdot \mathrm{diag}(\boldsymbol{e}_i) \qquad (4.86)$$

$$D_{\mu_i}\boldsymbol{A}_{\boldsymbol{d}_t} = D_{\mu_i}(\boldsymbol{P})(I - \boldsymbol{A}^o) \qquad (4.87)$$

$$D_{\mu_i}\boldsymbol{P} = \\ \operatorname{diag}\left(\frac{\phi(\boldsymbol{d}_t-1)-\phi(\boldsymbol{d}_t)+\phi(\boldsymbol{d}_t)\Phi(\boldsymbol{d}_t-1)-\phi(\boldsymbol{d}_t-1)\Phi(\boldsymbol{d}_t)}{(1-\Phi(\boldsymbol{d}_t-1))^2}\odot D_{\mu_i}\boldsymbol{d}_t\right) \tag{4.88}$$

$$D_{\mu_i}\boldsymbol{d}_t = \boldsymbol{d}_{t-1}\odot D_{\mu_i}\gamma_{t-1} + D_{\mu_i}\boldsymbol{d}_{t-1}\odot\gamma_{t-1} \tag{4.89}$$

$\underline{2\text{-}D_{\sigma^2}}$

$$D_{\sigma_i^2}u_{t+1} = \\ \langle\boldsymbol{1},\boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}\gamma_t\rangle^{-1}\left(\langle\boldsymbol{1},D_{\sigma_i^2}(\boldsymbol{B})\boldsymbol{A}'_{\boldsymbol{d}_t}\gamma_t\rangle+\langle\boldsymbol{1},\boldsymbol{B}D_{\sigma_i^2}(\boldsymbol{A}'_{\boldsymbol{d}_t})\gamma_t\rangle+\langle\boldsymbol{1},\boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}D_{\sigma_i^2}(\gamma_t)\rangle\right) \tag{4.90}$$

$$D_{\sigma_i^2}\gamma_t = D_{\sigma_i^2}(\boldsymbol{\alpha}_t)\langle\boldsymbol{1},\boldsymbol{\alpha}_t\rangle^{-1}+\boldsymbol{\alpha}_t\langle\boldsymbol{1},D_{\sigma_i^2}(\boldsymbol{\alpha}_t)\rangle^{-1} \tag{4.91}$$

$$D_{\sigma_i^2}\boldsymbol{\alpha}_t = D_{\sigma_i^2}(\boldsymbol{B})\boldsymbol{A}'_{\boldsymbol{d}_t}\boldsymbol{\alpha}_{t-1}+\boldsymbol{B}D_{\sigma_i^2}(\boldsymbol{A}'_{\boldsymbol{d}_t})\boldsymbol{\alpha}_{t-1}+\boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}D_{\sigma_i^2}(\boldsymbol{\alpha}_{t-1}) \tag{4.92}$$

$$D_{\sigma_i^2}\boldsymbol{B} = (\frac{(y_t-\mu_i)^2}{2\sigma_i^4}-\frac{1}{2\sigma_i^2})\boldsymbol{B}\cdot\operatorname{diag}(\boldsymbol{e}_i) \tag{4.93}$$

$$D_{\sigma_i^2}\boldsymbol{A}_{\boldsymbol{d}_t} = D_{\sigma_i^2}(\boldsymbol{P})(\boldsymbol{I}-\boldsymbol{A}^o) \tag{4.94}$$

$$D_{\sigma_i^2}\boldsymbol{P} = \\ \operatorname{diag}\left(\frac{\phi(\boldsymbol{d}_t-1)-\phi(\boldsymbol{d}_t)+\phi(\boldsymbol{d}_t)\Phi(\boldsymbol{d}_t-1)-\phi(\boldsymbol{d}_t-1)\Phi(\boldsymbol{d}_t)}{(1-\Phi(\boldsymbol{d}_t-1))^2}\odot D_{\sigma_i^2}\boldsymbol{d}_t\right) \tag{4.95}$$

$$D_{\sigma_i^2}\boldsymbol{d}_t = D_{\sigma_i^2}(\gamma_{t-1})\odot\boldsymbol{d}_{t-1}+\gamma_{t-1}\odot D_{\sigma_i^2}(\boldsymbol{d}_{t-1}) \tag{4.96}$$

3-$D_{c_{mn}}$

$$D_{c_{mn}} u_{t+1} = \langle \mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}\gamma_t\rangle^{-1} \left(\langle \mathbf{1}, \boldsymbol{B} D_{c_{mn}}(\boldsymbol{A}'_{\boldsymbol{d}_t})\gamma_t\rangle + \langle \mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t} D_{c_{mn}}(\gamma_t)\rangle\right) \tag{4.97}$$

$$D_{c_{mn}}\gamma_t = D_{c_{mn}}(\boldsymbol{\alpha}_t)\langle \mathbf{1}, \boldsymbol{\alpha}_t\rangle^{-1} + \boldsymbol{\alpha}_t\langle \mathbf{1}, D_{c_{mn}}(\boldsymbol{\alpha}_t)\rangle^{-1} \tag{4.98}$$

$$D_{c_{mn}}\boldsymbol{\alpha}_t = \boldsymbol{B} D_{c_{mn}}(\boldsymbol{A}'_{\boldsymbol{d}_t})\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t} D_{c_{mn}}(\boldsymbol{\alpha}_{t-1}) \tag{4.99}$$

$$D_{c_{mn}}\boldsymbol{A}_{\boldsymbol{d}_t} = -\boldsymbol{P} \cdot D_{c_{mn}}(\boldsymbol{A}^o) \tag{4.100}$$

$$D_{c_{mn}} a^o_{ij} = \begin{cases} 0 & \text{if } m \neq i \\ 2c_{ij} & \text{if } m = i,\ n = j \\ -2c_{mn} & \text{if } m = i,\ n \neq j \end{cases} \tag{4.101}$$

4- D_η

$$D_{\eta_i} u_{t+1} = \langle \mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t}\gamma_t\rangle^{-1} \left(\langle \mathbf{1}, \boldsymbol{B} D_{\eta_i}(\boldsymbol{A}'_{\boldsymbol{d}_t})\gamma_t\rangle + \langle \mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t} D_{\eta_i}(\gamma_t)\rangle\right) \tag{4.102}$$

$$D_{\eta_i}\gamma_t = D_{\eta_i}(\boldsymbol{\alpha}_t)\langle \mathbf{1}, \boldsymbol{\alpha}_t\rangle^{-1} + \boldsymbol{\alpha}_t\langle \mathbf{1}, D_{\eta_i}(\boldsymbol{\alpha}_t)\rangle^{-1} \tag{4.103}$$

$$D_{\eta_i}\boldsymbol{\alpha}_t = \boldsymbol{B} D_{\eta_i}(\boldsymbol{A}'_{\boldsymbol{d}_t})\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}\boldsymbol{A}'_{\boldsymbol{d}_t} D_{\eta_i}(\boldsymbol{\alpha}_{t-1}) \tag{4.104}$$

$$D_{\eta_i}\boldsymbol{A}_{\boldsymbol{d}_t} = D_{\eta_i}(\boldsymbol{P})(I - \boldsymbol{A}^o) \tag{4.105}$$

$D_{\eta_i}(\boldsymbol{P})$ is a matrix with all zero elements, except the element in row i and column i, which is given by

$$D_{\eta_i}\left(\frac{1 - \Phi(d_t(i); \eta_i, \nu_i)}{1 - \Phi(d_t(i) - 1; \eta_i, \nu_i)}\right) = \frac{D_{\eta_i}[\Phi(d_t(i) - 1)](1 - \Phi(d_t(i))) - D_{\eta_i}[\Phi(d_t(i))](1 - \Phi(d_t(i) - 1))}{(1 - \Phi(d_t(i) - 1))^2} \tag{4.106}$$

$D_{\eta_i}\Phi(d; \eta_i, \nu_i)$ is obtained by differentiating $\Phi(d; \eta_i, \nu_i)$ as defined in equation 4.27

$$D_{\eta_i}\Phi(d; \eta_i, \nu_i) = \frac{\nu_i}{\eta_i}(\Phi(d; \eta_i, \nu_i) - \Phi(d; \eta_i, \nu_i + 1)) \tag{4.107}$$

5- D_{ν_i}

$$D_{\nu_i} u_{t+1} = \langle \mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{d_t}\gamma_t\rangle^{-1} \left(\langle \mathbf{1}, \boldsymbol{B}D_{\nu_i}(\boldsymbol{A}'_{d_t})\gamma_t\rangle + \langle \mathbf{1}, \boldsymbol{B}\boldsymbol{A}'_{d_t}D_{\nu_i}(\gamma_t)\rangle\right) \tag{4.108}$$

$$D_{\nu_i}\gamma_t = D_{\nu_i}(\boldsymbol{\alpha}_t)\langle \mathbf{1}, \boldsymbol{\alpha}_t\rangle^{-1} + \boldsymbol{\alpha}_t\langle \mathbf{1}, D_{\nu_i}(\boldsymbol{\alpha}_t)\rangle^{-1} \tag{4.109}$$

$$D_{\nu_i}\boldsymbol{\alpha}_t = \boldsymbol{B}D_{\nu_i}(\boldsymbol{A}'_{d_t})\boldsymbol{\alpha}_{t-1} + \boldsymbol{B}\boldsymbol{A}'_{d_t}D_{\nu_i}(\boldsymbol{\alpha}_{t-1}) \tag{4.110}$$

$$D_{\nu_i}(\boldsymbol{A}_{d_t}) = D_{\nu_i}(\boldsymbol{P})(I - \boldsymbol{A}^o) \tag{4.111}$$

$D_{\nu_i}(\boldsymbol{P})$ is a matrix with all zero elements, except the element in row i and column i, which is given by

$$D_{\nu_i}\left(\frac{1 - \Phi(d_t(i); \eta_i, \nu_i)}{1 - \Phi(d_t(i) - 1; \eta_i, \nu_i)}\right) = \\ \frac{D_{\nu_i}[\Phi(d_t(i) - 1)](1 - \Phi(d_t(i))) - D_{\nu_i}[\Phi(d_t(i))](1 - \Phi(d_t(i) - 1))}{(1 - \Phi(d_t(i)))^2}) \tag{4.112}$$

Unfortunately, differentiating $\Phi(d; \eta, \nu)$ as defined in equation 4.26, with respect to ν does not result in a simple form as in equation 4.107. However, we can easily find the numerical value of $D_{\nu_i}\Phi(d; \eta_i, \nu_i)$. We have

$$D_\nu(\Phi(d; \eta, \nu)) = (\log(\eta) - \Psi(\nu))\Phi(d; \eta, \nu) + \int_0^d \log(x)\phi(x; \eta, \nu)dx \tag{4.113}$$

where Ψ is the '*digamma*' function [87, 88]. The digamma function is a known special function defined as

$$\Psi(z) = \frac{d}{dz}\log(\Gamma(z)) = \frac{\Gamma'(z)}{\Gamma(z)} \tag{4.114}$$

where $\Gamma(x)$ is the gamma function. The numerical value of the digamma function at any point is easily obtained from

$$\Psi(x{+}n) = \begin{cases} -\gamma + \displaystyle\sum_{k=1}^{\infty}(-1)^{k+1}\zeta(k+1)x^k & \text{, for } n = 1 \text{ and } -1 < x < 1 \\ \displaystyle\sum_{k=1}^{n-1}\frac{1}{x+k} + \Psi(x+1) & \text{, for } n > 1 \text{ and } -1 < x < 1 \end{cases} \tag{4.115}$$

It is shown that only twenty terms of equation 4.115 suffice to compute the $\Psi(x)$ to the full machine precision in 32-bit floating point format [77, 87]. The term $\int_0^d \log(x)\phi(x;\eta,\nu)dx$ is easily computed using numerical definite integral calculation methods. Therefore, we can compute the numerical values of $D_{\nu_i}(\Phi(d_t(i)-1)$ and $D_{\nu_i}(\Phi(d_t(i))$, and substitute them into equation 4.112 to obtain $D_{\nu_i}(\boldsymbol{P})$.

4.7.6 Parameter Update Equations

After finding ψ_{t+1}, the parameter vector $\boldsymbol{\theta}$ is updated using

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda_{t+1} \cdot R_{t+1}^{-1} \cdot \psi_{t+1} \tag{4.116}$$

4.7.7 Updating the Hessian Matrix Estimate, R_t

R_t is an estimate of the second derivative of the criterion function. R_t is updated recursively (see [84, 89]) as

$$R_t = R_{t-1} + \lambda_t[\psi(t)\psi'(t) - R(t-1)] \tag{4.117}$$

4.7.8 Implementation Issues

Our online identification algorithm finds the *local* maximum likelihood estimate of the model parameters. Hence, similar to the off-line case, it is important to start the algorithm with proper initial values. Furthermore, the following issues should be considered when implementing our algorithm.

Scaling

Similar to the off-line case, the forward variable $\boldsymbol{\alpha}_t$ converges rapidly to zero as t increases. This can be avoided by using the same scaling technique as in the off-line case, where $\boldsymbol{\alpha}_t$ is scaled as

$$\hat{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}_t \langle \mathbf{1}, \boldsymbol{\alpha}_t \rangle^{-1} \tag{4.118}$$

It can be shown that this scaling does not affect the update and recursion equations.

Choice of the step size λ_t

λ_t in equation 4.116 is a step size. In theory, λ_t can be any sequence satisfying

$$\lambda_t \geq 0, \qquad \sum_{t=1}^{\infty} \lambda_t = \infty, \qquad \sum_{t=1}^{\infty} \lambda_t^2 < \infty \tag{4.119}$$

A common choice is $\lambda_t = 1/t$, where λ_t has a *weighting* effect on the norm function, such that recent observations have a stronger effect on the update than older observations. There is a trade-off between the 'tracking ability' and 'noise sensitivity' of the algorithm in selecting λ_t. Choosing a larger step size (e.g., $\lambda_t = 1/\sqrt{t}$) results in faster convergence of the estimate to the real parameters and allows the algorithm to track the changes in the actual parameter values faster. However, selecting larger step sizes makes the estimate more sensitive to noise. The possibility of using different and more sophisticated choices for step sizes is discussed in [89,90] and [91].

Avoiding matrix inversion in the update equations

The update equation 4.116 involves inversion of R_t. This matrix inversion can be avoided by using the matrix inversion lemma [89]. If A,B,C and D are matrixes then the matrix inversion lemma states

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA{-}1B + C^{-1}]^{-1}DA^{-1} \tag{4.120}$$

By taking $A = (1 - \gamma_t)R_{t-1}$, $B = D' = \psi$ and $C = \gamma_t$, parameter 4.116 can be written as:

$$R_t^{-1} = \frac{1}{1-\gamma_t}\left(R_{t-1}^{-1} - \frac{\gamma_t R_{t-1}^{-1}\psi\psi' R_{t-1}^{-1}}{1 - \gamma_t + \gamma_t \psi' R_{t-1}^{-1}\psi}\right) \tag{4.121}$$

Projection of parameters to the constrained domain

Since the parameters of an HSMM are constrained, the estimated parameters should be projected into the constrained space. More precisely, the parameters $c_{ij} = (a_{ij}^o)^{1/2}$ are constrained to $\sum_{i=1}^{N} c_{ij} = 1$. Thus, after updating the parameters using equation 4.116, c_{ij}'s are re-normalized as

$$c_{ij}^2 = \frac{c_{ij}^2}{\sum_{i=1}^{N} c_{ij}^2} \tag{4.122}$$

4.8 Effective Bandwidth of a Hidden Semi-Markov Model

In this section, we show how the effective bandwidth of a signal is obtained from its HSMM's parameters. Our approach is to reformulate the Semi-Markov chain of an HSMM as a Markov chain with larger number of states. First, we will show how the effective bandwidth of an HMM is obtained. Next, we show how a Semi-Markov chain is reformulated as a Markov chain with larger number of states. Finally, we show how the effective bandwidth of an HSMM is obtained using this approach.

4.8.1 Effective Bandwidth of Hidden Markov Models

In this Section, we show how the effective bandwidth of an HMM is obtained. Let y_t be the observation process of an HMM with the hidden Markov state process x_t, and with the transition probabilities matrix A. Let $Y(t)$ be the cumulative sum of observation process y in $[0, t]$, that is $Y(t) = \sum_{\tau=0}^{t} y_\tau$. Also let $\varphi_i(\theta)$ be the moment generating function of the observation layer for state i, that is, $\varphi_i(\theta) = \mathbb{E}(e^{\theta y_t} | x_t = i)$. We define $\psi^i(\theta, t) := \mathbb{E}(e^{\theta Y(t)} | x_1 = i)$. We have

$$\begin{aligned}\psi^i(\theta,t) =&\mathbb{E}(e^{\theta Y(t)}|x_1=i)\\ =&\mathbb{E}(e^{\theta Y(t)-Y(1)}\cdot e^{\theta Y(1)}|x_1=i)\\ =&\mathbb{E}(e^{\theta y_1}|x_1=i)\times\mathbb{E}(e^{\theta(Y(t)-Y(1))}|x_1=1)\\ =&\varphi^i(\theta)\sum_{j=1}^{M}\mathbb{E}(e^{\theta Y(t)-Y(1)}|x_2=j,x_1=i)\times\mathbb{P}(x_2=j|x_1=i)\\ =&\varphi^i(\theta)\sum_{j=1}^{M}\mathbb{E}(e^{\theta Y(t)-Y(1)}|x_2=j)a_{ij}\\ =&\varphi^i(\theta)\sum_{j=1}^{M}\mathbb{E}(e^{\theta Y(t-1)}|x_1=j)a_{ij}\\ =&\varphi^i(\theta)\sum_{j=1}^{M}\psi_y^i(\theta,t)a_{ij}\end{aligned}\tag{4.123}$$

If we consider the vector $\Psi(\theta,t)=[\psi_i(\theta,t)]$ and the diagonal matrix $\Phi(\theta)=\text{diag}[\varphi_i(\theta)]$, then the above equation can be written in matrix form as

$$\Psi(\theta,t)=\Phi(\theta)\cdot A\cdot\Psi(\theta,t-1)\tag{4.124}$$

with the initial condition $\Psi(\theta,1)=\Phi(\theta)\cdot\mathbf{1}^T$, where $\mathbf{1}$ is a column vector with all its elements being one.

Now let π_i be the probability that initial state $x_1=i$, and also let $\boldsymbol{\pi}=[\pi_1,\pi_2,\cdots,\pi_M]$, thus

$$\begin{aligned}\mathbb{E}(e^{\theta Y(t)})&=\boldsymbol{\pi}\Psi(\theta,t)\\ &=\boldsymbol{\pi}(\Phi(\theta)A)^{t-1}\Phi(\theta)\mathbf{1}^T\end{aligned}\tag{4.125}$$

Since the transition matrix A is irreducible, then A is primitive. Since $\Phi(\theta)$ is diagonal, then matrix $\Phi(\theta)A$ is also primitive. Hence, we can use the Perron-Frobenious theorem (see Theorem 8.5.1 in [92]) from matrix theory to show that

$$\lim_{t\to\infty}[\Phi(\theta)A/sp(\Phi(\theta)A)]^n=L(\theta)\tag{4.126}$$

where $L(\theta)$ is a constant matrix and $sp(\Phi(\theta)A)$ is the spectral radius of the matrix $\Phi(\theta)A$. The spectral radius of a matrix is simply the largest absolute eigenvalue of that matrix. Therefore, equations 4.125 and 4.126 result in the following expression

$$\lim_{t\to\infty} \frac{1}{\theta t} \log \mathbb{E}(e^{\theta Y(t)}) = (1/\theta) \log(sp(\Phi(\theta)A)) \tag{4.127}$$

Hence, we have

$$\alpha(\theta) = (1/\theta) \log\left(sp(\Phi(\theta)A)\right) \tag{4.128}$$

where $\Phi(\theta)$ is a diagonal matrix.

4.8.2 Reformulating a Semi-Markov chain as a Markov chain

Let s_t be a Semi-Markov chain with N distinct states, taking its value from the space $\{1, 2, \ldots, N\}$. We assume that the time that signal stays in each state is limited, say by D time units. This means that the state process s_t will not stay in any state for more than D time units. Our motivation in here is to reformulate the N state Semi-Markov chain s_t in the form of a Markov chain with $N \times D$ states. Our approach in similar to what was presented in [78].

Assume that the state process is in state i at time t, i.e., $s_t = i$. Then define d_t as the time that s_t has spent in state i prior to time t. That is

$$d_t = \{d | s_t = i, s_{t-1} = i, \ldots, s_{t-d+1} = i, s_{t-d} \neq i\} \tag{4.129}$$

Notice that

$$d_{t+1} = \begin{cases} d_t + 1 & \text{if } s_{t+1} = s_t \\ 1 & \text{Otherwise} \end{cases} \tag{4.130}$$

Therefore, d_{t+1} depends only on d_t, s_t and s_{t+1}. Note that by definition, variable d_t is restricted to

$$d_{t+1} = \begin{cases} d_t + 1 & \text{if } s_{t+1} = s_t \text{ and } d_t < D \\ d_t & \text{if } s_{t+1} = s_t \text{ and } d_t = D \\ 1 & \text{if } s_{t+1} \neq s_t \end{cases} \tag{4.131}$$

Now consider the vector stochastic process x_t defined as $x_t = (s_t, d_t)$. x_t takes its values from the space $\{(i,d) | 1 \leq i \leq N, 1 \leq d \leq D\}$ and clearly is a finite-state process with $N \times D$ states. Note that

$$\mathbb{P}(x_{t+1} | x_t) = \begin{cases} a_{ii}(d_t) & \text{if } x_t = (i, d_t) \text{ and } x_{t+1} = (i, d_t + 1) \\ a_{ij}(d_t) & \text{if } x_t = (i, d_t) \text{ and } x_{t+1} = (j, 1),\ i \neq j \\ 0 & \text{Otherwise} \end{cases} \tag{4.132}$$

where $a_{ii}(d_t)$ is the probability that s_t stays in state i given that it has spent d_t time units in state i. Similarly $a_{ij}(d_t)$ is the probability that s_t transits from state i to state j given that it has spent d_t time units in state i. Note that $\mathbb{P}(x_{t+1}|x_t)$ is independent of t, and hence, x_t is a Markov chain. For simplicity of notation, we will denote the state space for x_t as $\{1, 2, \ldots, M\}$ from now on, where $M = N \times D$. If $x_t = m$, then we have

$$\begin{aligned} s_t &= \lfloor \frac{m-1}{D} \rfloor + 1 && (4.133) \\ d_t &= m - (s_t - 1)D && \\ &= m - \lfloor \frac{m-1}{D} \rfloor D && (4.134) \end{aligned}$$

where $\lfloor x \rfloor$ is the largest integer less than or equal to x. We also have $m = (s_t - 1)D + d_t$. Let $d = m - \lfloor \frac{m-1}{D} \rfloor D$, $i = \lfloor \frac{m-1}{D} \rfloor + 1$ and $j = \lfloor \frac{n-1}{D} \rfloor + 1$; then the state transition probabilities matrix for the Markov process x_t is given by

$$a_{mn} = \begin{cases} a_{ii}(d) & \text{if } n = m + 1, \text{ and } \lfloor \frac{m-1}{D} \rfloor = \lfloor \frac{n-1}{D} \rfloor \\ a_{ij}(d) & \text{if } n - \lfloor \frac{n-1}{D} \rfloor D = 1, \text{ and } \lfloor \frac{m-1}{D} \rfloor \neq \lfloor \frac{n-1}{D} \rfloor \\ 0 & \text{Otherwise} \end{cases} \qquad (4.135)$$

Since x_t has $M = N \times D$ states, then the state transition matrix for x_t has $M^2 = (ND)^2$ states. However, since variable d_t is restricted based on equation 4.131, it can be easily shown that $(M^2 - N^2D)$ elements of the state transition matrix are zero and only N^2D elements of the transition matrix are non-zero.

4.8.3 Effective Bandwidth of an HSMM by Reformulating as an HMM

Let y_t be an HSMM, with state process s_t. Following the approach presented in Section 4.8.2, s_t can be reformulated as a Markov chain x_t with larger number of states; Hence, y_t can be modelled as an HMM with modulating (or hidden) process x_t. Then, the effective bandwidth of y_t is easily obtained using the equation 4.128 as presented in Section 4.8.1.

Let $b_i(y_t) = \mathbb{P}(y_t|s_t = i)$ denote the densities of the observation process y_t when y_t is considered as an HSMM. Also let $\varphi_i(\theta)$ be the moment generating function of the observation process, given that the state process is in state i, that is $\varphi_i(\theta) = \mathbb{E}(e^{y_t\theta}|s_t = i)$. Now consider y_t as an HMM with state process

x_t. Let $b'_m(y_t)$ denote the densities of the observation process given that state at time t is $x_t = m$. According to equation 4.133, we have

$$b'_m(y_t) = \mathbb{P}(y_t|x_t = m) \tag{4.136}$$

$$= \mathbb{P}(y_t|s_t = \lfloor \frac{m-1}{D} \rfloor + 1) \tag{4.137}$$

$$= b_{\lfloor \frac{m-1}{D} \rfloor + 1}(y_t) \tag{4.138}$$

Based on this equation, one can easily construct the diagonal matrix $\Phi(\theta)$ (see equation 4.8.2) of size $ND \times ND$ as

$$\Phi(\theta) = diag\{\underbrace{\varphi_1(\theta), \cdots, \varphi_1(\theta)}_{D \text{ times}}, \underbrace{\varphi_2(\theta), \cdots, \varphi_2(\theta)}_{D \text{ times}}, \cdots, \underbrace{\varphi_N(\theta), \cdots, \varphi_N(\theta)}_{D \text{ times}}\} \tag{4.139}$$

By substituting the obtained $\Phi(\theta)$ in equation 4.128, one can easily obtain the effective bandwidth of y_t, $\alpha_y(\theta)$.

4.9 Numerical Results

In this section, we present the numerical results of implementing the methods presented in this chapter. In Section 4.9.1, we present the results of applying the off-line and online identification methods presented in sections 4.6 and 4.7 to synthetic data generated by simulating an HSMM. Our objective is to experimentally verify that our identification method actually finds the true parameter values. As our online identification method results in the same estimate as the off-line method, we only apply the off-line identification algorithm to empirical traffic samples of a few typical TV programs in Section 4.9.2. We also present numerical results of estimating the empirical bandwidth curve for these sources.

4.9.1 Numerical Results for Synthetic Data

To verify that our identification methods give accurate estimates of an HSMM parameters, we conducted two experiments, one for the off-line case, and one for the online case. In our first experiment, the parameters of two HSMM signals, each having $N = 3$ distinct states, were estimated using the off-line algorithm presented in Section 4.6. The number of samples for each model was

Parameter	Actual parameter values for for model 1	Actual parameter values for for model 2	Initial parameter values for for models 1 and 2
$\boldsymbol{A}^o$	$\begin{bmatrix} 0 & 0.30 & 0.70 \\ 0.75 & 0 & 0.25 \\ 0.15 & 0.85 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.30 & 0.70 \\ 0.70 & 0 & 0.30 \\ 0.50 & 0.50 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.00 & 0.50 & 0.50 \\ 0.10 & 0.00 & 0.90 \\ 0.50 & 0.50 & 0.00 \end{bmatrix}$
$\boldsymbol{\mu}$	$\begin{bmatrix} -10 & 0 & 10 \end{bmatrix}'$	$\begin{bmatrix} -10 & 0 & 10 \end{bmatrix}'$	$\begin{bmatrix} -15 & 3 & 15 \end{bmatrix}'$
$\boldsymbol{\sigma}^2$	$\begin{bmatrix} 5 & 5 & 5 \end{bmatrix}'$	$\begin{bmatrix} 10 & 10 & 10 \end{bmatrix}'$	$\begin{bmatrix} 8 & 8 & 8 \end{bmatrix}'$
$\boldsymbol{\mu}_s$	$\begin{bmatrix} 10 & 20 & 30 \end{bmatrix}'$	$\begin{bmatrix} 10 & 20 & 30 \end{bmatrix}'$	$\begin{bmatrix} 10 & 10 & 10 \end{bmatrix}'$
$\boldsymbol{\eta}$	$\begin{bmatrix} 5 & 10 & 15 \end{bmatrix}'$	$\begin{bmatrix} 5 & 10 & 6 \end{bmatrix}'$	$\begin{bmatrix} 1 & 10 & 20 \end{bmatrix}'$
$\boldsymbol{\nu}$	$\begin{bmatrix} 50 & 200 & 450 \end{bmatrix}'$	$\begin{bmatrix} 50 & 200 & 180 \end{bmatrix}'$	$\begin{bmatrix} 10 & 100 & 200 \end{bmatrix}'$

Table 4.2: Actual and initial values of the parameters of HSMM models used in simulating our off-line identification algorithm.

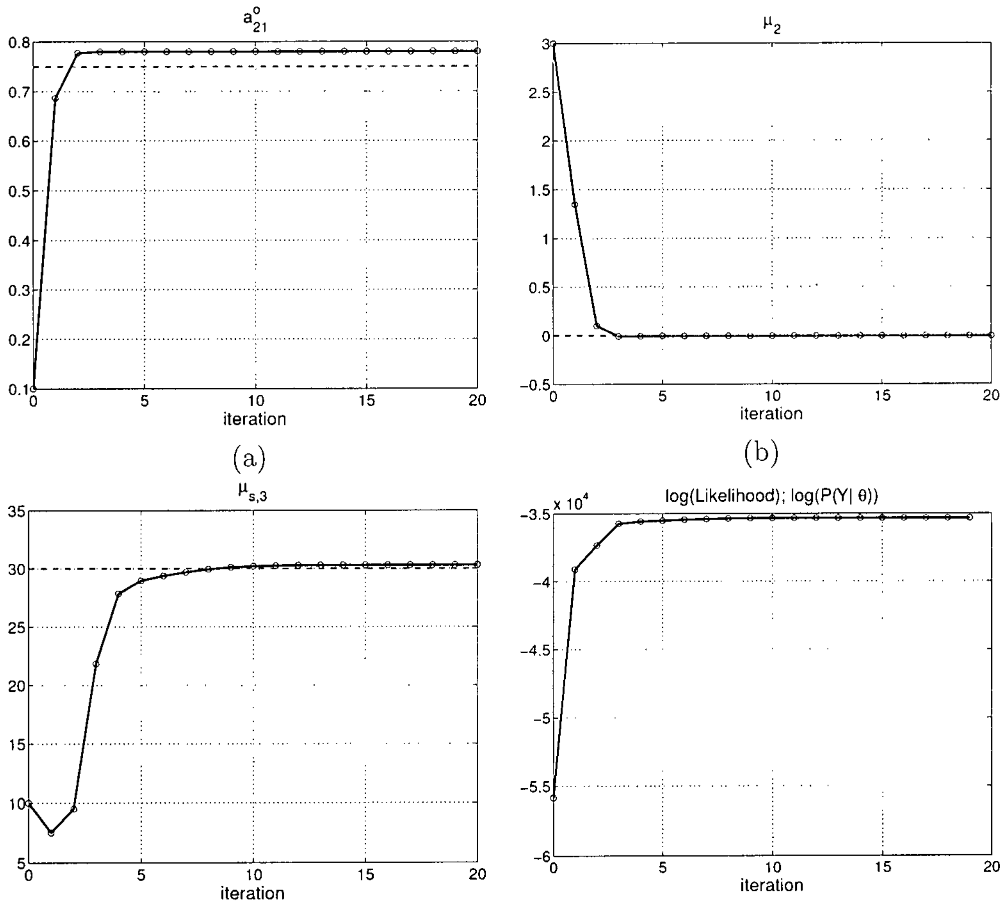
$T = 10000$. The actual values of the parameters are given in Table 4.2. The first model can be considered to be in a low-noise condition (i.e., $|\mu_i - \mu_{i+1}| \gg \sigma_i^2$) while the second model is in high-noise condition. The initial values for the model parameters in both cases are similar, and are shown in Table 4.2. Figures 4.8 and 4.9 illustrate how some of the parameter estimates are updated in each iteration. We observe that the parameter estimates converge to the actual value of the parameters after a few iterations. The log-likelihood of the total observation $\mathcal{Y}_T$ given the parameters estimate, $\log(\mathbb{P}(\mathcal{Y}_T|\boldsymbol{\theta}))$, is also plotted in figures 4.8-d and 4.9-d. As shown, this log-likelihood increases in each iteration, demonstrating that the algorithm finds the maximum-likelihood estimate of the model parameters.

In the next experiment, we applied our method for online identification of HSMMs to two HSMM signals, with the parameters shown in Table 4.3. As shown, the actual parameters of the second model change at $t = 5000$. The results of estimating some of the parameters of these models are presented in figures 4.10 and 4.11, respectively. We observe that the parameter estimates converge to the actual value of the parameter as t becomes large, and that our algorithm successfully tracks the changes in model parameters.

Parameter	Actual parameter values for model 1	Actual parameter values for model 2 $1 \le t \ \le 5 \times 10^3$
$\boldsymbol{A}^o$	$\begin{bmatrix} 0 & 0.80 & 0.20 \\ 0.50 & 0 & 0.50 \\ 0.30 & 0.70 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.50 & 0.50 \\ 0.50 & 0 & 0.50 \\ 0.30 & 0.70 & 0 \end{bmatrix}$
$\boldsymbol{\mu}$	$\begin{bmatrix} -10 & 0 & 10 \end{bmatrix}'$	$\begin{bmatrix} -10 & 0 & 10 \end{bmatrix}'$
$\boldsymbol{\sigma}^2$	$\begin{bmatrix} 2 & 2 & 2 \end{bmatrix}'$	$\begin{bmatrix} 4 & 4 & 4 \end{bmatrix}'$
$\boldsymbol{\mu}_s$	$\begin{bmatrix} 10 & 20 & 30 \end{bmatrix}'$	$\begin{bmatrix} 10 & 20 & 30 \end{bmatrix}'$
$\boldsymbol{\eta}$	$\begin{bmatrix} 5 & 10 & 15 \end{bmatrix}'$	$\begin{bmatrix} 5 & 10 & 15 \end{bmatrix}'$
$\boldsymbol{\nu}$	$\begin{bmatrix} 50 & 200 & 450 \end{bmatrix}'$	$\begin{bmatrix} 50 & 200 & 450 \end{bmatrix}'$

Parameter	Actual parameter values for model 2 $5 \times 10^3 \le t \ \le \times 10^4$	Initial parameter values for models 1 and 2
$\boldsymbol{A}^o$	$\begin{bmatrix} 0.00 & 0.50 & 0.50 \\ 0.15 & 0.00 & 0.85 \\ 0.30 & 0.70 & 0.00 \end{bmatrix}$	$\begin{bmatrix} 0.00 & 0.50 & 0.50 \\ 0.50 & 0.00 & 0.50 \\ 0.50 & 0.50 & 0.00 \end{bmatrix}$
$\boldsymbol{\mu}$	$\begin{bmatrix} -5 & 0 & 10 \end{bmatrix}'$	$\begin{bmatrix} -13 & 4 & 20 \end{bmatrix}'$
$\boldsymbol{\sigma}^2$	$\begin{bmatrix} 4 & 4 & 4 \end{bmatrix}'$	$\begin{bmatrix} 10 & 10 & 10 \end{bmatrix}'$
$\boldsymbol{\mu}_s$	$\begin{bmatrix} 10 & 20 & 30 \end{bmatrix}'$	$\begin{bmatrix} 5 & 10 & 10 \end{bmatrix}'$
$\boldsymbol{\eta}$	$\begin{bmatrix} 5 & 10 & 15 \end{bmatrix}'$	$\begin{bmatrix} 8 & 10 & 20 \end{bmatrix}'$
$\boldsymbol{\nu}$	$\begin{bmatrix} 50 & 200 & 450 \end{bmatrix}'$	$\begin{bmatrix} 40 & 100 & 200 \end{bmatrix}'$

Table 4.3: Actual and initial values of the parameters of HSMM models used in simulating our online identification algorithm.



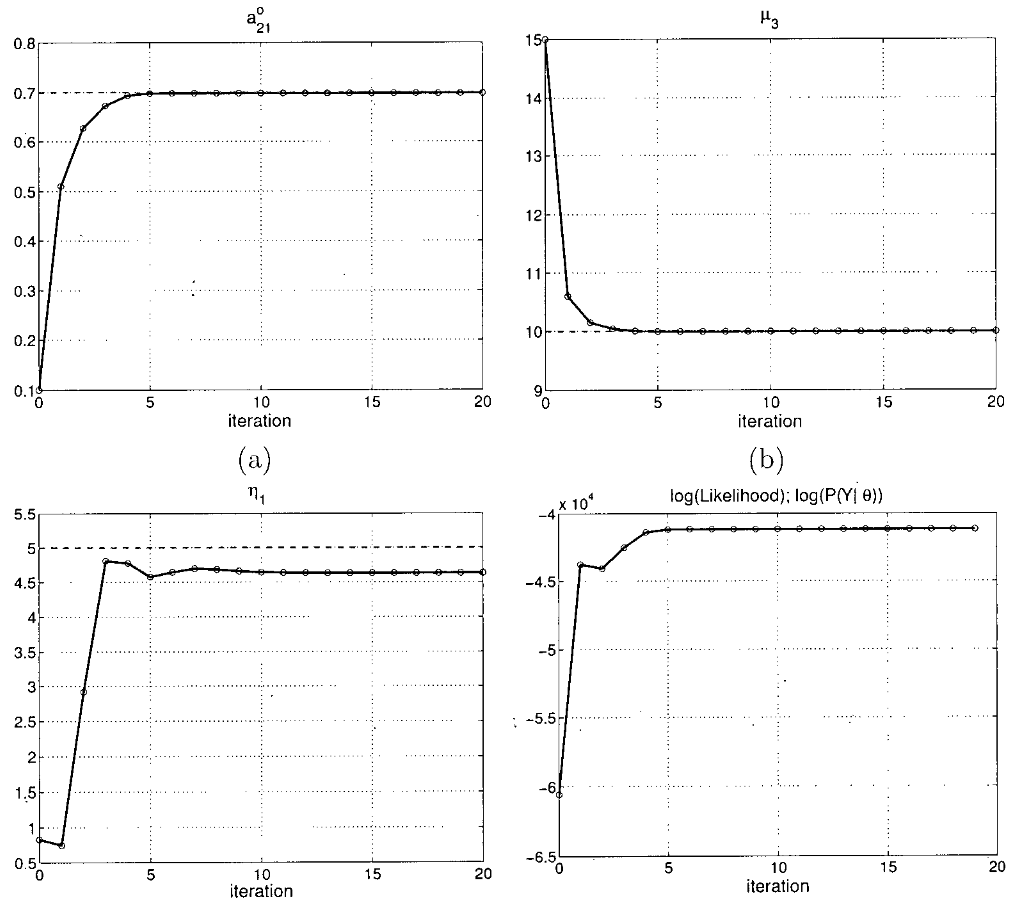
(a)

(b)

(c)

(d)

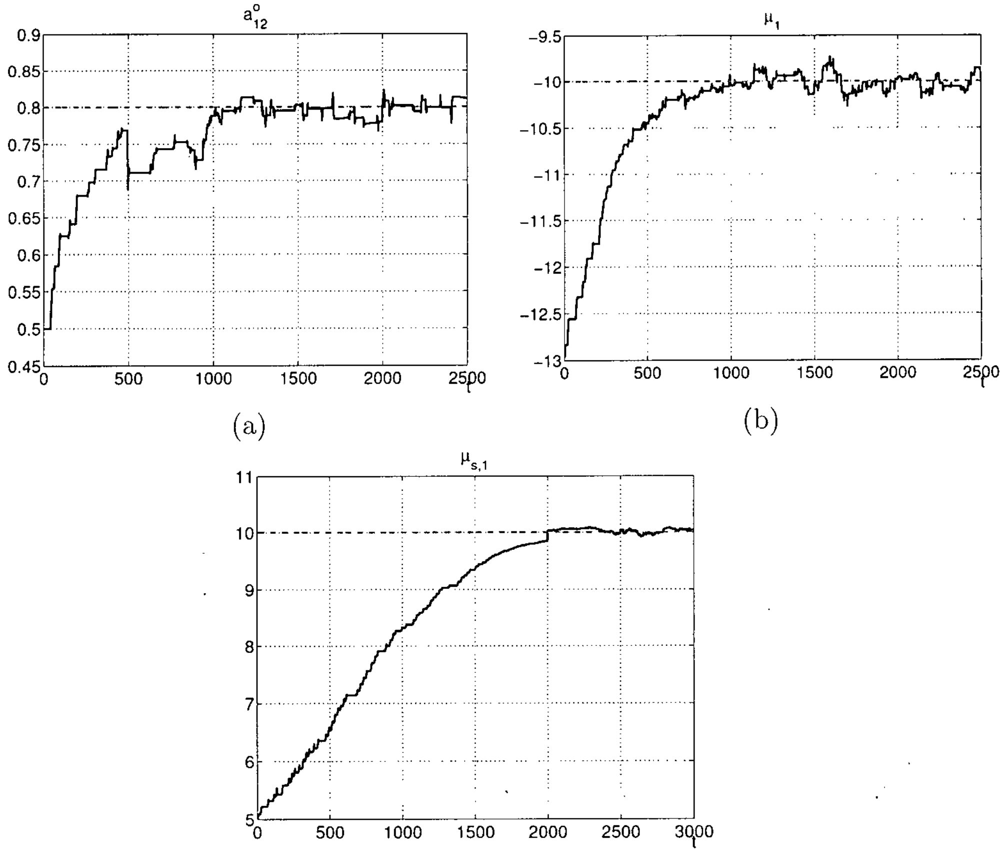
Figure 4.8: a-c) Parameter estimates for the first model versus the iteration number. The dotted lines show the actual value of the parameters. d) The log-likelihood function, $\log(\mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}))$, versus the iteration number. As shown, $\log(\mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}))$ increases in each iteration.



(c)

(d)

Figure 4.9: a-c) Parameter estimates for the second model versus the iteration number. The dotted lines show the actual value of the parameters. d) The log-likelihood function, $\log(\mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}))$, versus the iteration number. As shown, $\log(\mathbb{P}(y_1, y_2, \ldots, y_t|\boldsymbol{\theta}))$ increases in each iteration.

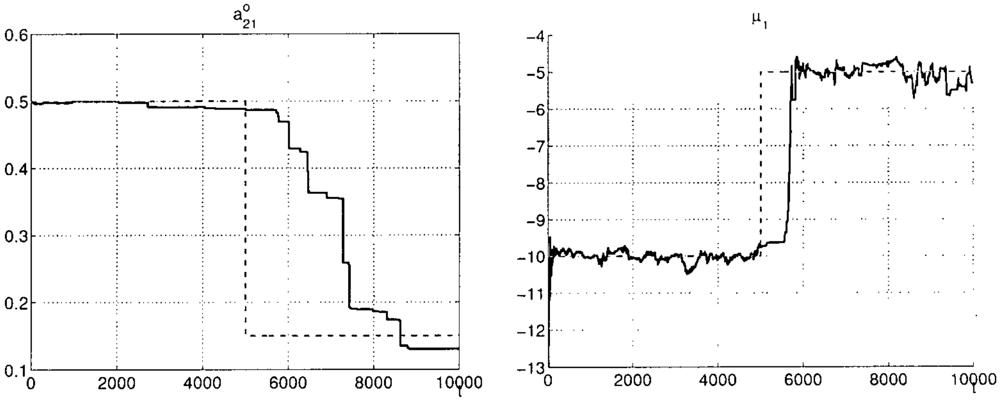


(a)

(b)

(c)

Figure 4.10: Online estimation of a 3 state HSMM: a) state transition probability a^o_{12}; b) observation mean for state 1, μ_1; c) state duration mean for state 1, $\mu_{s,1}$. The dotted line shows the actual value of the parameter. The parameter estimate converges to the actual value of the parameter.



(a) (b)

Figure 4.11: Online estimation of a 3 state HSMM, where the actual parameter changes at $t = 5000$: a) state transition probability a^{o}_{21}; b) observation mean for state 1, μ_1. The dotted lines show the actual value of the parameter. The parameter estimates follow the temporal changes in the actual value of the parameter.

4.9.2 Numerical Results for Empirical TV Traffic Traces

In this experiment, we first fitted an HSMM model to the empirical traffic traces of a few typical TV programs using the offline parameter identification method presented in Section 4.6. The empirical traffics are from the video sequences described in tables 2.4 and 2.5 in Section 2.5. Table 4.4 shows the numerical value of the estimated model parameters for three of the sources. Figures 4.12-4.14 show how the model parameters converge to the final estimate as more iterations of the off-line algorithm are performed.

Then, we obtained the effective bandwidth curve $\alpha(\theta)$ from the estimated HSMM model parameters using the method presented in Section 4.8. Figure 4.15 shows the obtained effective bandwidth curves. As shown, $\alpha(\theta)$ is an increasing function of θ, where $\alpha(0)$ is the average rate and $\alpha(\infty)$ is the maximum rate of the source.

Then, we employed the obtained effective bandwidth curves in our admission control scheme, and plotted the maximum waiting-time T_w versus rate R (see Figure 4.16) for an incidental stream that uses the stochastic service class, and for a constant loss probability p.

In next simulation, we examined the accuracy of our stochastic admission control mechanism. We considered a transmission system, which multiplexes N main video streams, one incidental stream using the deterministic service, and one incidental stream using the stochastic service. We conducted two experiments using two different simulation parameters, as shown in table 4.5. The first set of parameters are selected to simulate cable transmission medium, while the second set simulates a terrestrial medium[2]. The incidental stream using the deterministic service class has the rate R_d. Then, we used the stochastic admission control scheme to obtain the loss probability for an incidental stream with rate R and waiting time T_w that uses the stochastic service class. Finally, we simulated the buffering operations in the transmission system[3], and observed p the loss probability of incidental stream that uses stochastic service class. Figures 4.17-a and 4.17-b illustrate the loss probabil-

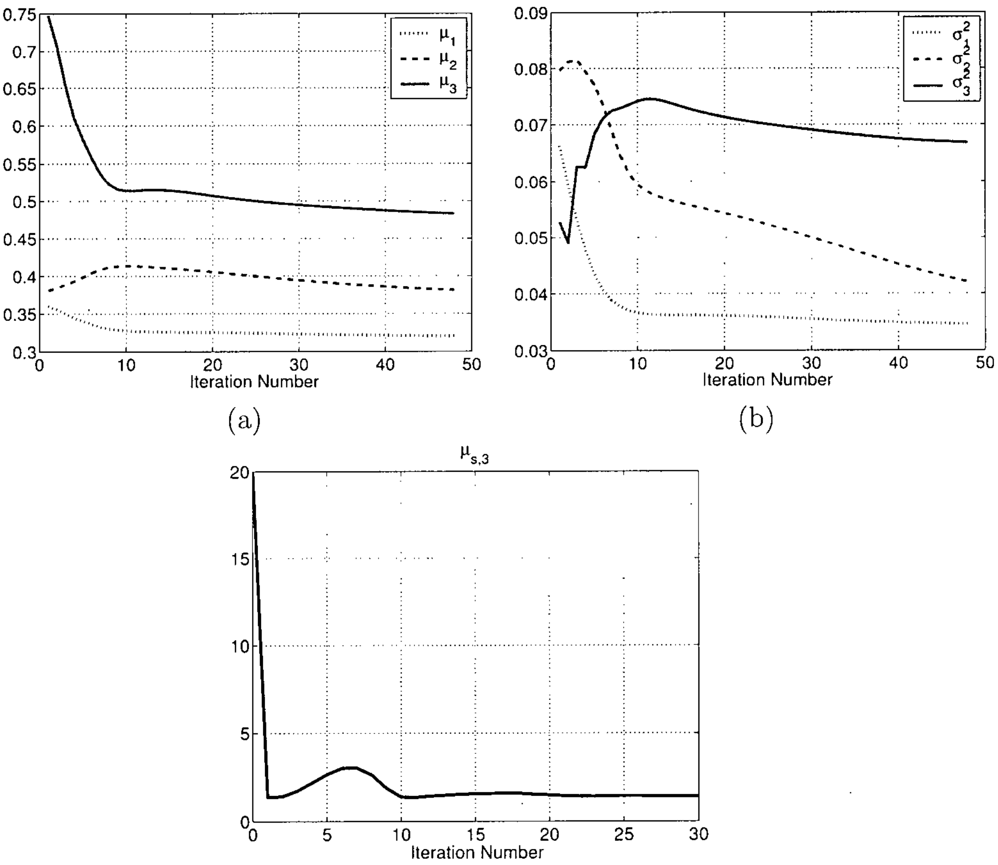
[2] A 6 MHz channel in the cable medium is capable of delivering digital data at the 19.8 Mbps rate. This capacity is usually shared by 4 or 5 TV programs. In terrestrial medium, a 6 MHz channel is capable of delivering at the 39.8 Mbps rate, which is usually shared by 8 or 9 TV programs

[3] The packet scheduling method employed in this simulation is described in detail in Chapter 5.

ity p versus R obtained from simulation and admission control scheme for the simulation parameter sets I and II respectively. As shown, the actual loss probabilities obtained from simulation, shown by the plus signs, are very to the loss probabilities estimated by the admission control scheme. These results verify that our stochastic admission control scheme can accurately estimate the system performance in terms of loss probabilities. Furthermore, it is noted that the admission control is more accurate for a larger T_w parameter. That is the actual loss values are closer the loss probabilities estimated by the admission control in Figure 4.17-a, where $T_w = 30$, than Figure 4.17-b, where $T_w = 5$. This is due to the fact that our admission control scheme is designed based on the assumption that the buffer waiting-time (or the buffer size) is very long.

Parameter	Estimated parameter values for 'Documentary' sequence	Estimated parameter values for 'Talk Show' sequence	Estimated parameter values for 'Mission Impossible' sequence
A^o	$\begin{bmatrix} 0 & 1.0000 & 0.0000 \\ 0.3946 & 0 & 0.6054 \\ 0.0000 & 1.0000 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & .5913 & .4087 \\ 0.5631 & 0 & 0.4369 \\ 0.4694 & 0.5306 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.8067 & 0.1933 \\ 0.9028 & 0 & 0.0972 \\ 0.7441 & 0.2559 & 0 \end{bmatrix}$
μ	$\begin{bmatrix} 0.3883 & 0.7413 & 0.9766 \end{bmatrix}'$	$\begin{bmatrix} 0.3201 & 0.3813 & 0.4828 \end{bmatrix}'$	$\begin{bmatrix} 0.5240 & 0.6351 & 0.9788 \end{bmatrix}'$
σ^2	$\begin{bmatrix} 0.1064 & 0.1160 & 0.0267 \end{bmatrix}'$	$\begin{bmatrix} 0.0346 & 0.0421 & 0.0668 \end{bmatrix}'$	$\begin{bmatrix} 0.2299 & 0.0520 & 0.0363 \end{bmatrix}'$
μ_s	$\begin{bmatrix} 12.7132 & 8.6438 & 12.5120 \end{bmatrix}'$	$\begin{bmatrix} 45.4538 & 6.4677 & 1.4434 \end{bmatrix}'$	$\begin{bmatrix} 5.0961 & 2.0330 & 4.4388 \end{bmatrix}'$
η	$\begin{bmatrix} 0.0768 & 0.1274 & 0.0796 \end{bmatrix}'$	$\begin{bmatrix} 255.2941 & 15.5759 & 7.2841 \end{bmatrix}'$	$\begin{bmatrix} 0.2299 & 0.0520 & 0.0363 \end{bmatrix}'$
ν	$\begin{bmatrix} 0.9761 & 1.1009 & 0.9965 \end{bmatrix}'$	$\begin{bmatrix} 11604 & 101 & 11 \end{bmatrix}'$	$\begin{bmatrix} 28.7533 & 0.4532 & 22.1767 \end{bmatrix}'$

Table 4.4: Estimated parameter values of HSMM model for empirical traffic traces of typical TV programs.

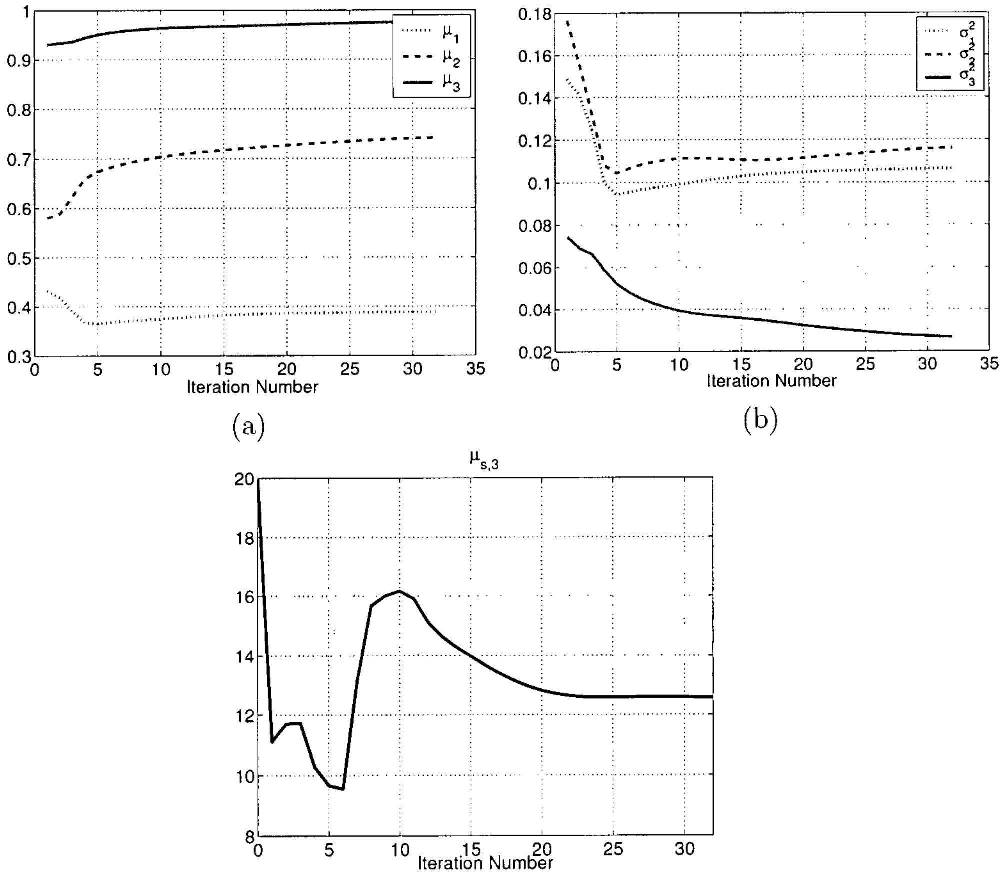


(a)

(b)

(c)

Figure 4.12: Online estimation of HSMM parameters for the 'Talk show' sequence.

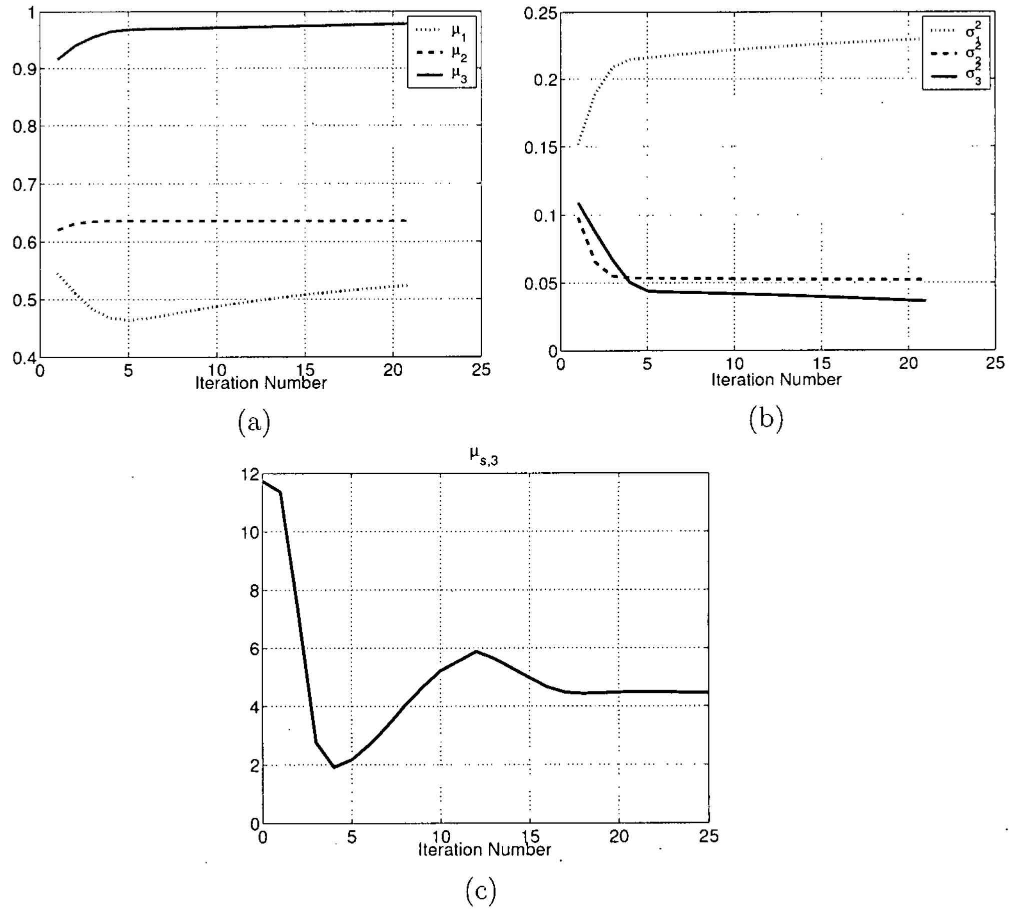


(a)

(b)

(c)

Figure 4.13: Online estimation of HSMM parameters for the Documentary (Best Places in Canada) sequence.



(a)

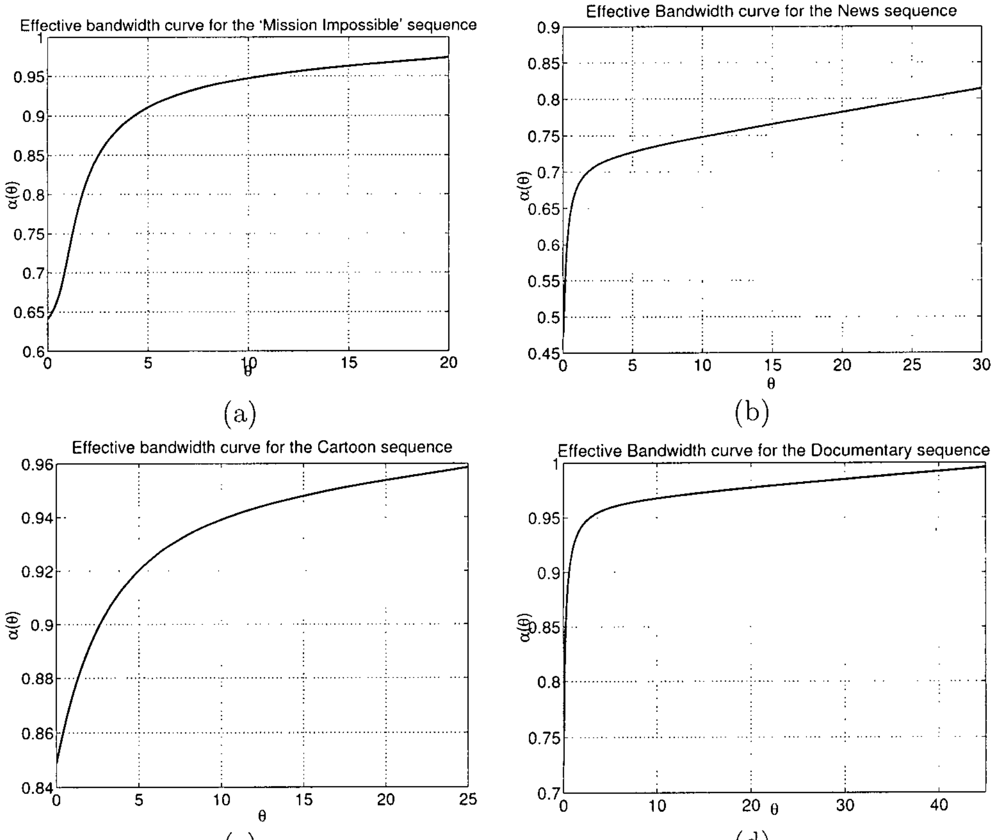
(b)

(c)

Figure 4.14: Online estimation of HSMM parameters for the 'Mission Impossible' sequence.

	Simulation parameter set I (cable medium)	Simulation parameter set II (terrestrial medium)
Transmission rate	19.8 Mbps	39.8 Mbps
Number of TV programs sharing the channel, N	4	8
Maximum bitrate assigned to each main video stream	4.5 Mbps	4.5 Mbps
Transmission capacity reserved for video streams	18 Mbps	36 Mbps
Transmission capacity reserved for audio streams and other ancillary data	1.8 Mbps	3.8 Mbps
Rate of the incidental stream that uses the deterministic service class	0.8 Mbps	1.5 Mbps
Main video stream sources	1. Mission Impossible 2. News 3. Talk Show 4. Documentary	1. Mission Impossible 2. News 3. Talk Show 4. Documentary 5. Court Show 6. Muppets show 7. Soap opera 8. Cartoon

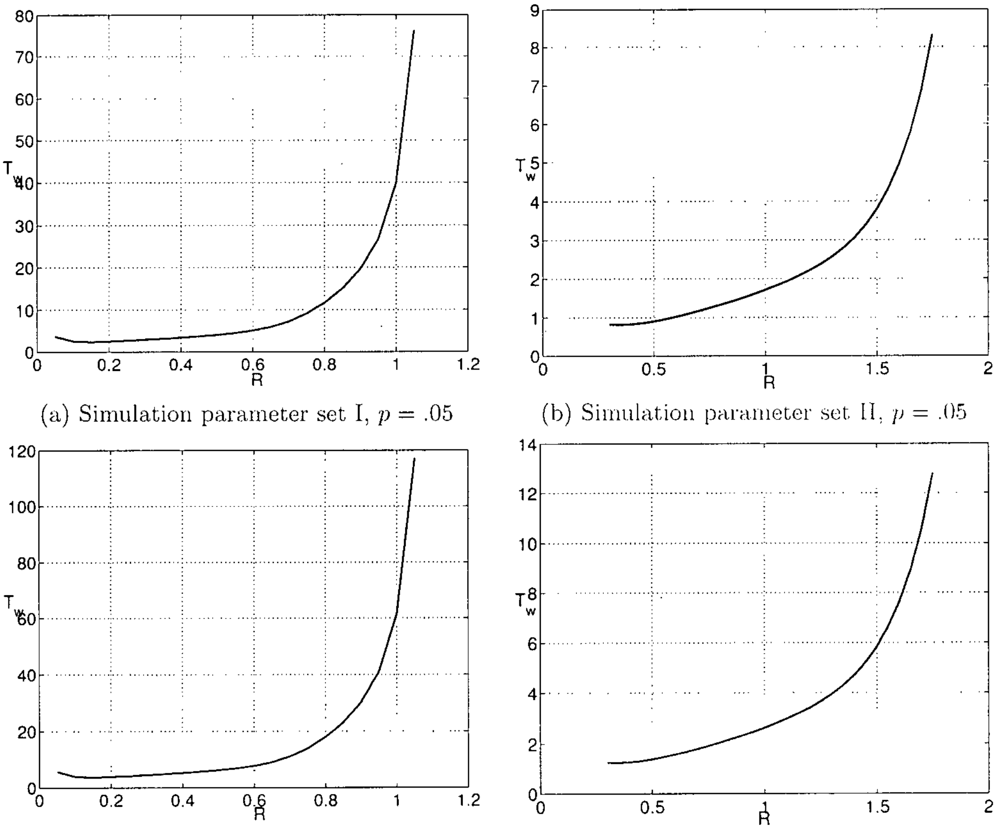
Table 4.5: Simulation parameters.



(a) (b)

(c) (d)

Figure 4.15: Effective Bandwidth curves of video sequences of typical TV programs.



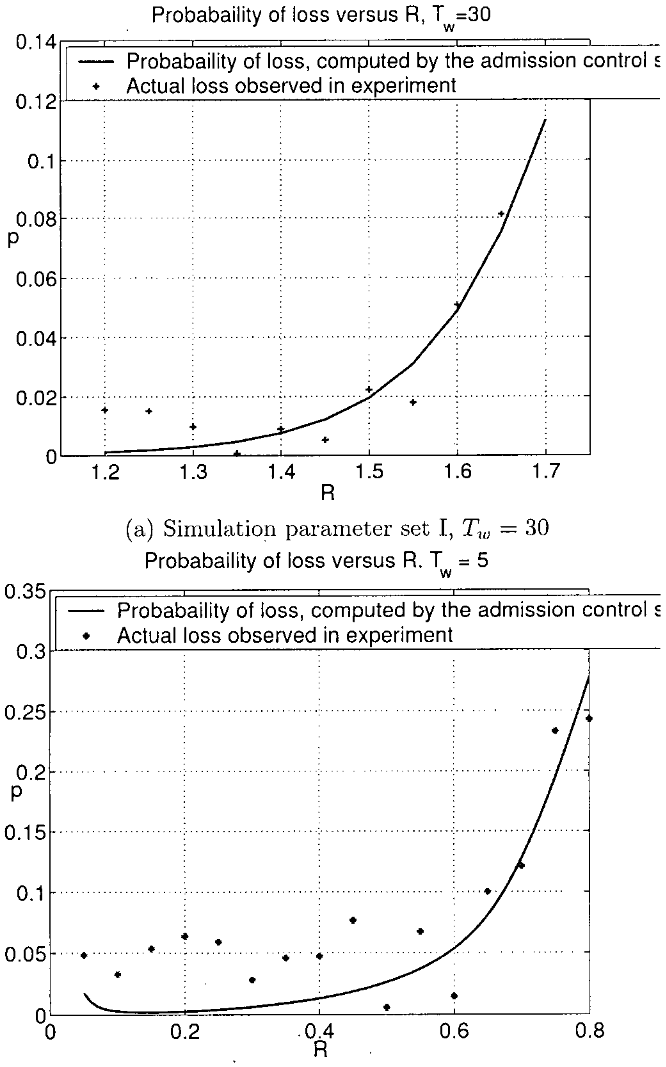
(a) Simulation parameter set I, $p = .05$

(b) Simulation parameter set II, $p = .05$

(c) Simulation parameter set I, $p = .01$

(d) Simulation parameter set II, $p = .01$

Figure 4.16: T_w versus R for an incidental stream that uses the stochastic service class.



(a) Simulation parameter set I, $T_w = 30$

(c) Simulation parameter set II, $T_w = 5$

Figure 4.17: Probability of loss p versus rate R, where T_w is constant. The solid lines illustrate the loss probabilities obtained by the admission control scheme, and '+' signs illustrate the actual loss probabilities obtained from simulating the multiplexing system. As shown, actual loss probabilities obtained from simulation are close to what obtained from the admission control.

4.10 Chapter Conclusion

In this chapter, we studied stochastic traffic models for modelling the traffic of full screen video sources in digital TV. We showed that though HMMs can adequately capture most of the stochastic properties of video traffic sources, they cannot adequately model the state duration densities for the full screen video sequences. Therefore, we selected the more sophisticated HSMM models for modelling the video traffic sources in the proposed ITV application. Furthermore, we gathered some evidence from empirical traffic samples of typical TV programs, which showed that Gamma distribution is a good model choice for modelling the state durations densities.

Next, we presented a novel signal model for HSMMs. We showed that our signal model results in easier and more efficient parameter identification algorithms. Based on our new model, we presented a variant of the EM algorithm for off-line identification of HSMMs. Furthermore, we presented an online identification algorithm based on our new signal model, and based on the general recursive prediction error technique. Using these methods, one can efficiently estimate the parameters of an HSMM for off-line or online cases from the traffic samples .

Next, we showed how the numerical value of the effective bandwidth function is obtained from the parameters of an HSMM. Our approach is based on reformulating an HSMM as an HMM of a higher dimension.

In summary, one can use the methods presented in this chapter to obtain the numerical value of the effective bandwidth function of a source from the traffic samples. The obtained effective bandwidth curve is used in conjunction with the admission control methods presented in Chapter 3 to find the maximum waiting-time for an incidental stream that uses the stochastic service class.

Up to this point (i.e., Chapters 2-4), we have presented methods which find the maximum waiting time for an incidental stream that uses the deterministic or the stochastic service classes. These methods determine when the data packets of an incidental stream should be made available to the transmitter for transmission. In the next Chapter, we describe how the data packets of main and incidental streams are actually handled during multiplexing.

Chapter 5

Broadcast Head-End Architecture

If something anticipated arrives too late it finds us numb, wrung out from waiting, and we feel - nothing at all. The best things arrive on time.

-Dorothy Gilman, A New Kind of Country,
1978.

Overview

A system for multiplexing main and incidental stream data is presented. The role and importance of packet scheduling policy is discussed, and a novel scheduling algorithm is presented. Our approach ensures that all the main and incidental streams are treated according to their importance.

5.1 Introduction

In this chapter, we present our design of the transmitter head-end for our interactive digital TV system. As discussed in Chapter 1, the digital TV standard requires that the encoded video and audio streams of a TV program be delivered to the TV receivers in the form of a single multiplexed stream called 'Transport Stream', (TS). The syntax of transport stream is defined in the digital TV standard. Here, we present our design of a multiplexer system, which is capable of multiplexing incidental streams data alongside the main streams data. Our system takes the priority of the main and incidental streams into account, and manages the flow of the data packets in the system such that: 1) all the main data packets are transmitted on time, 2) all the incidental data which are transmitted using the deterministic service class are transmitted on time, 3) the bandwidth that is not used by the main streams or deterministic service class streams is fairly shared among the incidental streams that use stochastic service class, and 4) the remainder unused bandwidth is fairly shared by the incidental streams that use the best-effort service class.

One crucial part of our multiplexing system is a scheduling algorithm, which determines the order of packets in the interleaved packet sequence that forms the transport stream. We present a scheduling algorithm for our multiplexing system. Our scheduling algorithm employs a prioritizing policy, where input data streams are divided into six different priority classes. Data packets belonging to each class are considered for transmission only if the higher priority classes do not have any data packet ready for transmission. This ensures that more important data (e.g., main streams) are given a higher priority than less important data (e.g., incidental streams). Furthermore, we use a weighted fair queuing policy for scheduling the streams of each priority class. In this policy, the waiting time of each data packet is considered as the key decision factor to decide which packet must be served next. Our approach ensures that the bandwidth is fairly divided among the streams of each priority class. Our scheduling algorithm also ensures that the generated transport stream is compliant with the standard TV receiver model, as indicated by the American digital TV standard ATSC. This ensure that any standard digital TV receiver or set-top box can extract and decode the main streams from the transport stream generated by our system.

The rest of this chapter is organized as follows. In Section 5.2, we present an overview of of standard digital TV multiplexing systems. Then, we present our design of multiplexing system for our interactive TV system in section 5.3. In 5.4, we present our scheduling algorithm. Chapter conclusion is presented in Section 5.5.

5.2 Multiplexing System Structure of Standard Digital TV System

A conceptual diagram of a multiplexer system at the head-end of a TV transmission system is shown in Figure 5.1. This system multiplexes the video and audio streams of a number of TV programs, and creates a multiplexed transport stream. This transport stream is then broadcasted over a cable, terrestrial or satellite channel to the TV receivers. For most TV programs, the source video and audio source sequences are captured at a different location than the transmitter head-end. In this scenario, the source video and audio streams are usually delivered to the transmitter system through a private high-speed link, such as a satellite link. For more details about the standard digital TV transmitter architecture see [93–96].

5.3 Multiplexing System for Our Interactive TV System

Figure 5.2 shows the diagram of our multiplexing system. The data inputs to our multiplexer systems are the video and audio streams, which come from a broadcast station in the case of live programs, or from an off-line storage medium in the case of prerecorded programs. The output is a single transport stream, which has the constant bitrate of R_{TS} bps. As shown, our multiplexer system consists of four basic units: 1) 'TS packetizer' units, which packetize the input stream and generate TS packet streams; 2) 'Traffic Shaping Unit', which passes the TS packets to the multiplex buffers at a *regulated rate*; 3) 'Multiplexing Buffers', which hold the TS packets ready for transmission; and 4) 'Scheduling Unit', which takes the TS packets from the multiplex buffers and creates the multiplexed transport stream. We will discuss the mechanisms of these units in more details in the following sections.

As shown, a 'scalable transcoder' re-encodes an incidental stream to generate a three layer scalable stream. In a general scenario, the base layer is transmitted using the deterministic service class; the first enhancement layer is transmitted using the stochastic service class; and the second enhancement layer is transmitted using the best effort service class. The bitrate of each layer is determined during the admission control process by the admission control

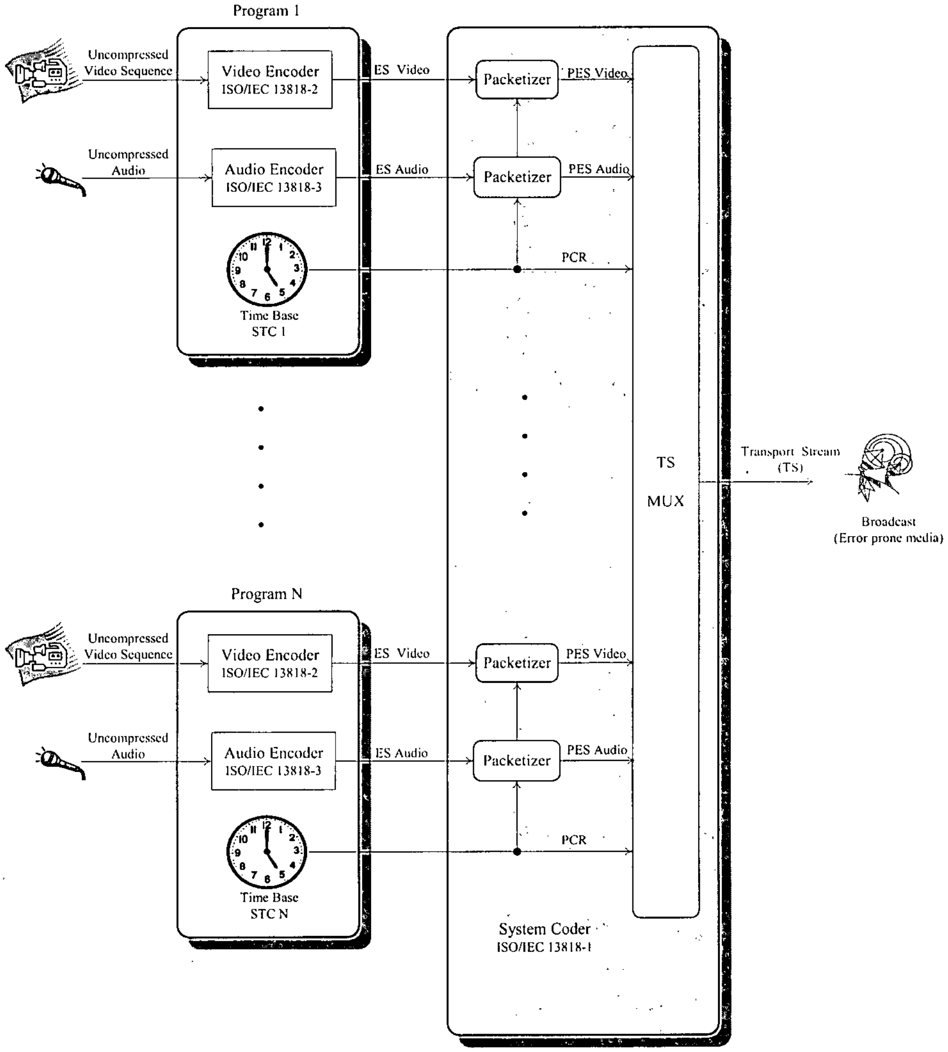

Program 1
Uncompressed Video Sequence
Video Encoder ISO/IEC 13818-2
ES Video
Uncompressed Audio
Audio Encoder ISO/IEC 13818-3
ES Audio
Time Base STC 1
Packetizer
PES Video
Packetizer
PES Audio
PCR
TS MUX
Transport Stream (TS)
Broadcast (Error prone media)
Program N
Uncompressed Video Sequence
Video Encoder ISO/IEC 13818-2
ES Video
Uncompressed Audio
Audio Encoder ISO/IEC 13818-3
ES Audio
Time Base STC N
Packetizer
PES Video
Packetizer
PES Audio
PCR
System Coder ISO/IEC 13818-1

Figure 5.1: Conceptual diagram of a Transport Multiplexer.

unit.

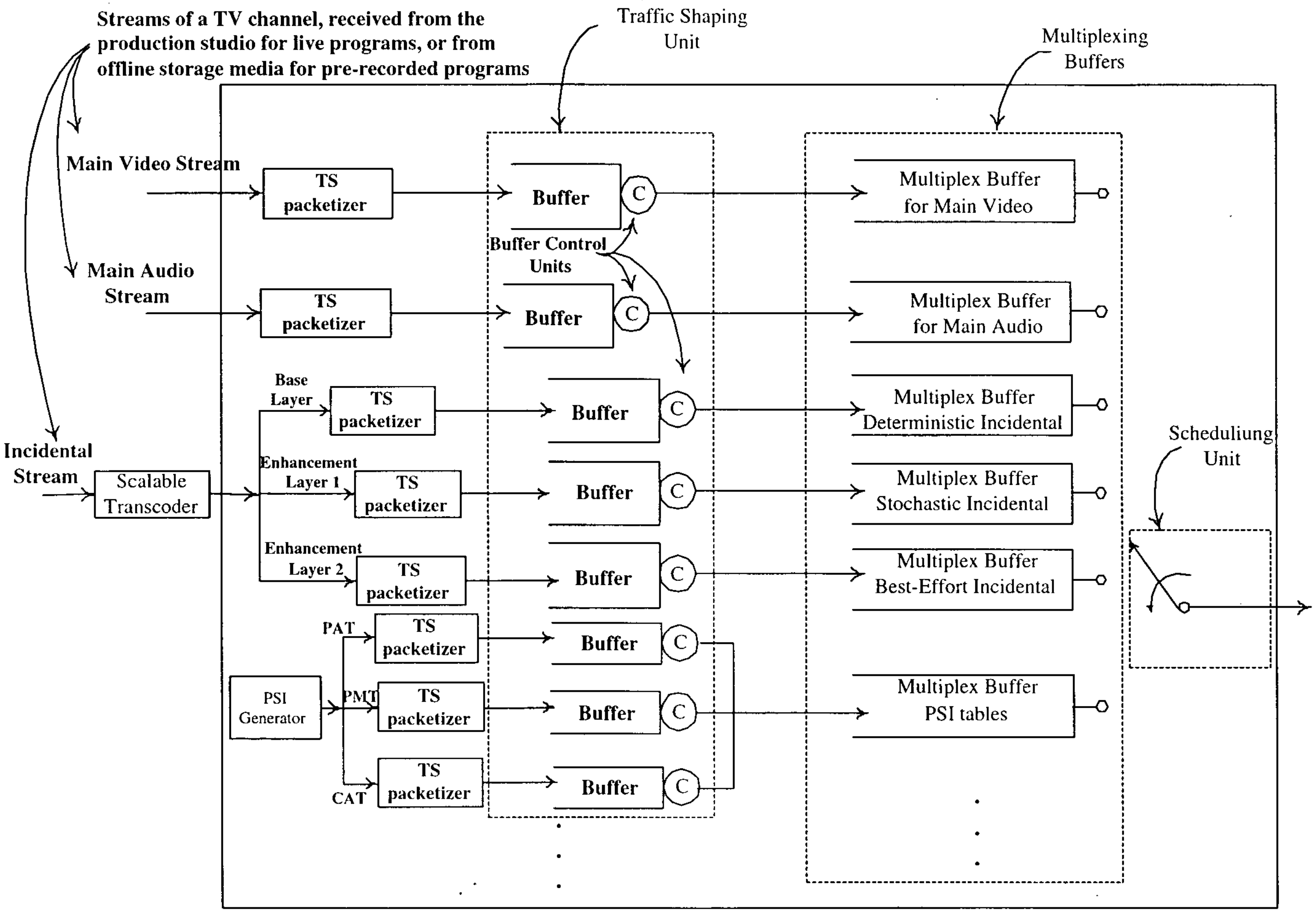


Figure 5.2: Diagram of multiplexing system.

5.3.1 TS packetizer

The TS packetizer units simply break the input bitstream into 184 byte units, add the 4 byte TS packet header as indicated by the standard, and generate the TS packets of size 188 byte. For more information about TS packet structure see [93, 94, 96, 97].

5.3.2 Traffic Shaping Unit

As shown in Figure 5.2, the traffic shaping unit passes the TS packets from the packetizer units to the multiplex buffers. The function of this unit is to *regulate* the rate of packet submission to the multiplex buffers. This rate regulation is necessary to ensure that the actual amount of data submitted to the multiplex buffers is in accordance with the bandwidth reserved for the main streams. Furthermore, the traffic shaping unit controls *when* the incidental data packets are submitted to the multiplex buffers. That is, this unit is responsible for sending the incidental data to the multiplex buffer T_w seconds before their transmission deadline.

As shown, the traffic shaping unit uses a buffer for each stream, and controls the packet departures from each buffer by using a *buffer control unit* for each buffer. These buffer control units use different schemes for each stream, as described below.

Main Audio Streams Main audio streams in digital TV applications have a constant bitrate. If the bitrate of a main audio stream is R_i, then the buffer control unit ensures that no more than $\lceil \frac{R_i}{8 \times 184} \rceil$ TS packets are submitted to the multiplex buffer during each 1 second period, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. For this purpose, the buffer control unit uses a *Token* variable. The token variable is updated periodically every 1 second to $\lceil \frac{R_i}{8 \times 184} \rceil$. Whenever a packet from the buffer is submitted to the multiplex buffers, its token variable is decremented. The buffer control unit sends a TS packet only if the assigned token variable is greater than zero.

Main Video Streams If a main video stream is characterized by its maximum bitrate and not a $(\vec{\sigma}, \vec{\rho})$ model, then a similar scheme to what is used for main audio streams is used by the buffer control units.

However, if a main video stream is characterized by a $(\vec{\sigma}, \vec{\rho})$ model, then the method used by the buffer control unit is different. In this case, the buffer control unit must ensure that the aggregate number of TS packets submitted to the multiplex buffer in any time window of *length* t is less than $min_i(\sigma_i + \rho_i t)$ [1]. Fortunately, there is a very powerful and efficient method for implementing this mechanism based on *Token Buckets.* A token bucket is a mechanism for ensuring that the traffic generated by a source is compliant with a single (σ, ρ) model. A token bucket is simply a variable initialized to σ and incremented at rate ρ. This variable is bounded from above by σ. Whenever a packet is submitted to the multiplex buffer, the token bucket variable is decremented. The buffer control unit submits a TS packet only when the token variable is greater than zero.

For a $(\vec{\sigma}, \vec{\rho})$ model consisting of N (σ, ρ) pairs, N token buckets should be employed. The buffer control unit submits a TS packet only when the minimum of all the N token variables is greater than zero. This mechanism ensure that the total traffic delivered to multiplex buffer complies with the $(\vec{\sigma}, \vec{\rho})$ model.

PSI tables Program Specific Information (PSI) tables are data tables embedded into the transport stream, which contain important information necessary for demultiplexing the transport stream [93,95,97]. For example, PSI tables carry the so called 'identification numbers', which tell the digital TV receivers which packets should be decoded for a specific TV program. There are four types of PSI tables: *Program Allocation Table* (PAT), *Program Map Table* (PMT), *Conditional Access Table* (CAT) and *Private Tables* [95,97]. Since PSI tables carry information necessary for decoding the transport stream, it is necessary that PSI tables are repeatedly inserted into the transport stream. The repetition frequency of PSI data is not specified by the MPEG standard. However, it is advised that the PSI tables be repeated between 10 to 50 times per second.

We employ a token variable for controlling the transmission frequency of each PSI table. These token variables are updated periodically according

[1] Note that the variables σ and ρ should be translated from bits and bps to 'TS packet count' and 'TS packets per seconds' by dividing them to 8×188 and rounding towards infinity.

to the repetition frequency of the PSI tables and the actual size of the PSI table in bytes. The buffer control unit of PSI tables submits a TS packet to the multiplex buffer only when the assigned token variable is greater than zero.

Incidental Streams The function of buffer control unit for incidental streams is to send the incidental TS packets to the multiplex buffer T_w seconds before their transmission deadline. Since incidental streams have a constant bitrate, this mechanism is easily implemented by sending the first TS packet of an incidental stream exactly T_w seconds before the decoding time of the first frame of the incidental stream; the consecutive TS packets are submitted to the multiplex buffer at the constant rate of the stream. For example, consider an incidental stream with maximum waiting time T_w and rate R bps, which is equivalent to $R/(8 \times 188)$ TS packets per second. Also suppose that the decoding of this stream should start at time T. Then the buffer control unit will send the first TS packet of this stream at $T - T_w$ to the multiplex buffer; and each consecutive TS packet is transmitted after $(8 \times 188)/R$ seconds.

5.3.3 Multiplex Buffer

Multiplex buffers hold the TS packets that are ready for transmission. The size of multiplex buffers for main streams is selected such that these buffers never overflow. A buffer of size $.5 \times R$, where R is the maximum bitrate of the stream in bps is usually enough. The size of multiplex buffers for incidental streams is $T_w \times R$, where T_w is the maximum waiting time assigned to the incidental stream during the admission control process.

5.4 Scheduling

5.4.1 Scheduling Unit Objective

Suppose the bitrate of transport stream is R_{TS} bps. Since each TS packet has the constant size of 188 bytes, then each TS packet is transmitted in exactly $\Delta = \frac{188 \times 8}{R_{TS}}$ seconds. We call Δ a *transmission time-slot.* That is, a transmission time-slot represents the time required for sending 188 bytes of data. Hence a

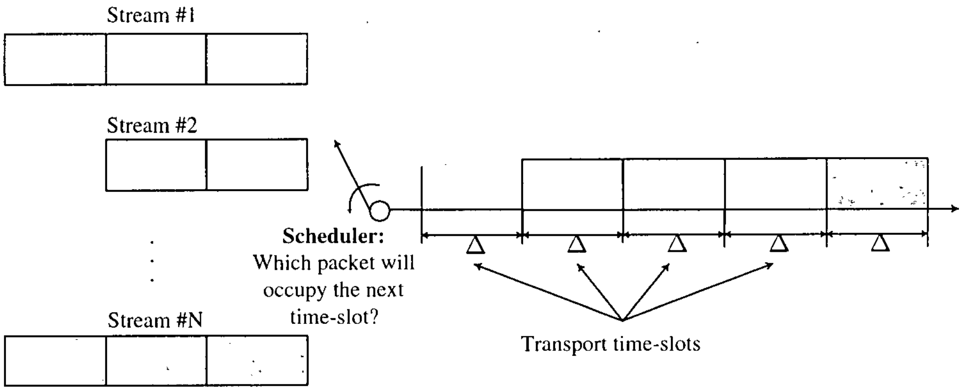


Figure 5.3: Scheduling unit decides which TS packet should be sent at the next transmission time-slot.

free time-slot represents the opportunity of sending only one TS packet. The function of the scheduling algorithm is to assign each transmission time-slot in the TS stream to one of the input streams, as shown in Figure 5.3.

5.4.2 Prioritizing Policy

Our algorithm employs a prioritizing policy, where the input streams are divided to different priority classes. Each class of streams is served only when higher priority streams do not have any data packet ready for transmission. In priority order, these classes are: 1) PSI tables, 2) main audio streams, 3) main video streams, 4) incidental streams with deterministic service class, 5) incidental streams with stochastic service class, and finally 5) incidental streams with best effort service class.

5.4.3 Limitations Imposed on Scheduling by the Digital TV Standard

The digital TV standard has defined a reference model for the buffering process in digital TV receivers, called 'Transport Stream System Target Decoder' or TS-STD. This reference model specifies a standard layered buffering structure required to de-multiplex and decode a transport stream [93–96]. It also specifies the minimum size of each buffer and how data flows between the buffers. The purpose of this reference model is to standardize the buffering process at TV receivers. All digital TV receivers are required to implement the TS-STD, and all the transport streams must be generated in compliance with this reference model.

Hence, our transport stream multiplexer must employ a mechanism to ensure that the generated transport stream is compliant with the reference target decoder. This function is performed by the scheduling unit. That is, the scheduling algorithm should ensure that assigning the current transmission time-slot to a specific stream will not result in a buffer overflow in the reference TS-STD model. We implement this mechanism by simulating the TS-STD model for each TV channel. Using the simulated TS-STD model, we first check that assigning the current time-slot to a stream does not result in a buffer overflow in the TS-STD model.

5.4.4 Scheduling Algorithm

As discussed before, the function of the scheduling algorithm is to decide which stream should occupy the next transmission time-slot in the transport stream.

The mechanisms of our scheduling algorithm can be broken into two conceptual steps. In the first step, our algorithm creates a set of *candidate* streams. The candidate streams are selected by selecting the streams that 1) whose multiplex buffer is not empty, 2) the TS-STD model allows them to be transmitted at the current time-slot, and 3) they have higher-priority than other streams that satisfy the first two conditions. Therefore, candidate streams are all selected from the same priority class, and are all eligible for transmission at the current time-slots.

In the second step, the algorithm selects one stream among the candidate streams for transmission. Depending on the type of the candidate streams, we use different policies to decide which candidate stream should be transmitted. For PSI tables data, a simple round robin policy is used. For main streams (either video or audio), let W_i denote the buffer workload and R_i the maximum rate of the stream. Then, we select the stream for which W_i/R_i is maximum. For incidental streams (either deterministic, stochastic or best effort services), we use an 'Earliest Deadline First' (EDF) policy. Let W_i be the buffer workload, R_i the stream rate and T_i the maximum waiting time

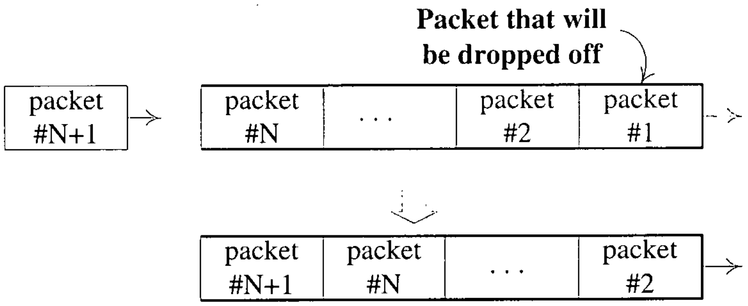


Figure 5.4: Packet drop off in Multiplex buffers of stochastic and best-effort service classes. When a new packet arrives to a full buffer, it is *pushed* into the buffer.

assigned to the stream. Then we assign the deadline

$$d_i = T_i - \frac{W_i}{R_i} \tag{5.1}$$

to each stream. Note that a small d_i means that data packets in the buffer are close to passing their maximum waiting time. The scheduler selects the stream that has the smallest d_i.

5.4.5 Packet Drop-Off

The admission control schemes, along with the mechanisms used by the traffic shaping unit, ensure that the multiplex buffers of PSI tables, main streams, and incidental streams using deterministic service will never overflow. That is, these mechanisms ensure that the aggregate number of TS packets submitted from these stream to the multiplex buffer during each scheduling cycle is less than or equal the number of packets that can be transmitted. Therefore, we expect to experience no packet drop off for these streams.

However, we anticipate that the multiplex buffers of the incidental streams with stochastic and best effort service classes overflow occasionally. This overflow occurs when the transmission line is committed to the main and incidental streams with deterministic service class for a long time, and the scheduler cannot send enough TS packets from the incidental streams with stochastic or best effort service classes. In this case, the exceeding TS packet in the multiplex buffer should be dropped off. This is shown in Figure 5.4.

As shown, when a new TS packet arrives to a full multiplex buffer, the new packet is *pushed* into the buffer. That is, a packet from the buffer beginning is dropped, other packets are shifted, and the new packet is added to the buffer end. The reason that we drop the first packet from the beginning of the buffer, and not the new packet, is that first packet has been in the buffer for more than the assigned maximum waiting time T_w, and hence it is too late to transmit this packet. Note that the multiplex buffer size is $T_w \times R$, and is filled at the constant rate R.

5.5 Chapter Conclusion

In this Chapter, we designed a multiplexer system for the transmitter head-end of our interactive TV system. We described the buffering structure required for handling the main, incidental and other ancillary data packets. Then, we presented a novel scheduling scheme for controlling the multiplexing operations. Our scheduling method ensures that all the main and incidental streams data packets are treated according to their importance during the multiplexing process.

Furthermore, our scheduling algorithm employs a technique which ensures that the broadcasted stream is backward compatible with conventional digital TV receivers. This guarantees that conventional digital TV receivers, which are not programmed for our interactive TV system, are able to display the main video and audio content without any discrepancy.

Chapter 6

Thesis Summary

In this thesis, we proposed and defined a new interactive system for digital TV. This system gives TV viewers the freedom to control TV program content. In so doing, we have introduced a new technological concept, which improves the home entertainment technology.

We then addressed the most challenging issue involved in the design of the proposed interactive TV system. This issue concerns adding extra incidental data to a digital TV transmission channel. This must be accomplished without increasing the bandwidth or degrading the quality of other programs. We then presented data transmission schemes for our interactive TV system that allows to transmit the incidental data. We efficiently took advantage of any unused bandwidth in the transmission channel to transmit the incidental data. We classified the transmission schemes of incidental data into three classes, *deterministic, stochastic,* and *best-effort* service classes.

We proposed to use scalable video coding for the incidental streams. In this approach, an incidental stream is encoded to a three-layer scalable stream. The base, first enhancement and second enhancement layers are transmitted using the deterministic, stochastic and best-effort service classes respectively. This technique not only results in very efficient bandwidth utilization, but also improves the perceived picture or audio quality of incidental streams.

We then designed an admission control scheme for the deterministic and stochastic service classes. These admission control schemes answer the crucial question of whether an incidental stream can be added to a TV program or not. Our approach in designing the admission control schemes was based on modelling the traffic of main video streams using a traffic model. This model is then used for designing the admission control test.

In the case of the deterministic service class, we employed the $(\vec{\sigma}, \vec{\rho})$ model for modelling the traffic of main main streams. We developed methods for fitting the $(\vec{\sigma}, \vec{\rho})$ model to traffic sources. These methods are more efficient and more accurate than previously available methods. In so doing, we helped to advance the knowledge in the deterministic traffic modelling field. We then adapted the 'Network Calculus' theory, and designed an efficient admission control scheme for the deterministic service class.

For the stochastic service class, we employed Hidden Semi-Markov Models (HSMM) for modelling the traffic of main video streams. We developed efficient methods for the identification of HSMM model parameters for off-line and online cases. In so doing, we have advanced existing knowledge about the general semi-Markovian signal models, off-line and online identification of HSMMs, and stochastic traffic modelling of full-screen video sources. Using the 'Effective Bandwidth' theory, we then designed an efficient admission control scheme for the stochastic service class.

Then, we presented our design of a data multiplexer for the transmitter head-end of our interactive digital TV system. Our design is capable of multiplexing incidental stream data alongside the main streams data. We described how the flow of main and incidental data packets are controlled during the multiplexing process. We presented a novel scheduling scheme, which determines the order of data packets in the broadcasted packet sequence. Furthermore, we employed mechanisms which ensure that the conventional digital TV receivers can extract and display main video and audio content from the multiplexed stream. This makes our system backward compatible with the presently existing conventional digital TV receivers.

We have tested the validity and efficiency of the methods presented in this thesis via simulation experiments. Numerical results of these experiments are presented throughout the thesis.

In summary, this thesis presents efficient data transmission schemes for transmitting extra video and audio content alongside conventional digital TV data. By exploiting the methods presented in this thesis, new interactive TV applications are enabled, and the home entertainment technology is advanced. Furthermore, some research results presented herein, can benefit other research areas, such as deterministic traffic modelling, QoS enabled data networks, and semi-Markovian stochastic models.

6.1 Thesis Contributions Summary

The major contributions of this thesis are summarized as follows, where they are listed in the order of appearance in the thesis.

- **New interactivity concept:** We defined a new interactivity concept for TV, which allows TV viewers to personalize the video or audio content of TV programs. This new interactivity concept drastically enhances TV viewers experience, and advances the home entertainment technology.

- **Data transmission before presentation time:** We developed a novel transmission technique for transmitting the incidental data units ahead of their presentation time. This technique allows us to take optimal advantage of the transmission bandwidth that is unoccupied by the main streams.

- **Deterministic admission control:** A new admission control scheme was developed in chapter 2 to be used in the deterministic service class. This is the most important line of development of this thesis in the context of the deterministic service class.

- $(\vec{\sigma}, \vec{\rho})$ **Model fitting:** The algorithm presented in chapter 2 for fitting $(\vec{\sigma}, \vec{\rho})$ model to a traffic source is one of the contributions of this thesis. This algorithm is useful in any application that employs $(\vec{\sigma}, \vec{\rho})$ model.

- **Physical interpretation of effective bandwidth:** A new physical interpretation of effective bandwidth is offered in chapter 3. Such interpretation is important because it helps in advancing the stochastic queuing theory.

- **Stochastic admission control:** A new admission control scheme was developed in chapter 3 to be used in the stochastic service class. This is the most important line of development of this thesis in the context of the stochastic service class.

- **Employing HSMM for modelling the full-screen video traffic:** We showed in chapter 4 that HSMMs are a better model choice than HMMs for modelling the stochastic properties of full-screen high bitrate video. This line of development advances the video modelling field.

- **New signal model and identification algorithms for HSMM:** We presented a new signal model for HSMMs in chapter 4. This model results in more efficient model parameter identification algorithms. We also presented off-line and online parameter identification algorithms based on our new signal model. The new signal model and identification algorithms are useful in any application that employs HSMMs, and are one of the most important contributions of this thesis.

- **Effective bandwidth of HSMMs:** In chapter 4, we showed for the first time how to obtain the effective bandwidth of an HSMM signal. This line development is useful in any queuing application that uses HSMM traffic.

6.2 Future Research

In this thesis, we mainly focused on the mechanisms used at the transmitter head-end. Obviously, in order to display an incidental stream, a digital set-box receiver is required, which should be specifically designed and programmed for the proposed ITV application. In the context of this thesis, a set-top box is considered as a black box equipped with a large buffer (e.g., a hard disk) for caching the incidental stream data. This set-top box is assumed to be capable of controlling the playback of main and incidental streams. Though, the design concepts for the set-top box receiver architecture are simple, there is room for improvement. Thus, it is beneficiary to exploit the set-top box architecture in more detail in future research. For example, one can improve the buffer management schemes used at the receiver end for controlling the cashing of the interactive content, such that the incidental data is not lost when a TV viewer switches channels, and such that the random access delay for incidental streams is reduced. One can also improve the user interface (e.g., menus where TV viewers select their choices about a TV program), such that TV viewers can interact with the TV program more efficiently, and navigate among the main and incidental streams easier.

Furthermore, future research may improve the traffic models used in this thesis. For example, one can exploit the possibility of using stochastic traffic models other than HSMMs, such as self-similar models, or Transform-Expand-Sample (TES) traffic models, for modelling the traffic of main video sources.

This can result in easier, more efficient, or more accurate traffic models.

Appendix A

Proof of Theorem 1 in Section 2.4

Theorem 1 *Assume a traffic source constrained by $A^*_{in}(t)$ traverses a system that offers the service curve $\beta(t)$. The waiting time $d(t)$ for all t satisfies: $d(t) \leq h(A^*_{in}, \beta)$ [35, 36].*

Proof It follows from the definitions of $d(t)$ (equation 2.28) and $h(A_{in}, A_{out})$ (equation 2.31) that

$$\tau \leq h(A_{in}, A_{out}) \iff A_{in}(t-\tau) \geq A_{out}(t) \qquad \text{(A.1)}$$

Now consider some fixed t. From the definition of $d(t)$, for all $\tau \leq d(t)$ we have

$$A_{in}(t) \geq A_{out}(t+\tau) \qquad \text{(A.2)}$$

Now the service curve property at time $t+\tau$ (equation 2.29) implies that there is some s such that

$$A_{out}(t+\tau) \geq A_{in}(t+\tau-s) + \beta(s) \qquad \text{(A.3)}$$

So, from A.2 and A.3 we have

$$A_{in}(t) \geq A_{in}(t+\tau-s) + \beta(s) \qquad \text{(A.4)}$$

It follows from this equation that $A_{in}(t) > A_{in}(t+\tau-s)$, which implies that $t > t+\tau-s$. Thus,

$$\beta(s) \leq A_{in}(t) - A_{in}(t+\tau-s) \leq A^*_{in}(s-\tau) \qquad \text{(A.5)}$$

From the definition of $h(A^*_{in}, \beta)$ (see A.1) and A.5 it follows that $\tau \leq h(A^*_{in}, \beta)$. Since this is true for all $\tau < d(t)$, we conclude that $d(t) \leq h(A^*_{in}, \beta)$, Q.E.I.

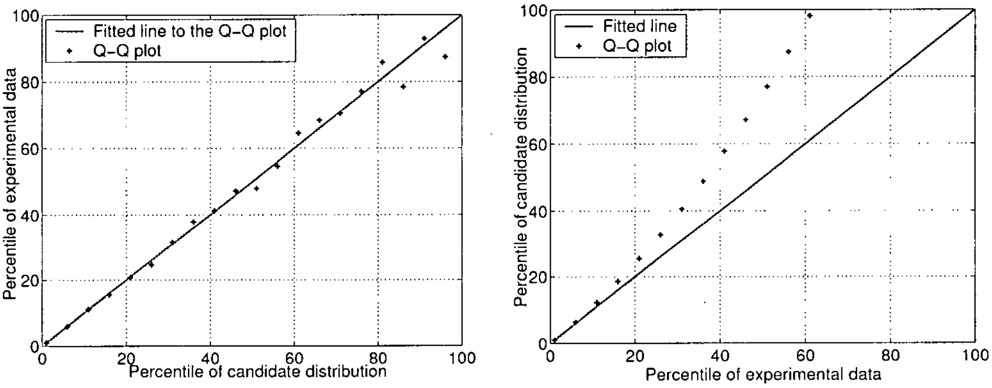
Appendix B

Q-Q Plot

The 'Quantile-Quantile' (Q-Q) plot, also known as 'probability plot' is a graphical technique for assessing whether or not an experimental data set follows a given distribution such as the normal or Weibull [98–100]. This technique is also used for determining if two data sets come from populations with a common distribution. By a 'quantile', we mean the fraction (or percent) of points below the given value. That is, the 0.3 (or 30%) quantile is the point at which 30% percent of the data fall below and 70%fall above that value.

A Q-Q plot is a plot of the quantiles of an experimental data set against the quantiles of the assumed distribution. If Q-Q plot is used to determine if two data sets come from the same distribution, then quantiles of the first data set are plotted against the quantiles of the second second data set. If the experimental data are actually from the assumed theoretical distribution, then the points in Q-Q plot should form approximately a straight line. This case is illustrated in Figure B.1-a. In this figure, the vertical axis is the quantile of the experimental data, and the horizontal axis is the quantile of a candidate probability distribution. As shown, the points in this Q-Q plot are very close to form a line. This indicates that the experimental data are actually drawn from a population with the assumed distribution. However, departures from this straight line indicate departures from the specified distribution. This is illustrated in Figure B.1-b. Hence, one can use the correlation coefficient associated with the linear fit to the data in the Q-Q plot as a measure of the goodness of the fit.

In practice, Q-Q plots can be generated for several competing distributions to see which provides the best fit. In this case the probability plot generating the highest correlation coefficient is the best choice since it gener-



(a)

(b)

Figure B.1: Q-Q plot

ates the straightest probability plot.

Appendix C

Likelihood Ratio Test

C.1 General Likelihood Test

Let $\mathbf{S}_T = \{s_1, s_2, \cdots, s_T\}$ be samples from a stochastic model, and let $\boldsymbol{\theta}$ denote the model parameters, which takes it values from the space Ω. Using the maximum likelihood principle, one can estimate the model parameters by finding $\hat{\boldsymbol{\theta}}$ such that $L(\boldsymbol{\theta}) = \mathbb{P}(\mathbf{S}_T|\boldsymbol{\theta})$ is maximized. This can be regarded as finding the 'best' explanation for the observed $\mathbf{S}_T$.

Now suppose one wishes to test whether some of the model parameters are *restricted* or not, for example, if some of the model parameters are bounded or if some of the model parameters are zero. Formally, this is denoted by testing if $\boldsymbol{\theta} \in \omega$, where ω is a subspace of Ω. The usual approach to this problem is based on the *likelihood ratio* concept, which is defined as

$$\Lambda(\mathbf{S}_T) = \frac{\sup\limits_{\omega} L(\boldsymbol{\theta})}{\sup\limits_{\Omega} L(\boldsymbol{\theta})} \tag{C.1}$$

That is, $\Lambda(\mathbf{S}_T)$ is the ratio of the best chance of observing $\mathbf{S}_T$ for $\boldsymbol{\theta} \in \omega$ to the best chance of observing $\mathbf{S}_T$ for $\boldsymbol{\theta} \in \Omega$. Since $\omega \subset \Omega$, then Λ is always between 0 and 1. Values of Λ close to 1 suggest that the data are very compatible with $\boldsymbol{\theta} \in \omega$. That is, $\mathbf{S}_T$ is explained almost as well by the parameter estimates under $\boldsymbol{\theta} \in \omega$ as by parameter estimates under $\boldsymbol{\theta} \in \Omega$. For these values of Λ we should accept $\boldsymbol{\theta} \in \omega$. Conversely, if Λ is close to 0, then the data would not be very compatible with $\boldsymbol{\theta} \in \omega$ and it would make sense to reject $\boldsymbol{\theta} \in \omega$. This is the rationale behind the *likelihood ratio test*. A likelihood ratio test is a hypothesis test for testing

$$\text{H0: } \boldsymbol{\theta} \in \omega \tag{C.2}$$

against

$$\text{H1: } \boldsymbol{\theta} \in \Omega, \qquad \omega \subset \Omega \tag{C.3}$$

In order to obtain the rejection region and confidence intervals, it is necessary to know the distribution of Λ. However, this is ordinarily very complicated. Fortunately, it is shown that under very general conditions $-2\ln(\Lambda)$ has a χ^2 distribution with n degree of freedom, where n is the difference in the dimension of ω and Ω. Hence, by comparing $-2\ln(\Lambda)$ to the upper $100\times(1-\alpha)$ percentile point of a χ^2 distribution, one can decide to reject $H0$ or not. α is known as the significance level of the test, and is usually selected $\alpha = 5$.

REMARK As a general concept in hypothesis testing, the $H1$ hypothesis represent a more general case (or more complex concept) than $H0$. In these cases, the $H1$ hypothesis is adopted unless there is sufficient evidence to reject the special (or simple) hypothesis $H0$. This concept is conveyed in the test by the notion of $\omega \subset \Omega$.

C.2 Likelihood Ratio Test for testing the HMM model against HSMM model

Suppose we have two signal model candidates for modelling an empirical sequence $\mathbf{S}_T$, and would like to use the likelihood ratio test to determine which candidate is the better choice. The first model candidate is a Markov chain with N states. In such a model, the signal makes a transition to a new state or stays at the same state at each time instance. The next state of signal depends only on the current state, and is selected according to a constant state transition probabilities matrix $A = [a_{ij}]$. This model is parameterized with $\boldsymbol{\theta} = (a_{11}, \cdots, a_{N-1,N})$. It is easily shown that state durations have a Geometrical probability mass function, where the probability of staying exactly d time units in state i is given by $\varphi_i(d) = a_{ii}^{d-1}(1 - a_{ii})$

The second model candidate is a Semi-Markov chain with the same number of states (i.e., N states). In this model, once the signal enters a new state, a state duration d is selected, and the signal stays exactly for d time units in the same state. After d time units, the signal will make a transition to a new state. The state duration d for state i is selected according to the probability mass function $\varphi_i(d)$, where $\varphi_i(d)$ is a discretized Gamma probability density function. That is,

$$\varphi_i(d) = \int_{d-1}^{d} \frac{\eta_i^{\nu_i}}{\Gamma(\nu_i)} x^{\nu_i - 1} e^{-\eta_i x} dx \tag{C.4}$$

ν_i and η_i are parameters of the Gamma pdf. This signal model is parameterized with $\boldsymbol{\theta} = (a_{12}, a_{13}, \cdots, a_{N-1,N}, \eta_1, \cdots, \eta_N, \nu_1, \cdots, \nu_N)$, where $A = [a_{ij}]$ is the state transition probabilities. Note that a_{ii}'s are zero, and $\sum_{j=1}^{N} a_{ij} = 1$ for all $1 \leq i \leq N$.

We use the likelihood ratio test to determine if the Semi-Markov chain is a better model choice for the empirical trace $\mathbf{S}_T$. More precisely, we test the null hypothesis '$H0$: $\mathbf{S}_T$ is generated by a Markov chain' against '$H1$: $\mathbf{S}_T$ is generated by a Semi-Markov chain'. According to equation C.3, it is necessary to parameterize the signal models such that $H0$ represent a special case of $H1$. This means that we should model a Markov chain as a special case of a Semi-Markov chain with Gamma state duration densities. This is easily done by defining

$$\begin{aligned} &\text{H0: } \boldsymbol{\theta} \in \omega, && \omega = \{\boldsymbol{\theta} | \nu_1 = \nu_2 = \cdots = \nu_N = 1, \sum_{j=1}^{N} a_{ij} = 1, 1 \leq i \leq N\} \\ &\text{H1: } \boldsymbol{\theta} \in \Omega, && \Omega = \{\boldsymbol{\theta} | \sum_{j=1}^{N} a_{ij} = 1, 1 \leq i \leq N\} \end{aligned} \tag{C.5}$$

Note that ω is a subspace of Ω. We should just show that conditions in $H0$ represent a Markov chain. That is, letting $\nu_1 = \nu_2 = \cdots = \nu_N = 1$ in the Semi-Markov chain model will result in a Markov chain. This is easily done by letting $\nu_i = 1$ in equation C.6.

$$\begin{aligned} \varphi_i(d) &= \int_{d-1}^{d} \frac{\eta_i^{1}}{\Gamma(1)} x^{1-1} e^{-\eta_i x} dx \\ &= e^{-\eta(d-1)} - e^{-\eta d} \end{aligned} \tag{C.6}$$

Selecting $a_{ii} = e^{-\eta}$ results in $\phi_i(d) = a_{ii}^{d-1}(1 - a_{ii})$, which is identical to Geometrical state duration densities of Markov chains.

Hence, one can find $-2\ln(\Lambda)$ and compare it to the upper $100 \times (1 - \alpha)$ percentile point of a χ^2 distribution with N degree of freedom to decide to reject the null hypothesis $H0$ or not.

Note that the hypothesis testing approach presented here is applicable to testing the validity of a HMM model against a HSMM model for a given empirical trace $\mathcal{Y}_T$ with very minor changes.

Bibliography

[1] J. Liebeherr and D. E. Wrege, "Traffic characterization algorithms for VBR video in multimedia networks," *ACM/Springer Multimedia Systems Journal*, vol. 6, no. 4, pp. 271–283, 1998.

[2] ATVEF forum, "ATVEF specicifcations for enhanced TV," avaialbale from http://www.atvef.com.

[3] Y. Wang, J. Osterman, and Y.Q. Zhang, *Digital Video Processing and Communications*, chapter 11, Scalable Video Coding, Prentice Hall, 2002.

[4] W. Dapeng, Y.T. Hou, and Y.Q. Zhang, "Scalable video coding and transport over broadband wireless networks," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 6–20, January 2001.

[5] H. Yanagihara, M. Sugano, A. Yoneyama, and Y. Nakajima, "Scalable video decoder and its application to multi-channel multicast system," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 866–871, 2000.

[6] D.P. Heyman, A. Tabatabai, and T.V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 2, no. 1, pp. 49–59, March 1992.

[7] O. Rose, "Simple and efficient models for variable bit rate MPEG video traffic," *Performance Evaluation*, vol. 30, pp. 69–85, 1997.

[8] M.Devetsikiotis M.R.Ismail, I.Lambadaris and A.R.Kaye, "Simulation and modeling of variable bit rate MPEG video transmission over ATM networks," *International Journal*

of Communication Systems, vol. 9, 1996, Available from http://www.sce.carleton.ca/bbnlab/bnlpapers.shtml.

[9] T.V. Lakshman D.P. Heyman, "Source models for VBR broadcast-video traffic," *Proc. IEEE Infocom*, pp. 664–671, June 1994.

[10] H. Hughes M. Krunz, "A traffic model for MPEG-coded VBR streams," *Proc. ACM Sigmetrics*, pp. 47–55, May 1995.

[11] A. Mukherjee A. Adas, "On resource management and qos guarantee for long range dependent traffic," *Proc. IEEE Infocom*, pp. 779–787, April 1995.

[12] W. Willinger M. W. Garret, "Analysis, modelling and generation of self-similar VBR video traffic," *Proc. ACM Sigcomm '94*, pp. 269–280, August 1994.

[13] J. F. Kurose, "On computing per-session performance bounds in high-speed multi hop computer networks," *Proc. ACM Sigmetrics and Performance '92*, pp. 128–139, June 1992.

[14] H. Zhang and E. W. Knightly, "Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models," *Proc. of ACM Sigmetrics*, pp. 211–220, May 1994.

[15] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Trans. on Information Theory*, vol. 37, no. 1, pp. 114–131, January 1991.

[16] R. L. Cruz, "A calculus for network delay, part II: Network analysis," *IEEE Trans. on Information Theory*, vol. 37, no. 1, pp. 132–141, January 1991.

[17] D. E. Wrege et. al., "Deterministic delay bounds for VBR video in packet switching networks: Fundamental limits and practical tradeoffs," *IEEE/ACM Trans. on Networking*, vol. 4, no. 3, pp. 352–362, June 1996.

[18] Dallas E. Wrege, *Multimedia Networks with Deterministic Quality-of-Service Guarantee*, Ph.d. thesis, University of Virginia, August 1996.

[19] R. L. Cruz and C. M. Okino, "Service guarantees for window flow control," *34th Allerton Conf. Communication, Control, and Computing*, October 1996.

[20] J.Y. Le Boudec, "Application of netwrok calculus to guaranteed services network," *IEEE Tran. on Information Theory*, vol. 44, no. 3, pp. 1087–1096, 1998.

[21] C. S. Chang, "On deterministic traffic regulation and service guarantees: A systematic approach by filtering," *IEEE Trans. On Information Theory*, vol. 44, no. 3, pp. 1097 –1110, 1998.

[22] ATM Forum, "ATM forum traffic management specification version 4.0," *Contribution 95-0013R11*, March 1996.

[23] ATM Forum, "ATM user-network user interface specification version 3.0," *Prentice-Hall*, 1993.

[24] E. W. Knightly, "Traffic models and admission control integrated services networks," *Ph.D. Thesis, University of Californiam, Berkeley*, May 1996.

[25] E. W. Knightly and H. Zhang, "Traffic characterization and switch utilization using a deterministic bounding interval dependent traffic model," *INFOCOM*, vol. 3, pp. 1137–1145, 1995.

[26] K. Moth L. Dittman, S. B. Jacobsen, "Flow enforcement algorithms for ATM netwroks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 3, pp. 343–350, April 1991.

[27] E. P. Ratgheb, "Modelling and performance comparison of policing mechanisms for ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 4, pp. 325–334, April 1991.

[28] M. Pancha, P.; El Zarki, "Leaky bucket access control for VBR MPEG video," *INFOCOM*, vol. 2, pp. 796–803, 1995.

[29] F. Guillemin, C. Rosenberg, and J. Mignault, "On characterizing an ATM source via the sustainable cell rate traffic descriptor," *INFOCOM*, vol. 3, pp. 1129 –1136, 1995.

[30] D. E. Wrege and J. Liebherr, "Video traffic characterization for multimedia networks with a deterministic service," *INFOCOM*, pp. 537–544, 1996.

[31] T. H. Cormen, *Introduction to Algorithms*, McGraw Hill, 2001.

[32] P. E. Gill et. al., *Practical Optimization*, Academic Press, 1981.

[33] R. Fletcher, *Practical Methods of Optimization*, John Wiley, 1980.

[34] C. Rosenberg et. al., "New approach for traffic characterisation in ATM networks," *IEEE Proc. on Comm.*, vol. 142, pp. 87 –90, 1995.

[35] P. Thiran J.Y. Le Boudec, *Network calculus : a theory of deterministic queuing systems for the Internet*, Springer, 2001.

[36] J.Y. Le Boudec and P. Thiran, "A short tutorial on network calculus. I. fundamental bounds in communication networks," *The 2000 IEEE Int. Symposium on Circuits and Systems*, vol. 4, pp. 93–96, 2000.

[37] R. J. Gibben et al, "Effective bandwidth for multi-type UAS channel," *Queueing systems*, vol. 9, pp. 17–28, 1991.

[38] F. P. Kelly, "Effective bandwidth at multi-class queues," *Queueing systems*, vol. 9, pp. 5–16, 1991.

[39] R. Guerin et al, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Selected Areas in Communication*, vol. 9, pp. 968–981, 1991.

[40] Alan Weiss, "An introduction to large deviations for communication networks," *IEEE Journal on selected areas in Communications*, vol. 13, no. 6, pp. 938–952, 1995.

[41] C.S. Chang, *Performance Quarantees in Communication Networks*, Springer-Verlag, 2000.

[42] N.G. Duffield, "A large deviation analysis of errors in measurement based admission control to buffered and bufferless resources," *Queueing Systems*, vol. 34, no. 1, pp. 131–168, January 2000.

[43] Jean C. Walrand Gustavo de Veciana, George Kesidis, "Resource management in wide-area ATM networks using effective bandwiths," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1081–1090, 1995.

[44] N.G. Duffield, J.T. Lewis, N. O'Connell, R. Russell, and F. Toomey, "Entropy of ATM traffic streams: a tool for estimating qos parameters," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 981–990, August 1995.

[45] P. Rabinovitch, "Statistical estimation of effective bandwidth," M.S. thesis, Carleton University, April 2000.

[46] S. Tartarelli, M. Falkner, M. Devetsikiotis, I. Lambadaris, and S. Giordano, "Empirical effective bandwidths," *Global Telecommunications Conference, 2000. GLOBECOM '00*, vol. 1, no. 27, pp. 672–678, December 2000.

[47] H. R. Kunsch, "The jacknife and the bootstrap for general stationary observations," *The annals of Statistics*, vol. 17, pp. 1217–1241, 1989.

[48] Raoul LePage and Lynne Billard, Eds., *Exploring the Limits of Bootstrap*, John Wiley and sons, April 1992.

[49] G.A. Young, "Bootstarb: More than a stab in the dark?," *Statistical Science*, vol. 9, pp. 382–415, 1994.

[50] N.G. Duffield; J.T. Lewis; N. O'Connell, "The entropy of an arrival process: A tool for estimating QOS parameters of ATM traffic," *Proceedings of 11th IEE UK Teletraffic Symposium*, March 1994.

[51] N.G. Duffield, "A large deviation analysis of errors in measurement based admission control to buffered and bufferless resources," *AT&T research lab report*, 1998.

[52] C. Courcoubetis; G. Kesidis; A. Ridder; J. Walrand; R. Weber, "Admission control and routing in ATM networks using inferences from measured buffer occupancy," *IEEE Trans. on Communication*, vol. 43, no. 2/3/4, pp. 1778–1784, April 1995.

[53] A . Adas, "Traffic models in broadband networks," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 82–89, July 1997.

[54] D. L. Jagerman and B. Melamed, "The transition and autocorrelation structure of TES processes, part I: General theory," *Stochastic Models*, vol. 8, no. 2, pp. 193–219, 1992.

[55] D. L. Jagerman and B. Melamed, "The transition and autocorrelation structure of TES processes, part II: Special cases," *Stochastic Models*, vol. 8, no. 3, pp. 499–527, 1992.

[56] B. Melamed, D. Raychaudhuri, B. Sengupta, and J. Zdepski, "TES-based traffic modeling for performance evaluation of integrated networks," *INFOCOM '92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 75–84, 1992.

[57] B. Melamed and D. E. Pendarakis, "Modeling full-length VBR video using markov-renewal-modulated tes models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 600–611, June 1998.

[58] Chang Bum Lee, Kyeong Bong Ha, and Rae-Hong Park;, "Computation of effective bandwidth of aggregated vbr mpeg video traffic in atm networks using the modified equivalent capacity," *IEEE International Conference on Communications, ICC 96, Conference Record, Converging Technologies for Tomorrow's Applications*, vol. 2, no. 2, pp. 627–631, June 1996.

[59] N. Ansari, Y.Q. Hai Liu Shi, and Hong Zhao, "On modeling mpeg video traffics," *IEEE Transactions on Broadcasting*, vol. 48, no. 4, pp. 337 –347, December 2002.

[60] J.C. Cano and P. Manzoni, "On the use and calculation of the hurst parameter with mpeg videos data traffic," *Proceedings of the 26th Euromicro Conference*, vol. 1, pp. 448–455, September 2000.

[61] R. Narasimha and R.M. Rao, "Discrete-time self-similar systems and stable distributions: applications to vbr video modeling," *IEEE Signal Processing Letters*, vol. 10, no. 3, pp. 65–68, March 2003.

[62] B.N. Bashforth and C.L. Williamson, "Statistical multiplexing of self-similar video streams: simulation study and performance results," *Proceedings of the Sixth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 119–126, July 1998.

[63] O. Lazaro, D. Girma, and J. Dunlop, "Statistical analysis and evaluation of modelling techniques for self-similar video source traffic," *The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, vol. 2, pp. 1540 –1544, September 2000.

[64] V.S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, vol. 32, no. 3, pp. 70–81, March 1994.

[65] Z. Liu, J. Huang, and Y. Wang, "Classification TV programs based on audio information using hidden markov model," *IEEE Second Workshop on Multimedia Signal Processing*, pp. 27–32, December 1998.

[66] J. Huang, Z. Liu, Y. Wang, Y. Chen, and E.K. Wong, "Integration of multimodal features for video scene classification based on HMM," *IEEE 3rd Workshop on Multimedia Signal Processing*, pp. 53–58, September 1999.

[67] D.L. Mclaren and D.T. Nguyen, "An HMM-based simulation model for the production of coded image data," *International Conference on Image Processing and its Applications*, pp. 93–96, April 1992.

[68] J. Huang, Z. Liu, and Y. Wang, "Joint video scene segmentation and classification based on hidden markov model," *IEEE International Conference on Multimedia and Expo, ICME 2000*, vol. 3.

[69] L. Chaisorn, C.T. Seng, and C.H. Lee, "The segmentation of news video into story units," *IEEE International Conference on Multimedia and Expo, ICME 2002*, vol. 1.

[70] P. Morguet and M. Lang, "An integral stochastic approach to image sequence segmentation and classification," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1998*, vol. 5.

[71] R.J. Elliott, L. Aggoun, and J.B. Moore, *Hidden Markov Models: Estimation and Control*, Springer-Verlag,New York, 1995.

[72] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 257–286, February 1989.

[73] A. A. Lazar, G. Pacifici, and D. E. Pendarakis, "Modeling of video sources for real time scheduling," *Multimedia Systems*, vol. 2, no. 6, pp. 253–266, 1994.

[74] M. J. Russel and R. K. Moore, "Explicit modelling of state occupancy in hidden markov models for automatic speech recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5–8, March 1985.

[75] B. Sin and J. H. Kim, "Nonstationary hidden markov model," *Signal Processing*, vol. 46, pp. 31–46, 1995.

[76] L. H. Jamieson and C. D. Mitchell, "Modelling duration in a hidden markov model with the exponential family," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 331 –334, 1993.

[77] S. E. Levinson, "Continuously variable duration hidden markov models for automatic speech recognition," *Computer Speech and Language*, vol. 1, pp. 29–45, 1986.

[78] V. Krishnamurthy, J. B. Moore, and S. H. Chung, "Hidden fractal model signal processing," *Signal Processing*, vol. 24, no. 2, pp. 177–192, 1991.

[79] S. V. Vaseghi, "State duration modelling in hidden markov models," *Signal Processing*, vol. 41, no. 1, pp. 31–41, 1995.

[80] E. Weinstein, M. Feder, and A. V. Oppenheim, "Sequential algorithms for parameter estimation based on the kullback-leibler information measure," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 9, pp. 1652–1654, September 1990.

[81] U. Holst and G. Lindgren, "Recursive estimation in mixture models with markov regime," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1683–1690, 1991.

[82] I. B. Collings, V. Krishnamurthy, and J. B. Moore, "Online identification of hidden markov models via recursive prediction error techniques," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3535–3539, 1994.

[83] I. B. Collings and T. Ryden, "A new maximum likelihood gradient algorithm for online hidden markov model identification," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2261–2264", May 1998.

[84] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, 1983.

[85] R. S. Burington, *Handbook of probability and statistics with tables*, McGraw-Hill, 1970.

[86] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[87] M. Abramowitz and I. Stegun, *Handbook Of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables*, 1964.

[88] G. E. Andrews, R. Askey, and R. Roy, *Special Functions*, Cambridge University Press, 1999.

[89] L. Ljung, *System Identification: Theory For The User*, Prentice Hall PTR, 1999.

[90] B. T. Polyak, "New method of stochastic approximation type," *Automat. Remote Control*, vol. 51, pp. 937–946, 1990.

[91] D. Ruppert, "Stochastic approximation," *in Handbook in Sequential Analysis, B. K. Ghosh and P. K. Sen, Eds.*, pp. 503–529, New York: Marcel Dekker 1991.

[92] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge, 1987.

[93] Advanced Television Systems Committee, "ATSC digital television standard A-53, ATSC digital television standard," 2001.

[94] Digital Video Broadcasting Project, "DVB digital television standard," .

[95] R. S. Chernock, *Data broadcasting: Understanding the ATSC Data Broadcast Standard*, McGraw-Hill, New York, 2001.

[96] H. Benoit, *Digital television : MPEG-1, MPEG-2, and principles of the DVB system*, J. Wiley and Sons, 1997.

[97] Advanced Television Systems Committee, "ATSC digital television standard A-65, program and system information protocol for terrestrial broadcast and cable," 2000.

[98] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*, New York: John Wiley, 2000.

[99] V. Barnett, "Probability plotting methods and order statistics," *Applied Statistics*, , no. 24, pp. 95–108, 1975.

[100] V. Barnett, "Quantile-quantile plot," *NIST/SEMATECH e-Handbook of Statistical Methods.*