QoS evaluation of Bandwidth Schedulers in IPTV Networks Offered SRD Fluid Video Traffic

Mohammad Ahasan Habib

2009 Master Thesis in Computer Engineering Reg. No: E3682D
| Program: Master of Science in Computer Engineering | Reg. number: E3682D | Extent: 30 ECTS |
| Name of student: Mohammad Ahasan Habib | Year-Month-Day 2009-04-01 |
| Supervisor: Prof. Mark Dougherty | Examiner: Prof. Mark Dougherty |
| Company/Department: Department of Culture, Media & Computer Science, Dalarna University | Supervisor at the Company/Department Dr. Ernst Nordström |
| Title: QoS evaluation of bandwidth schedulers in IPTV networks offered SRD fluid video traffic. |
| Keywords: Internet Protocol Television (IPTV), Quality of Service, Short Range Dependence, Packets Scheduler, Markov Fluid Flow Model. |
Abstract

Internet protocol TV (IPTV) is predicted to be the key technology winner in the future. Efforts to accelerate the deployment of IPTV centralized model which is combined of VHO, encoders, controller, access network and Home network. Regardless of whether the network is delivering live TV, VOD, or Time-shift TV, all content and network traffic resulting from subscriber requests must traverse the entire network from the super-head-end all the way to each subscriber's Set-Top Box (STB). IPTV services require very stringent QoS guarantees When IPTV traffic shares the network resources with other traffic like data and voice, how to ensure their QoS and efficiently utilize the network resources is a key and challenging issue. For QoS measured in the network-centric terms of delay jitter, packet losses and bounds on delay.

The main focus of this thesis is on the optimized bandwidth allocation and smooth data transmission. The proposed traffic model for smooth delivering video service IPTV network with its QoS performance evaluation. According to Maglaris et al [5] first, analyze the coding bit rate of a single video source. Various statistical quantities are derived from bit rate data collected with a conditional replenishment inter frame coding scheme. Two correlated Markov process models (one in discrete time and one in continuous time) are shown to fit the experimental data and are used to model the input rates of several independent sources into a statistical multiplexer. Preventive control mechanism which is to be including CAC, traffic policing used for traffic control.

QoS has been evaluated of common bandwidth scheduler( FIFO) by use fluid models with Markovian queuing method and analysis the result by using simulator and analytically, Which is measured the performance of the packet loss, overflow and mean waiting time among the network users.
Acknowledgements

First of all I would like to thank to Almighty Allah, who has given me the strength to successfully reach to the end of this program.

I would like to express my deepest gratefulness to my supervisor Prof. Mark Dougherty, for his valuable time and taking his time to explain a lot of things to me in order to complete this thesis. He is one of the best person other than a teacher.

I am highly indebted to Dr. Ernst Nordström, Ernst Consulting & Education who not only evinced keen interest in the project but also encouraged and guided me in the execution of the project. He also monitored the progress of my work at different stages. I learn a lot on the aspect of doing thesis from him.

Also, special thanks to Dr. Pascal Rebreyend and Dr. Hasan Fleyeh, Dr. Siril Yella, Jerker Westin, you all are very good teachers and I am benefited a lot from your tuitions. Thanks for your sincere help during the last two years.

Finally, thanks to everyone that gave us some words of encouragement or advice.
Abbreviations and Acronyms

3GPP – Third Generation Partnership Project
AAA – Authentication, Authorization and Accounting
ACF – Autocorrelation Function
ADSL – Asymmetric Digital Subscriber Line
AN – Access network
AR – Autoregressive
ASCII – American Standard Code for Information Interchange
ATD – Asynchronous Time Division
AVC – Advanced Video Coding
BT – Burst Tolerance
CAC – Call Admission Control
CBWFQ – Class-Based Weighted Fair Queueing
CD – Compact Disk
CDV – Cell Delay Variation
CLR – Cell Loss Ratio
CN – Core Network
CM – Commercial
CSCF – Call Session Control Functions
CTD – Cell Transfer Delay
CTMP – Continuous Time Markov Process
DCT – Discrete Cosine Transform
DES – Discrete Event Simulation
DRM – Digital Rights Management
DSL – Digital Subscriber Line
DSLAM – Digital Subscriber Line Access Multiplexer
DTDM – Discrete Time Division Multiplexing
DTMP – Discrete Time Markov Process
DVD – Digital Video Disc
ETSI – European Telecommunications Standards Institute
FBM – Fractional Brownian motion
FCFS – First Come First Serve
FIFO – First in First Out
GPS – Generalized Processor Sharing
GCRA – Generic Cell Rate Algorithm
GoP – Group of Pictures
GoS – Grade of Service
HDTV – High Definition Television
HN – Home Network
IMS – IP Multimedia Subsystem
IPTV – Internet Protocol Television
ISO – International Standards Organization
ITU-T – The International Telecommunication Union Telecommunication
LLQ – Low Latency Queueing
LRD – Long-Range Dependence
MN – Metro Network
MPEG – Moving Pictures Experts Group
NASS – Network Attachment Subsystem
NACF – Network Attachment Control Functions
NGN – Next Generation Network
NP – Network performance
NVOD – Near Video on Demand
PCR – Peak Cell Rate
PD-FE – Policy Decision Functional Entity
PQ – Priority Queuing
PSTN – Public Service Telephone Network
QoS – Quality of Service
QoE – Quality of Experience
RACF – Resource and admission control functions
RACS – Resource and admission control subsystem
RG – Residential Gateway
SCR – Sustained Cell Rate
SCF – Service Control Functions
SHE – Super Head End
SRD – Short Range Dependence
STB – Set Top Box
TDM – Time Division Multiplexing
TEFID – Traffic Engineering in Future Internet Domains
TE – Traffic Engineering
TF – Transport Functions
TRC-FE – Transport resource Control Functional Entity
VBR – Variable Bit-Rate
VDSL – Very high-speed Digital Subscriber Line
VHO – Video Hub Office
VoD – Video on Demand
VSO – Video Serving Office
WFQ – Weighted Fair Queuing
WRR – Weighted Round-Robin
# Table of Contents

1. Introduction: .................................................................................................................... 1  
   1.1 Thesis environment: .................................................................................................. 1  
   1.2 Background: .............................................................................................................. 1  
   1.3 Objectives: ................................................................................................................ 3  
   1.4 Limitations: ............................................................................................................... 4  
   1.5 Disposition and Responsibility distribution: ............................................................. 4  
2. Problem formulation: ...................................................................................................... 6  
   2.1 Problem statement:.................................................................................................... 6  
   2.2 Questions for Investigation: ...................................................................................... 8  
3. IPTV System: .................................................................................................................. 9  
   3.1 IPTV Services: .......................................................................................................... 9  
       3.1.1 IPTV Video on Demand (VoD): ........................................................................ 9  
       3.1.2 Time-Shifted TV: ............................................................................................. 10  
       3.1.3 Broadcast TV Service: ..................................................................................... 11  
       3.1.3.1 Effect of Broadcast TV (IPTV) services: .................................................. 12  
       3.1.5 Near Video on Demand (NVOD): ................................................................... 12  
   3.2 IPTV Network Architecture: ................................................................................... 13  
   3.3 The IPTV network: ................................................................................................. 14  
       3.3.1 Core Network (CN): ........................................................................................ 14  
       3.3.2 Metro Network (MN):...................................................................................... 15  
       3.3.3 Access Network (AN): ..................................................................................... 15  
       3.3.4 Home Network (HN): ...................................................................................... 15  
   3.4 NGN-based IPTV solution:..................................................................................... 16  
       3.4.1 Generic NGN model: ....................................................................................... 16  
       3.4.2 Generic vertical IPTV model: .......................................................................... 17  
   3.5 IPTV service delivery within NGN architecture: ................................................... 18  
       3.5.1 NGN architecture framework: ......................................................................... 19  
       3.5.2 Resource and Admission Control Functions (RACF): .................................... 21  
4. Traffic Control: ............................................................................................................. 23  
   4.1 Performance measures: ........................................................................................... 23  
       4.1.1 Quality of Experience (QoE): .......................................................................... 24  
       4.1.2 Quality of Service (QoS): ................................................................................ 25  
       4.1.3 Grade of service (GoS): ................................................................................... 26  
   4.2 Preventive Control Schemes: .................................................................................. 27  
       4.2.1 Traffic Policing: ............................................................................................... 27  
       4.2.2 Traffic Shaping: ............................................................................................... 28  
       4.2.3 Call Admission Control: .................................................................................. 29  
   4.3 Reactive Control Schemes: ..................................................................................... 30  
       4.3.1 Flow Control: ................................................................................................... 31  
       4.3.2 Congestion Control: ......................................................................................... 31  
5. VBR Video Traffic: ...................................................................................................... 33  
   5.1 Overview of MPEG Encoding: ............................................................................... 33
5.1.1 MPEG-2: .............................................................. 35
5.1.2 MPEG-4: .............................................................. 36
5.2 Layer video traffic modelling: ................................................................. 37
  5.2.1 Scene layer: .................................................................................. 37
  5.2.2 GOP Layer: ............................................................................... 37
  5.2.3 Frame Layer: ............................................................................... 38
  5.2.4 Packet Layer: ............................................................................... 38
5.3 Traffic Characteristics: ........................................................................... 39
  5.3.1 Short-Range Dependence (SRD): ................................................. 39
  5.3.2 Long-Range Dependence (LRD): .................................................... 39
  5.3.3 Self-Similar Process: ................................................................. 42
    5.3.3.1 Continuous-time self-similar process: ...................................... 43
5.4 Video Traffic Modeling: ........................................................................... 44
  5.4.1 Markov based Model: ................................................................. 44
    5.4.1.1 Discrete time Markov chains: ............................................... 44
    5.4.1.2 Continuous time Markov chains: .......................................... 45
  5.4.4 Definition of autoregressive Process: ........................................... 46
  5.4.5 Markov Modulated Fluid Models: ............................................... 48
6. Packet scheduling in IP routers: ................................................................. 50
  6.1 First in First out Queuing (FIFO): ....................................................... 52
  6.2 Priority Queuing (PQ): ....................................................................... 53
  6.3 Weighted Fair Queuing (WFQ): ............................................................ 53
  6.4 Weighted Round Robin (WRR): ............................................................ 54
  6.5 Generalized Processor Sharing (GPS) ............................................... 55
  6.6 Low Latency Queuing (LLQ) ............................................................... 56
7. Video source models: .................................................................................. 57
  7.1 The modeling process: ....................................................................... 59
  7.2 Source Model Inferences from Experimental Results: ....................... 62
  7.3 Source Model A: Continuous-State Autoregressive Markov Model: ......... 66
  7.4 Queueing Simulation Using Model A: .............................................. 68
  7.5 Source Model B: ............................................................................... 69
    7.5.1 Discrete-State, Continuous-Time Markov Process: ....................... 69
  8.1 Multiplexed Queue System: ............................................................... 76
    8.1.1 Fluid flow model: .................................................................... 77
  8.2 Solution: ......................................................................................... 79
  8.3 Performance analysis with the Fluid Flow approach: ......................... 80
    8.3.1 Fluid Overflow Probability: .................................................... 80
    8.3.2 Fluid Loss Probability: ........................................................... 81
    8.3.3 Fluid Mean Delay Probability: .................................................. 81
  9.1 Discrete-Event Simulation: ................................................................. 83
  9.2 Fluid flow simulation model for FIFO Multiplexer: .............................. 84
  9.3 Simulation flow chart: ....................................................................... 90
10. Numerical Analysis: ................................................................................. 91
10.1 Experiment A: ................................................................................................................. 92
  10.1.1 QoS evaluation parameter analysis: ............................................................... 92
  10.1.2 Comparison of results with analytical simulator: ......................................... 94
10.2 Experiment B: ......................................................................................................... 95
  10.2.1 Simulation-1: Investigation of variable buffer size for GOP: ......................... 96
  10.2.2 Simulation-2: Investigation of variable link capacity for GOP: ....................... 98
  10.2.3 Simulation-3: Investigation of variable number of sources for GOP: .......... 100
  10.2.4 Simulation-1: Investigation of variable buffer size for Frame: ..................... 102
11. Conclusion and Future work: .................................................................................... 105
  11.1 Conclusion: ......................................................................................................... 105
  11.2 Further Work Issues: ........................................................................................ 106
Appendix ......................................................................................................................... 111
## List of Figures

3.1 Time–Shifted TV viewing ................................................................. 10
3.2 A typical IPTV network architecture .............................................. 13
3.3 An illustrative network deployment of IPTV network ....................... 14
3.4 a) The generic NGN model; b) the generic vertical IPTV model ............. 17
3.5 IPTV service component include in NGN architecture ...................... 20
3.6 Resource and admission control functional architecture .................... 21
4.1 The general processing sequence of traffic control ......................... 23
4.2 QoE dimensions ........................................................................... 24
4.3 A queuing model for a leaky bucket method ..................................... 28
5.1 MPEG frame pattern ..................................................................... 34
6.1 General architecture of a network switch ........................................ 50
6.2 The model of a scheduling algorithm .............................................. 51
6.3 First in first out scheduling ............................................................ 52
7.1 Number of frame in terms of bit rate for full movie (Star Wars IV) .... 60
7.2 Number of GOP in terms of bit rate for source movie (Star Wars IV)  61
7.3 Coding bit rate of five different movies ......................................... 63
7.4 ACF Vs Lags for GOP. ................................................................. 65
7.5 ACF Vs Lags for Frame ............................................................... 67
7.6 Fluid-flow model for a finite-buffer switching node under ON/OFF traffic sources. 70
7.7 Number of GOP in terms of bit rate for Aggregate movie source (Star Wars IV) 70
7.8 Fitting curve for GOP in aggregate movie ...................................... 73
7.9 Buffer size Vs Waiting time in different sources ............................... 73
7.10 Buffer size Vs Loss probability in different sources .......................... 74
7.11 Buffer size Vs Overflow in different sources .................................... 75
8.1 The multiplexed Overflow system .................................................. 77
9.1 Sequence flow of the Simulation ..................................................... 86
10.1 Buffer size Vs Loss Probability with statistical smoothing and without smoothing 92
10.2 Buffer size Vs Overflow with statistical smoothing and without smoothing .... 93
10.3 Buffer size Vs Waiting time with statistical smoothing and without smoothing ... 94
10.4 Buffer size Vs Loss probability in two different simulators ................ 95
10.5 Buffer size Vs Overflow in different simulators ............................... 95
10.6 Buffer size Vs Loss probability for different trace files in GOP .......... 96
10.7 Buffer size Vs Overflow for different trace files in GOP ..................... 97
10.8 Buffer size Vs Waiting time for different trace files in GOP ............... 97
10.9 Link capacity Vs Loss probability for different trace files in GOP ...... 98
10.10 Link capacity Vs Overflow for different trace files in GOP ............... 99
10.11 Link capacity Vs Waiting time for different trace files in GOP .................. 99
10.12 Number of source Vs Loss probability for different trace file in GOP ........ 100
10.13 Number of source Vs Overflow for different trace files in GOP .............. 101
10.14 Number of source Vs Waiting time for different trace files in GOP ....... 101
10.15 Buffer Size Vs Loss probability for different trace files in Frame .......... 102
10.16 Buffer Size Vs Overflow for different trace files in Frame .................. 103
10.17 Buffer Size Vs Waiting time for different trace files in Frame ............. 103
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Characteristics Coefficient values for different video traces</td>
<td>67</td>
</tr>
<tr>
<td>7.2</td>
<td>Model parameter values for different trace files</td>
<td>68</td>
</tr>
<tr>
<td>7.3</td>
<td>Peak rate values and Markov source transition parameter values for different video traces</td>
<td>72</td>
</tr>
<tr>
<td>10.1</td>
<td>QoS evaluation parameter values for before and after smoothing</td>
<td>93</td>
</tr>
<tr>
<td>10.2</td>
<td>QoS evaluation parameter values for Simulator and Analytical program</td>
<td>95</td>
</tr>
<tr>
<td>10.3</td>
<td>QoS evaluation parameter values for simulation-1 in GOP</td>
<td>96</td>
</tr>
<tr>
<td>10.4</td>
<td>QoS evaluation parameter values for simulation-2 in GOP</td>
<td>98</td>
</tr>
<tr>
<td>10.5</td>
<td>QoS evaluation parameter values for simulation-3 in GOP</td>
<td>100</td>
</tr>
<tr>
<td>10.6</td>
<td>QoS evaluation parameter values for simulation-1 in Frame</td>
<td>102</td>
</tr>
</tbody>
</table>
CHAPTER -1

1. Introduction:

Internet Protocol Television (IPTV) is a system where a digital television service is delivered by using Internet Protocol over a network infrastructure, which may include delivery by a broadband connection for residential and business users at a lower cost. These IPTV services include commercial grade multicasting TV, video on demand (VoD), triple play, voice over IP (VoIP), and Web/email access, well beyond traditional cable television services. Moreover IPTV is a convergence of communication, computing and content as well as an integration of broadcasting and telecommunication. IPTV delivering its services over IP based networks managed to provide the required level of Quality of Service (QoS) and Quality of Experience (QoE), security, interactivity and reliability.

1.1 Thesis environment:

This is a Master thesis work done in partial fulfilment of the requirements for the award of International Master of Science in Computer Engineering degree, Högskolan Dalarna (Dalarna University), Sweden. This thesis is part of a research project entitled Traffic Engineering in Future Internet Domains (TEFID) carried out at the Department of Economics and Social Sciences at Dalarna University, Sweden.

1.2 Background:

IPTV is an integration of voice, video, and data services using high bandwidth and high speed Internet access. IPTV using broadcast video over private IP networks that are isolated from Internet. IPTV services rely on transmission of real-time video and stored video. Unlike cable TV delivery, IPTV is very different in that it only delivers the single channel that is requested by the consumer's individual TV all the way from the IPTV service provider's head-end equipment. Therefore, with IPTV the infrastructure needed to support huge amounts of bandwidth being delivered all the time is not needed. The infrastructure for IPTV service providers only needs to
support the specific request for channel bandwidth that is requested from the consumer at any given time. For IP unicast, IPTV services are video on demand (VOD) and time-shifted TV and in IP multicast, IPTV services are broadcast TV and near video on demand.

IPTV services require very stringent QoS guarantees. Quality of Service ensures that IPTV sessions are guaranteed the correct network session parameters to provide a quality experience for the subscriber. QoS measured in the network-centric terms of delay jitter, packet losses and bounds on delay. A small amount of delay does not directly affect the quality of experience (QoE) of IPTV. However, a delay longer than 1 second may result a less than satisfying end-user experience. Packet losses will likely cause visible artefacts due to the high compression rates of MPEG encoded TV signals. In order to have less than one visible artefact per movie on the TV screen, the packet loss rate must be lower than $10^{-6}$, 200 ms delay and 50 ms jitter. Low delay is essential for satisfactory trick mode performance like pause, resume and fast forward for VoD and fast channel change time for BTV.

Traffic Engineering (TE) is the process of steering traffic across to the backbone to facilitate efficient use of available bandwidth between a pair of routers. Internet Traffic Engineering (TE) is defined in RFC3272 and involves both capacity management and traffic management. Capacity management includes capacity planning, routing control, and resource management. Traffic management includes nodal traffic control functions such as traffic conditioning, queue management, scheduling, and other functions that regulate traffic flow through the network or that arbitrate access to network resources. From a QoS control perspective, traffic management in the IPTV network can be particularly challenging. This is because traffic management solutions have to be implemented at different levels of control granularity. Those levels include [29]:

- The individual services active used by a given subscriber,
- The individual DSL link-load for the given subscriber,
- The aggregate subscribers supported on a given line card, and
The aggregate line cards supported on a given uplink card.

We know Traffic engineering (TE) in IPTV networks have both preventive (open-loop) and reactive (closed-loop) traffic control mechanisms. TE in NGN-IMS-based IPTV networks will rely on open-loop preventive traffic control mechanisms due to the stringent QoS requirements. Call admission control (CAC) and traffic policing forms the basis for preventive traffic control. Admission control decides to accept or reject upstream and downstream bandwidth requests, ensuring an accepted flow of bandwidth that satisfies the QoS requirement.

Triple Play technology is nowadays considered as an indisputable trend. The Differentiated Service (DiffServ) architecture is preferred over “Hard Quality of Service” (QoS) Integrated Service (IntServ) architecture. Moreover it applies perfectly to triple play, as it satisfies differing QoS requirements. For full exploitation of available bandwidth and providing adequate QoS to subscribed users by meeting the requirements of all three supported services (video, voice and data) must be addressed [6]. In this context each IP router in the core and metro networks implements some queueing and packet scheduling mechanism at the output link controller. Popular schedulers that are used in IP networks which are Priority Queueing (PQ), Class-Based Weighted Fair Queueing (CBWFQ), and Low Latency Queueing (LLQ) which combines PQ and CBWFQ. The evaluation of the schedulers is based on simulations designed upon real triple play networks, while trace files are used for the simulation of video flows.

1.3 Objectives:

The objective of this thesis work is to analyze source model for evaluate QoS of bandwidth scheduling in IPTV networks. Preventive traffic control mechanisms which is call admission control (CAC) and traffic policing are applied for this project. Call admission control (CAC) accepts or rejects arriving call requests with the objective of keeping the risk of congestion within tolerable limits. We only study reactive control from a theoretical perspective and do not make any simulations with reactive control mechanisms. It is describe the traffic of each VBR video source by a
short range dependent (SRD) model including Markov models and regression models. The main task in this project is to make a program in the C language which fits a superposition of Markov fluid ON/OFF sources which is SRD traffic model to a smoothed VBR video source and evaluates the QoS of common bandwidth schedulers given traffic from this model. This evaluation done by analytically and simulation, which evaluates the performance of this method. The analytical approaches for evaluation of the performance of a multiplexer loaded with SRD video sources. The performance of the scheduling system based on the probability of loss or overflow and the mean waiting time.

1.4 Limitations:

- We consider one traffic class which could be more in reality.
- In this thesis, FIFO scheduler scheme and other schedulers can be introduced and evaluated.
- Only Preventive traffic control mechanism is used.
- The calls are assumed to have an exponential holding time instead of general holding times.
- The simulation is configured for a smaller version (scale) of the realistic network due to huge computational times.

1.5 Disposition and Responsibility distribution:

This thesis report consists of 11 chapters.
This field is very new and very little amount of work is done on this particular area. However, the object and responsibilities specified is a bit too large for an individual. This is a joint work and the responsibilities are assigned to two M.Sc thesis students to accomplish. Detail is given below:
This chapter is basically about the basics of the thesis. The rest of this report is organized as follows.

Chapter 2: States the problem statement and the questions for investigation of the thesis. Chapter 3: Deals with IPTV system which include IPTV services, its network Architecture and NGN-based solution.

Chapter 4: Describe in brief about traffic control which is QoE, QoS, GoS, and Preventive traffic control and Reactive traffic control mechanism.

Chapter 5: Deals with VBR video traffic where MPEG encoding, Layered Video traffic model and Traffic characteristics are described.

Chapter 6: Theoretical overview of some popular scheduling methods in IP networks.

Chapter 7: Describe the Video source models based on Markov model.

Chapter 8: Evaluate the QoS by analysis of fluid flow model.

Chapter 9: Evaluate the QoS by Simulation.

Chapter 10: Numerical results and analysis.

Chapter 11: Conclusion and future of this work.

<table>
<thead>
<tr>
<th>Chapter#</th>
<th>Description</th>
<th>Assign mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>Joint</td>
</tr>
<tr>
<td>2</td>
<td>Problem Formulation</td>
<td>Joint</td>
</tr>
<tr>
<td>3</td>
<td>IPTV System</td>
<td>Chandra Shekhar Mondal</td>
</tr>
<tr>
<td>4</td>
<td>Traffic Control</td>
<td>Mohammad Ahasan Habib</td>
</tr>
<tr>
<td>5</td>
<td>VBR Video Traffic</td>
<td>Chandra Shekhar Mondal</td>
</tr>
<tr>
<td>6</td>
<td>Packet scheduling in IP routers</td>
<td>Mohammad Ahasan Habib</td>
</tr>
<tr>
<td>7</td>
<td>Video source models</td>
<td>Chandra Shekhar Mondal</td>
</tr>
<tr>
<td>8</td>
<td>Analytical framework for QoS evaluation</td>
<td>Chandra Shekhar Mondal</td>
</tr>
<tr>
<td>9</td>
<td>Simulation framework for QoS evaluation</td>
<td>Mohammad Ahasan Habib</td>
</tr>
<tr>
<td>10</td>
<td>Numerical Analysis</td>
<td>Mohammad Ahasan Habib</td>
</tr>
<tr>
<td>11</td>
<td>Conclusion and Future work</td>
<td>Joint</td>
</tr>
</tbody>
</table>
CHAPTER-2

2. Problem formulation:

2.1 Problem statement:

In the literature, intensive research has been conducted to analyze the performance of scheduling schemes. Statistical or asynchronous time division multiplexing of variable bit rate is analyzed to efficiently utilize a common communications channel while maintaining uniform picture quality at the receiver. One traffic class, modeled as Markov fluid sources, was multiplexed by FIFO scheduling. Queueing models of such schemes are tested and their probabilistic behavior is assessed to smooth the transmission over an asynchronous time division queueing channel. Finally, the Quality of Service of the whole network is evaluated on the basis of the loss of packets, data overflow and queueing delay.

In traditional IPTV network model, data transmission performance over downlink is severely limited by the discrete and asynchronous mode of burstiness. Call arrives to the sender in a discrete or sometimes in a continuous fashion. The queueing mechanism only handles the available packet in the arrival channel. As a result the communication channel is not optimized and takes a longer time for data transition. The component is sometimes called a bottleneck point. The term is metaphorically derived from the neck of a bottle, where the flow speed of the liquid is limited by its neck.

A good traffic model should not be overly complicated and should provide insights into the statistics of video traffic that have the greatest impact on queue behavior. Video frame size modeling has drawn great attention in recent decades. Among the existing models, Markov chain based models are desirable for queue analysis given the well-established fluid-flow analytical framework. Maglaris et. al [5] proposed a mini-source based Markov model to describe the variable source rates of different
Groups of Pictures (GoPs). It is effective in obtaining the queue distribution when the buffer size is sufficiently large.

Total contribution to this work is performed in three stages. First, Markovian model is used for IPTV traffic, considering both temporal and spatial correlation, presented by GoP correlation and frame correlation in the same GoP, respectively. The model contains a GoP-level Markov chain and a frame-level Markov chain, to capture both the inter-GoP and intra-GoP correlation, so it can accurately reflect the queue behaviors with both small and large buffer sizes. The statistic properties of instantaneous incoming video traffic that have greatest impacts on queue behavior are estimated in this model, so the proposed traffic model is used for network performance evaluation of IPTV systems. Extensive simulations with the network simulator demonstrate the feasibility and effectiveness of the proposed traffic model. Thus, it will be an effective tool for performance evaluation of IPTV services via analysis and/or simulations.

Second, to quantify the number of IPTV connections being supported with satisfactory QoS and determine the network performance with very bursty IPTV traffic, a fluid flow based analytical framework is developed in the thesis. Simulation results with another simulator validate the accuracy of the analysis. The analytical approach could be applied for different network scenarios considering the properties of the video traffic and the accompanying network characteristics.

Third, the fluid flow based analytical framework is developed. Given the queueing system, i.e., the generating matrix of the underlying Markov chain, arrival rate and service rate at each state, the queue distribution can be obtained by solving a uniform derivative equation. The loss rate of traffic with a given queue length is need to be determined. In other words, the maximum buffer size will be used to support IPTV connections with guaranteed QoS. By quantifying queue performance analytically, the admission region of IPTV traffic will be obtained. It can help service providers improve the design and deployment of networks to support IPTV traffic and choose a proper resource (buffer and/or bandwidth) allocation scheme. In addition, it provides
important insights into which system parameters and/or traffic characteristics affecting the admission region of networks.

Since successive video frames do not vary much visually, Auto Regressive models have been used to model the output bit rate of VBR encoder. Usually a video source is approximated by a continuous fluid flow model. In the model, the output bit rate within a frame period is constant and changes from frame to frame according to the AR(1) model. Two correlated Markov process models (one in discrete time and one in continuous time) are shown to fit the experimental data and are used to model the input rates of several independent sources into a statistical multiplexer. The continuous-time model is used with a flow equivalent approximate queueing analysis to obtain the common buffer length distribution. The analysis is validated with computer simulations that use the discrete time source model and take into account the discrete nature of the packet queue. Numerical results are presented for variable channel utilization and number of multiplexed video sources.

2.2 Questions for Investigation:

Answers to the following questions shall be given in this thesis:

- Traditional IPTV network data transmission is synchronous or not.
- If not what can be the solution?
- How does the solution work?
- Is the theoretical model in line with simulation?
- What is the effect of the bandwidth factor on the system?
- The analytical model accuracy for FIFO.
- Data lost rate probability in scheduled queue system.
- How the proposed model improves the performance of the whole network for uplink data rate without jeopardizing QoS?
3. IPTV System:

3.1 IPTV Services:

IPTV service distributes broadcasting data to multiple users through the Network. IPTV delivery networks have a different starting point: IP-based, they have been initially designed and deployed for a much more flexible and general service, starting with data services, voice services, video streaming delivery to the TV, and other services, namely, low delays and negligible packet-loss rates comparable to those of an expensive dedicated TV distribution network. This service can have a main IPTV server in multicast server farm and several mirror servers in multicast local server farms. The data of this service can be live data or recording data. A user can enjoy and change a channel in several channels of service provider.

IPTV services can be provided through IP unicast or IP multicast or broadcast depending on the service type and the required distribution efficiency. For both delivery methods and for all services, reliability is an essential asset for successful service deployment and acceptance. IPTV services, networks must be able to scale to millions of customers, maximize bandwidth resources, and provide quality of service (QoS), Admission Control, Video broadcast channel change time, Comprehensive service availability, Service lifecycle and security on an end-to-end basis.

3.1.1 IPTV Video on Demand (VoD):

VoD is a service which provides television programs per the demands of the subscribers. The users interactively request and can receive television channels. These television services are beamed from previously stored media consisting of entertainment movies or education videos. It has a live access through live connection, such as news events in real time. The VoD application provides freedom to the individual subscribers to select a video content and view it at their convenience.
Content on an IPTV video on demand system is unlimited-recorded lectures for the education sector, training and reference material for business, movies for the hospitality industry. IPTV applications and potential revenue-generating services, such as video telephony and video conferencing, remote education, and home security/monitoring cameras, will be available. There are also some additional features and services available, which are much more advanced in comparison to traditional broadcast television systems. In addition to providing the basic television services and features, IP Television can provide the following advanced features and services:

- Anywhere Television Service.
- Global Television Channels.
- Personal Media Channels.
- Addressable Advertising.

3.1.2 Time-Shifted TV:

Time–Shifted TV allows subscribers to watch their favorite broadcast TV program at a more convenient time within an expanded time window. One example of Time–Shifted TV is program restart. For example, one program normally broadcasts from 8:00 pm to 9:00 pm. With program restart, service providers can make that program available for viewing at any point between 8:00 p.m. and 10:30 p.m.

![Figure 3.1: Time–Shifted TV viewing](image)

From the above figure 3.1:

- Broadcast Period: Actual time the program is being broadcast to all viewers.
• Start Window: Time frame subscribers can begin watching a program.
• View Window: Time frame subscribers can view a program.

During watch a Time-Shifted TV program, the viewer may pause, rewind or fast forward. To support viable Time-Shifted TV services, service providers need a video server that can

• Simultaneously ingest, store and stream video services
• Sustain massive ingest capacity.
• Play ingested video streams within 5 seconds.
• Support ultra-high concurrency rates.
• Provide carrier-class reliability.

For time-shifted TV, the network provider stores the programs in a (circular) buffer somewhere in the network. In fact, the time-shifted TV buffer is effectively a scaled up version of the fast-channel change circular buffer. However, for time-shifted TV, the buffer must be a lot larger and to mitigate the resource demands placed on the network by the unicast flows, the time-shifted TV buffer should be located close to the edge of the network. Some operators may choose to offer a limited time-shifted TV functionality supported by features in the STB. In this case only live-pause and rewind are possible. A user cannot start to watch the beginning of a program after it started, nor temporarily watch another program while live pausing the first one on a bandwidth constrained network.

3.1.3 Broadcast TV Service:

In general, all of today’s pay TV networks use some form of encryption to secure the video and audio components of their program services to maintain control over the distribution of their signals. First became a major concern when satellites began to make it possible for TV signals to be transmitted live from one part of the world to any other. Each IPTV channel is sent only once from the video headend to the network, independent of the number of potential TV broadcast receivers. The distribution to all subscribers is achieved by multicast implementations in the core and the access networks [3].
3.1.3.1 Effect of Broadcast TV (IPTV) services:

Broadcast TV system is to do with moving pictures and sounds. The transmission would be on the basis of digital signals. The total number of IPTV channels streamed on-line determines the total bandwidth requirements. The total transmission rate of the IPTV content measured in Mbit/s equals the sum of all concurrent streams. For example, if 30 IPTV channels are broadcasted and each channel is encoded by H.264 codec providing a gross bit rate of 2 Mbit/s (incl. Ethernet overhead), 60 Mbit/s bandwidth is required for the IPTV service. The calculated 60 Mbit/s IPTV traffic will be transmitted via the network operator’s IP core network to the DSLAMs independent of the number of end-customers. This amount of traffic does not affect the throughput of the IP core network dramatically.

3.1.5 Near Video on Demand (NVOD):

NVOD service is a consumer based video service which is broadcasted by multi-channel broadcasters using high-bandwidth distribution mechanisms. Multiple copies of a program are broadcast at short time intervals (typically 10–20 minutes) providing convenience for viewers, who can watch the program without needing to tune in at a scheduled point in time. The customer has no control over the session except in choosing which program to watch.

NVOD system and method for incorporating and/or updating a commercial (CM) or promotion video program. This system for use in a subscription television broadcasting system which is called a pay television system. If a pay television broadcasting station is able to establish an NVOD system which broadcasts only one set of video movie materials and CM video materials at certain time intervals in a plurality of channels, then such an NVOD system will be highly convenient and inexpensive for users.
3.2 IPTV Network Architecture:

IPTV network receives video streams and stores it in a local storage which treat as a sender. The users are at the receiving end. So we have a multicast environment where the video stream will be transferred through an allocator and an edge router to process the request and response of the receivers and the Host devices.

The IPTV network connects the following devices:

- SHE: Central video server location.
- VHO: Regional video server location
- VSO: Video aggregation office.
- DSLAM: Connects metro and access networks.
- RG: Connects access and home networks.
- STB: User terminal for system interaction.

![Figure 3.2: A typical IPTV network architecture [14].](image)

This architecture that supports high bandwidth, multicast group management, dynamic policy-driven resource control, subscriber management, and home networking while providing the service provider with the ability to continuously monitor and ensure the subscriber’s quality of experience.
3.3 The IPTV network:

The IPTV network consists of four main components as shown (Fig 3.3): Core network, Metro network, Access network, and Home network. The simplified IPTV network flowchart is depicted below.

![Simplified IPTV network flowchart](image)

**Figure 3.3: An illustrative network deployment of IPTV network [15].**

3.3.1 Core Network (CN):

The core network groups the encoded video streams into the respective channel line up. The core network is unique to the service provider, and often includes equipment from multiple vendors. The core network that connects the head end to the local exchange, live television channels are carried as unicast / multicast streams. IPTV traffic can be separated from other non real time data traffic to guarantee the high level of its QoS requirements [1]. The core network were digital, it was possible to offer services such as the integrated services digital network, which offered higher speed digital communications.
3.3.2 Metro Network (MN):

The metro network is a key part of the IPTV architecture which transporting traffic between core network and access network and also providing transport-based connectivity services. Metro networks interacting with legacy systems, which eventually lead towards an ambient environment in which IP technology will provide transparent connectivity. The metro network must be built from a consistent range of inter-operable equipment, packaged for local conditions to deliver the very same quality of experience to all consumers. It built for IPTV by requiring higher bandwidths, flexible any-to-any connectivity, and the ability to insert customized content closer to the edge of the network. Metro networks node using a combination of packet and wavelength processing to aggregate traffic for efficient transport and presentation to the service edge equipment. High bandwidth video and enterprise services will typically justify dedicated wavelength transport in the metro network, close to the access nodes. The metro network faces three key challenges:

- The metro convergence challenge – a single network to deliver all services.
- The metro flexibility challenge – a network optimized for every situation.
- The metro cost challenge – a network with low cost of ownership.

3.3.3 Access Network (AN):

The access network is the link from the service provider to the individual household. The broadband connection between the service provider and the household can be accomplished using a variety of technologies. IPTV networks will use variants of asymmetrical DSL (ADSL) and very high speed DSL (VDSL) to provide the required bandwidth to run an IPTV service to the household. The service provider will place a device (like a DSL modem) at the customer premises to deliver an Ethernet connection to the home network.

3.3.4 Home Network (HN):

The home network is also known as ‘the last meter’ which distributes the IPTV service throughout the home. There are many different types of home networks, but
IPTV requires a very robust high bandwidth home network that can only be accomplished today using wire line technology. The end point in the home network, to which the television set is connected, is the set-top box (STB). The home network integrates the delivery of data, voice and video (IPTV) traffic. Home network needs to efficiently and effectively manage network resources to guarantee a high level of user-perceived Quality of Service (QoS) for triple play service.

3.4 NGN-based IPTV solution:

IPTV services can be enabled via the telecom broadband network or mobile Internet, the bi-directional digital TV broadcast network. NGN is able to provide overall services, which includes: voice, data, video, streaming, Internet access, TV broadcast and mobile services. Regardless of the bandwidth requirement, terminal type, fixed or mobile, each service can find a suitable service interface in NGN network. As a type of NGN service, IPTV caters to the development trends of NGN.

3.4.1 Generic NGN model:

The NGN concept has been addressed, thoroughly discussed, and well defined from both the research and development sector and the standardization and regulatory bodies; among these are foremost, the International Telecommunication Union Telecommunication (ITU-T), the Third Generation Partnership Project (3GPP), and the European Telecommunications Standards Institute (ETSI) TISPAN. There are several architectural proposals available; nevertheless, they all share certain basic characteristics, represented within the following generic NGN model (Fig. 3.4a):

- IP multimedia subsystem (IMS), providing core session control, service triggering, and Authentication, Authorization and Accounting (AAA) mechanisms.
- Network attachment subsystem (NASS) for the end user’s device initialization and network attachment procedures.
- Resource and admission control subsystem (RACS) for policy enforcement, admission control and resource management.
3.4.2 Generic vertical IPTV model:

From the architectural viewpoint, current IPTV deployments are typically proprietary based vertical solutions, comprising four segments (Fig 3.4b)

- Content provisioning, where content and the associated metadata are ingested, aggregated, and prepared through adapting processes and digital rights management (DRM).
- IPTV control, implementing service provisioning control and AAA functionalities (middleware).
- Content delivery, where content is packed into services and delivered to the end users.
- Customer premises, represented by a set of content acquisition and processing functionalities and service execution functionalities within end user equipment.

The content provisioning, IPTV control, and content delivery segments together are
known as the head-end. The advantage of a vertical approach is a dedicated platform for a chosen service type, namely, IPTV. It has also disadvantages for maintenance of separate parallel vertical networks which are

- There are dedicated networks for voice over Internet protocol services, Web services and video streaming services.
- The incapability of blending voice, data, video, presence, and messaging services due to separate and conceptually incompatible provisioning platforms.

Traditional services are based on a vertical network structure, with its service-unique functionality for management, provisioning, charging, is very costly and complex to build and maintain. Separate implementations of each layer must be built for every service, and the structure is replicated across the network, from the terminal via the core network to the other user's terminal. This vertical convergence and integration strategy is critical to reduce network complexity, lower capital and operational expenses, and even more importantly enhance the network’s ability to quickly and effectively provide new services and revenues.

Moreover, a vertical IPTV solution was proven to be considerably walled and proprietary-based, which presents issues of interconnectivity, multi-vendor environments, and third party provisioning. The possible solution to these challenges is offered through the deployment of IPTV services within the NGN environment.

### 3.5 IPTV service delivery within NGN architecture:

IPTV is developing functional architecture models based on NGN architecture model such as using "Transport Stratum" and "Service Stratum". Combining IPTV with NGN will bring new opportunity to open next generation business infrastructure not only for enterprises but also end users.
3.5.1 NGN architecture framework:

NGN provides certain mechanisms to support end-to-end QoS including security concerns. For this, NACF and RACF are being developed for provide better mechanisms to support manageable capabilities than legacy IP based networks. These two functions collaborate with various functions in "Service control" of NGN functional architecture such as CSCFs and "Media Resource Control Function" as well as direct coordination between two NACF and RACF functions. As a result of these, NGN provide mechanisms to support delivery end-to-end QoS and security.

Physical transport networks provide the connectivity for all components and physically separated functions within the NGN. Transport is divided into access transport networks and the core transport network, with a border gateway linking the two transport network categories.

End-user interfaces are supported by both physical and functional (control) interfaces, and both are shown in the fig 3.5. No assumptions are made about the diverse end-user interfaces and end-user networks that may be connected to the NGN access network. All categories of end-user equipment are supported in the NGN, from single-line legacy telephones to complex corporate networks. End-user equipment may be either mobile or fixed.

The NGN interface(s) to other networks includes many existing networks, such as PSTN/ISDN and the public Internet. The NGN interfaces other networks both at the service stratum level and at the transport stratum level, by using border gateways. The border gateways may involve media transcoding and bearer adaptation. Interactions between the service stratum and transport stratum may take place, either directly or through the RACF. In release 2, some configurations are identified in the service stratum: IP multimedia service, PSTN/ISDN emulation service, and IPTV service configurations. Regarding the transport stratum, multiple configurations are represented in the access transport area.
In the NGN network both the NGN service stratum and the NGN transport stratum, the general architectural concepts of data (or user) plane, control plane and management plane can be logically identified.

- **NGN service stratum**: This provides the user functions that transfer service-related data and the functions that control and manage service resources and network services to enable user services and applications.

- **NGN transport stratum**: This provides the user functions that transfer data and the functions that control and manage transport resources to carry such data between terminating entities. The transport stratum facilitates like user-to-user connectivity; user-to-services platform connectivity; services platform-to-services platform connectivity.
3.5.2 Resource and Admission Control Functions (RACF):

In the NGN architecture, the RACF (Resource and Admission Control Functions) acts as the arbitrator between SCF (Service Control Functions) and TF (Transport Functions) for QoS related transport resource control within access and core networks. The RACF makes the policy decisions based on transport resource status and utilization information. The RACF interacts with transport functions for the purpose of controlling one or more of the following functions in the transport stratum: bandwidth reservation and allocation, packet filtering; traffic classification, marking, policing, and priority handling; network address and port translation; and firewall.

![Resource and Admission Control Functional Architecture](image)

Figure 3.6: Resource and Admission Control Functional Architecture [19].

As showing in fig-3.6, The RACF executes policy-based transport resource control upon the request of the SCF, determines transport resource availability, makes admission decisions, and applies controls to transport functions for enforcing the policy decisions. The RACF consists of two types of resource and admission control functional entities, the PD-FE (Policy decision functional entity) and the TRC-FE (Transport resource control functional entity). This decomposition of PD-FE and TRC-FE enables the RACF to support a variety of access and core networks (e.g. fixed and mobile access networks) within a general resource control framework.
Therefore, if IPTV services are provided over an IMS-enabled network, it makes sense to integrate the control of IPTV multicast channels with the RACF function. With this integration, decisions on how many and which channels are sent to access nodes is taken based on more information than only channel usage, but also from the usage of other services and the resource monitoring performed by RACF.
CHAPTER-4

4. Traffic Control:

The traffic control goal is to protect both the receiving terminal equipments and the network elements (e.g. multiplexers, switches, etc.) against traffic excess. Once the traffic is controlled, it becomes more predictable for the network and resource utilization can be optimized. For the traffic controlling and administering across a network, mainly including bandwidth allocation, admission control, packet classification/marking, congestion avoidance, traffic policing, and traffic shaping, etc.

Traffic control is concerned with the three-way relationship between traffic volume, network resources and realized QoS and GoS. Traffic control methods can be classified as preventive or reactive. Preventive traffic control aims at avoiding congestion from occurring too often. Reactive traffic control resolves congestion situations occurring during the information transfer phase.

Figure 4.1: The general processing sequence of traffic control [20].

4.1 Performance measures:

Networks traffic performance is measured in terms of quality of service (QoS), Quality of Experience (QoE) and Grade of Service (GoS). Successful transmission
delivery is measured by the highest quality at the lowest prices and user behaviour and user experience of the network. Network operators also need to bring new technology into service very quickly to create competitive advantage.

4.1.1 Quality of Experience (QoE):

QoE is defined in as the overall acceptability of an application or service, as perceived subjectively by the end-user. It includes the complete end-to-end system effects (client, terminal, network, services infrastructure, etc) and may be influenced by user expectations and context. The QoE requirements are defined from an end user perspective and are agnostic to network deployment architectures and transport protocols. The QoE requirements are specified as end-to-end and information is provided on how they influence network transport and application layer factors.

Hence the QoE is measured subjectively by the end user and may differ from one user to the other. Contributing to the QoE are objective service performance measures such as information loss and delay. Those objective measures together with human components that may include emotions, linguistic background, attitude, motivation, etc and determine the overall acceptability of the service by the end user.

![QoE Dimensions](image)

Figure 4.2: QoE Dimensions [21].

To ensure that the appropriate service quality is delivered, QoE targets should be established for each service and be included early on in system design and engineering processes where they are translated into objective service level
performance metrics. Quality of experience will be an important factor in the marketplace success of triple-play services and is expected to be a key differentiator with respect to competing service offerings.

4.1.2 Quality of Service (QoS):

Quality of Service (QoS) is essential in systems with limited capacity, which is defined as the ability of the network to provide a service at an assured service level. Quality of service (QoS) is defined in (ITU-T E.800) as the collective effect of performance which determines the degree of satisfaction of a user of the service. In telecom QoS is usually a measure of performance of the network itself. QoS mechanisms include any mechanism that contributes to improvement of the overall performance of the system and hence to improving end user experience. QoS mechanisms can be implemented at different levels. For example at the network level it includes traffic management mechanisms such as buffering and scheduling employed to differentiate between traffics belong to different applications. Other QoS mechanisms at levels other than the transport include loss concealment, application forward error correction (FEC), etc.

Related to QoS are the QoS performance parameters. Similar to the QoS mechanisms QoS parameters can be defined for different layers. Common QoS parameters used for characterizing the network performance are [34]:

- Bandwidth (throughput): Number of bits or bytes transmitted over the network in a specific time period.

- Delay: the time it takes for the data packet to traverse from its source to its destination. It consists of three components: propagation delay, transmission delay, and queuing delay.

- Delay jitter: The variation in delay encountered by a data packet. This is the difference between the maximum and the minimum possible packet delay.
• Loss probability: The chance of a packet being lost in the network. There are a number of situations that may result in the loss, such as buffer overflow in the network switching nodes or a call set-up request denial.

• Utilization: The ratio of busy time to the total elapsed time in a given period. It can be measured in each of the network elements like sources, switches, and links.

The weakness of existing best-effort IP networks requires additional network QoS mechanisms to resolve these needs:

• To provide differentiated service to different classes of traffic.
• For basic mechanisms that measure use of network resources, and operator’s service level agreements.
• For more predictable response in the face of real-time transient congestion.

Quality of service guarantees are important if the network capacity is insufficient, especially for real-time streaming multimedia applications such as voice over IP, online games and IPTV, since these often require fixed bit rate and are delay sensitive, and in networks where the capacity is a limited resource. For example, a real-time application might require QoS guarantees such as an end-to-end packet delay of 200 ms, packet loss rate of $10^{-6}$ and 50 ms jitter.

4.1.3 Grade of service (GoS):

Grade of Service (GoS) is the traffic related part of network performance (NP), defined as the ability of a network or network portion to provide the functions related to communications between users. Grade of Service (GoS) has been used in the telecommunications industry to indicate components which contribute to overall quality of service what the user experiences.

Grade of service is the probability of a call in a circuit group being blocked or delayed for more than a specified interval, expressed as a vulgar fraction or decimal fraction.
This is always with reference to the busy hour when the traffic intensity is the greatest. Grade of service may be viewed independently from the perspective of incoming versus outgoing calls, and is not necessarily equal in each direction or between different source-destination pairs [4]. When a user tries to make a call, the system handling the call determines whether to accept or reject the call. Rejected call can be waited for a certain time or lost depending on the method used.

4.2 Preventive Control Schemes:

Preventive control methods provide a fair allocation of bandwidth by requiring, at times of high network load, that each connection’s traffic flow remains within specified bounds appropriate for the supported service. Preventive control does not wait until congestion actually occurs, but rather tries to prevent the network from reaching an unacceptable level of congestion. The most common and effective approach is to control traffic flow at entry points to the network (i.e., at the access nodes). This approach is especially effective in connection-oriented transport. Preventive control for IPTV network can be performed in two ways: admission control and bandwidth enforcement [7]. Admission control determines whether to accept or reject a new connection at the time of call setup. This decision is based on traffic characteristics of the new connection and the current network load. The bandwidth enforcement monitors individual connections to ensure that the actual traffic flow conforms to that reported at call establishment.

4.2.1 Traffic Policing:

The traffic policing in IPTV networks is one of the most critical preventive congestion control mechanisms. It intends to ensure that each source conforms to its traffic parameters negotiated during the Call Admission Control phase. The well known Leaky Bucket mechanism for police the distribution of the traffic. This method can enforce the average bandwidth and the burst factor of a traffic source. One possible implementation of a Leaky Bucket method is to control the traffic flow by means of tokens. A queuing model for the Leaky Bucket method is illustrated in Fig 4.3.
An arriving cell first enters a queue. If the queue is full, cells are simply discarded. To enter the network, a cell must first obtain a token from a token-pool; if there is no token, a cell must wait in the queue until a new token is generated. Tokens are generated at a fixed rate corresponding to the average rate of the connection. If the number of tokens in the token pool exceeds some refined threshold value, the process of token generation stops. This threshold value corresponds to the burstiness of the transmission the larger the threshold value, the bigger the burstiness [7]. This method enforces the average input rate while allowing for a certain degree of burstiness. The Leaky Bucket method can also enforce the peak bandwidth by generating tokens at the rate corresponding to the peak rate.

4.2.2 Traffic Shaping:

Traffic shaping also known as "packet shaping" is the control of computer network traffic in order to optimize or guarantee performance, lower latency, and/or increase usable bandwidth by delaying packets that meet certain criteria. Shaping of traffic means smoothing out any traffic bursts. Traffic shapers do not want to discard violating traffic, but to store them in actual buffers and smooth it out. Traffic shaping is a traffic control mechanism which actively alters the traffic characteristics of a stream of cells. It can perform the following actions: reduce the peak cell rate, limit

![A queuing model for a leaky bucket method.](image)

Figure 4.3: A queuing model for a leaky bucket method. [7].
the burst length or reduce the cell delay variation by suitably spacing cells in time. Traffic shaping provides a means to control the volume of traffic being sent into a network in a specified period (bandwidth throttling), or the maximum rate at which the traffic is sent (rate limiting), or more complex criteria such as GCRA. The primary reasons for using traffic shaping are to control access to available bandwidth, to ensure that traffic conforms to the policies established for it, and to regulate the flow of traffic in order to avoid congestion that can occur when the sent traffic exceeds the access speed of its remote target interface.

The shaping function could either be performed by the access control at a user-network interface or at a data source by buffering and injecting cells into a network at a slower speed. Since traffic shaping reduces network congestion by suppressing inputs to the network. Traffic shaping uses a token-bucket system to determine whether to transmit, delay, or drop new packets. The key to good traffic-shaping design is to create a bucket that will constantly have enough tokens to either queue and forward each packet, and replace tokens after packets have been removed from the buffer and transmitted.

4.2.3 Call Admission Control:

Call Admission control is used for networks which provide QoS guarantees and QoS enhanced IP networks. It is responsible for maintaining the QoS and network availability while the network operators still achieves optimal utilization of their network resources for maximum revenue in competitive environment. Call admission control (CAC) is a preventive traffic control and enforcement scheme that decide for each new call request, maybe to be accepted or if it must be rejected. Preventive call admission control (traffic and resource management) in this work depends on the function of evaluated performance in terms of overflow probability and waiting time. This helps to support a certain quality of services (QoS) requirements by strategically limiting the number of call connections into the networks in order to reduce the network congestion and call dropping. Good CAC schemes should balance the call blocking and call dropping in order to provide the desired QoS requirements.
The main requirements in a CAC procedure can be summarised as follows:

- The network must be protected from overload.
- Resources must be allocated in such a way that the QoS requirements are met for all established connections.
- Maximal statistical multiplexing gain should be obtained.
- The required real-time processing should be reasonable.

Call admission control can be tricky, because the volume of traffic in communications networks is inherently chaotic or bursty, and traffic bursts are virtually impossible to predict. Another problem is that the actual content of a call may not conform to its descriptor.

### 4.3 Reactive Control Schemes:

Reactive control is closed-loop control, in which the source node traffic is regulated by the destination node and/or the network. In this method, the destination node or the network transmit special Control/status information to the source node. The source node examines this information and responds accordingly. At the onset of congestion, reactive control instructs the source node to throttle its traffic by sending feedback to the source node. It is important to observe that the reactive control reacts to the congestion after it happens and tries to bring the degree of network congestion to an acceptable level.

But the main problem with reactive control in high-speed networks is slow feedback. The effects of high-speed channels make the overhead due to propagation delay significant. Therefore, by the time that feedback reaches the source nodes and the control is triggered, it may be too late to react effectively. It is a possible to overcome this problem, if reactive control is performed between network users and the edge of the network then the effect of propagation delay may not be significant since the distance feedback information propagates is short. However, this limits the reactive control to the edge of the network. Reactive control may not be effective in an ATM
environment because of the above problem. In general, reactive becomes less effective when the bit-length increases and congestion-period-to-propagation-delay decreases, due to the large transient traffic and sluggish response time. Conventional terrestrial networks normally use reactive control because of the relative small bit-length, and optical-fiber based networks tend to use preventive control. With reactive control, it is usually not easy to provide quality of service guarantees since the bandwidth that each connection can use depends on the actual network load condition. Lack of quality of service guarantees makes reactive control incapable of supporting real-time applications like interactive audio/video transmissions.

4.3.1 Flow Control:

Flow control is the process of managing the rate of data transmission between two nodes. This should be distinguished from congestion control, which is used for controlling the flow of data when congestion has actually occurred. Flow control mechanisms can be classified by whether or not the receiving node sends feedback to the sending node. The flow control negotiated quality of service (QoS) during transient overload conditions. It allows the source to exceed its negotiated throughput when there is unused capacity available along the path of communication.

Flow control is important because it is possible for a sending computer to transmit information at a faster rate than the destination computer can receive and process them. This can happen if the receiving computers have a heavy traffic load in comparison to the sending computer, or if the receiving computer has less processing power than the sending computer.

4.3.2 Congestion Control:

Congestion happens whenever the input rate is more than the available link capacity

\[ \sum \text{Input Rate} > \text{Available link capacity} \]

Most congestion control schemes consist of adjusting the input rates to match the
available link capacity. On the high-speed channels, congestion control is a challenge for an IPTV network. High-speed channels significantly limit the congestion control schemes applicable. Reactive control possibly hardwired protocols are preferred in IPTV networks in order to match the high speed of the network channels. For example, replacing 1 Mbits/s channel with 1 Gbits/s channel reduces the cell transmission time from 0.5 ms to 0.5 µs [7]. On the other hand, the time required to process a protocol remains the same. As a result, in a high-speed network environment protocol processing time can be a bottleneck. In order to avoid such a bottleneck, the networks use simplified protocols, pushing most of the link-by-link layer protocols to higher edge-to-edge layers. This makes it difficult to implement link-by-link congestion control schemes. For these reasons, many of the congestion schemes developed for existing networks may not be applicable to IPTV networks. Many of the congestion control schemes developed for existing networks fall in the class of reactive control [7]. Reactive control reacts to the congestion after it happens and tries to bring the degree of network congestion to an acceptable level. However, reactive control is not suitable for use in IPTV networks.
CHAPTER-5

5. VBR Video Traffic:

Variable bit-rate used in sound or video encoding. VBR files vary the amount of output data per time segment. In a video distribution environment, where several links with different capacities are employed, it is likely to find nodes with less capacity than the offered traffic by the distributed bit stream. VBR allows a higher bit-rate (and therefore more storage space) to be allocated to the more complex segments of media files. Video people want to send video at variable bit-rate since it for guarantee of QoS for VBR traffic and the optimal rate-distortion performance. Variable bit rate traffic generated by multimedia and video services may experience cell loss when transmitted over IPTV networks.

5.1 Overview of MPEG Encoding:

MPEG (Moving Pictures Experts Group) is a group of researchers who meet under the ISO (International Standards Organization) to generate standards for digital video (sequences of images in time) and audio compression. In particular, they defined a compressed bit stream, which implicitly defines a de-compressor [52].

MPEG encoding is the process of capturing (digitizing) or converting (re-encoding) video and audio to one of several MPEG video and/or audio standards for distribution and/or archiving to optical disc (CD, DVD). The coding syntax that MPEG has defined provides tools to cover different applications, and parameters can be chosen to allow working at different bit-rates, picture sizes and resolutions etc. The primary focus of MPEG encoding algorithms is:

- To create fast and efficient motion vector search techniques.
- To find "good" encoding parameters that provides a balance of encoding speed, compression and quality.

The MPEG encoder input sequence consists of a series of frames, each containing a
two-dimensional array of picture elements, called pels. The number of frames per second as well as the number of lines per frame and pels per line depends on national standards. For each pel, both luminance and chrominance information is stored. The compression algorithm is used to reduce the data rate before transmitting the video stream over communication networks. For that MPEG uses the DCT compression algorithm and defines how it is used to reduce the data rate, how packets of video and audio data are multiplexed together in a way that will be understood by an MPEG decoder. MPEG achieves high video compression by using two main compression techniques [53]:

- Intra-frame compression: Compression within individual frames (also known as spatial compression because the compression is applied along the image dimensions).

- Inter-frame compression: Compression between frames (also known as temporal compression because the compression is applied along the time dimension).

The intra-frame compression is performed by transforms and entropy coding. The inter-frame compression is performed by prediction of future frames based on the motion vector. This is achieved using three types of frames:

- I-frames are Intra-frame coded frames that need no additional information for decoding.

- P-frames are forward predicted from an earlier frame with the addition of motion compensation. The earlier frame could be an I or a P-frame.

- B-frames are bi-directionally predicted from earlier or later I or P-frame.

![Figure 5.1: MPEG frame pattern [43].](image)
Typically, I-frames require more bits than P-frames. B-frames have the lowest bandwidth requirement. After coding, the frames are arranged in a deterministic periodic sequence, e.g. “IBBPBB” or “IBBPBBPBBPBB” which is called Group of Pictures (GOP).

5.1.1 MPEG-2:

MPEG also established the MPEG-2 standard for high-quality video playback at higher data rates between 1.5 to 6 Mbps. MPEG-2 is a superset of MPEG-1 intended for services such as video-on-demand, DVD (digital video disc), digital TV, and HDTV (high definition television) broadcasts with additional features, frame formats and encoding patterns [52]. The basic requirements of MPEG-2 video coding are a high compression ratio with good image quality and the support of a number of optional features, such as random access, fast search, reverse playback, etc.

MPEG-2 makes extensive use of motion compensated prediction to eliminate redundancy. The prediction error remaining after motion compensation is coded using DCT, followed by quantization and statistical coding of the remaining data. The most significant extension of MPEG-2 main profile over MPEG-1 introduces the concept of video fields and interlaced frames. Variable quantization is extended, DC precision can be customized, and another coding table is defined. Higher resolutions, bit rates, and frame rates are expected. Parameter ranges such as picture width, height, bit rate, and buffer sizes are extended. Copyright information can be recorded. Display characteristics can be transmitted including an offset for viewing a subset of the picture in a window. This main profile uses all three frame types I, P, and B. This profile is commonly used for deploying multicast IPTV services over a broadband network.

Overall, MPEG–2 is a well-defined, stable compression system with a wide variety of applications. Hundreds of millions of devices installed around the world are capable of receiving and decoding MPEG–2 video in a wide variety of flavors. MPEG–2 is
commonly used in contribution, distribution and delivery networks. It is interesting to note that, for video signals coded at bit rates below about 3 Mbps, MPEG-1 may be more efficient than MPEG-2. MPEG-2 video decoding and encoding are more CPU intensive than for MPEG-1 [51].

5.1.2 MPEG-4:

MPEG-4 is a more powerful compression algorithm, with multimedia access tools to facilitate indexing, downloading, and querying. Initially, MPEG-4 was aimed primarily at low bit-rate video communications; but over time its focus changed beyond just high compression. MPEG-4 will provide a syntactic description language (MSDL). MPEG-4 is efficient across a variety of bit-rates ranging from a few kilobits per second to tens of megabits per second. MPEG-4 uses object-based coding which is different from the frame-based coding used in MPEG-1 and MPEG-2. MPEG-4 provides the following functionalities:

- Improved coding efficiency.
- Ability to encode mixed media data (video, audio, speech).
- Error resilience to enable robust transmission.
- Ability to interact with the audio-visual scene generated at the receiver.

MPEG-4 is designed for a wide variety of networks with widely varying performance characteristics. A three-layer system standard for MPEG-4 was developed to help MPEG-4 interface and adapt to the characteristics of different networks. The synchronization layer adds the timing and synchronization information for the coded media. The flexible multiplex layer multiplexes the content the coded media. And the transport multiplex layer interfaces the coded media to the network environment. This three-layer system makes MPEG-4 more versatile and robust than the MPEG-1 and MPEG-2 system [54].

One potential drawback of MPEG–4 is that decoders are more complex for MPEG–4 than for MPEG–2. According to the MPEG–4 Industry Forum (www.m4if.org), an MPEG–4 decoder will be 2.5 to 4 times as complex as an MPEG–2 decoder for
similar applications. This means more complicated hardware devices and greater demand on processor resources for software decoders [51].

Overall, MPEG–4 is an exciting new collection of technologies that promises to greatly increase the amount of video information that can be squeezed into a given amount of network bandwidth. Through MPEG–4 AVC, much more efficient video coding is possible, and the variety of object types available makes integration with computer-generated graphics simple and extremely bandwidth efficient. Because of MPEG–4’s complexity and its relative newness, much development work needs to be done to reach the level of sophistication and maturity enjoyed by MPEG–2 technologies [51].

5.2 Layer video traffic modelling:

Some layers for video traffic modelling are representing in below. For noise cancelling, modeling and prediction, and echo cancellation are examples of such applications.

5.2.1 Scene layer:

The scene layer is the visual content and captures the video stream non-stationarity while the scene layer captures the stationary behaviour within the scene. The visual content is a result of the complete video production process (i.e., shot composition, scene editing, object motion, etc). The particular characteristics of the bit stream within the scene are due to the video compression mechanism.

5.2.2 GOP Layer:

Group-of-Picture Layer contains information relevant to a group of pictures (GOP). A GOP is defined as a set of pictures that are coded without any dependencies to other pictures. The rational choice of modelling at GOP layer is that GOP reflects the behaviour of video scene activity and plays the most important role concerning the autocorrelation effects. In other words, there is a direct correspondence between GOP sizes and the actual image sizes. Hence, GOPs always start with an I-picture and may
serve as a random access point for applications that need such. A GOP should be self-contained as long as the sequence header information is available. The GOP layer is the appropriate place for information concerned with all pictures of the GOP. The bit rate averaged over a GoP varies in a correlated way from GoP to GoP as the image content changes; changes can be gradual within a scene or sharp in the event of a change of scene.

5.2.3 Frame Layer:

The MPEG algorithm introduces systematic variations from frame to frame due to the pattern of frames. The frame layer introduced to achieve the correlation among I, P, and B frames (order of p = 25, 100, and 35 respectively). Let \( c \in \{I, P, B\} \), then the frame size is estimated using the equation:

\[
X^c(n) = \sum_{k=1}^{p} a_i^c X^c(n-k) + e^c(n)
\]

Where \( e^c(n) \) is the i.i.d Gaussian error. To ensure sufficient correlated I, P, and B components, correlation of I, P, and B prediction errors was used to generate the model for the aggregate MPEG sequence.

5.2.4 Packet Layer:

During an established Internet connection many packets with different sizes are sent and received. Packet layer provides a packet sequence number based flow control mechanism between a source and a destination. The packet layer send-sequence number \( P(S) \) identifies the current packet with respect to the packet header sequence number modulus. A receiver uses \( P(R) \) in the acknowledge packet to indicate the send-sequence number of the next expected packet from the sender. Note that this sequence number based packet layer flow control is on a per call basis while the LAPB data link layer flow control is on a per link basis.
5.3 Traffic Characteristics:

5.3.1 Short-Range Dependence (SRD):

Consider a stationary process $X = \{X_t : t = 1, 2, \ldots\}$ with mean $\mu = E[X_t]$ and variance $\sigma^2$. The autocorrelation function and the variance of $X$ are denoted as finite variance

$$r(k) = E[(X_t - \mu)(X_{t+k} - \mu)] / E[(X_t - \mu)^2]$$

and

$$\sigma^2 = E[(X_t - \mu)^2],$$

$X$ is called short-range dependent if it satisfies the following properties:

- Variances decay more quickly (VAR $[X^{(m)}_k]$ decays as $m^{-1}$).
- $r(k)$ decays in an exponential manner
  $$\lim_{k \to \infty} r(k) \approx \gamma^{-k}.$$
- A finite and approximately constant power spectral density occurs about the frequency origin.
- The process is similar to pure noise in a second order sense for
  $$\lim_{m \to \infty} r^{(m)}(k) = 0$$
  Where $k$ is not equal to zero.

A simple, direct parameter, characterizing the degree of SRD, is the Hurst parameter,

$$H = 1 - \frac{\beta}{2}.$$  

If $H = 0.5$, then $r(k) = 0$ and $X(t)$ is SRD because it is completely uncorrelated. For $0 < H < 0.5$, the summation of $r(k)$ is 0, an artificial condition rarely occurring in SRD applications. If $H = 1$, then $r(k) = 1$, an uninteresting case where $X(t)$ is always perfectly correlated. $H > 1$ is prohibited because of the stationarity condition on $X(t)$ [56].

5.3.2 Long-Range Dependence (LRD):

The presence of long-range dependence in a time series indicates that while long-term correlations (large lags) are individually small, their cumulative effect is non-
negligible and produces scenarios which are drastically different from those experienced with traditional short range dependent models such as Markovian processes. Differently, all short-range dependent processes are characterized by an autocorrelation function which decays exponentially fast; processes with long-range dependence exhibit a much slower decay of the correlations - their autocorrelation functions typically obey some power-law. While the commonly made assumptions in standard time series analysis require that observations separated by a large time span are roughly independent, but in practice, it seems to be the rule rather than the exception that long time series of absolute measurements violate this independence assumption and exhibit long-range dependence instead.

Definition:

Let \( X = (X_t; t = 0, 1, \ldots) \) be a covariance stationary (sometimes called wide-sense stationary) stochastic process with mean \( m \), variance \( \sigma^2 \) and autocorrelation function \( r(k), k \geq 0 \) with time lag \( k \). \( X \) is said to exhibit long-range dependence if

\[
r(k) \sim k^{-D} L_1(k) \text{ as } k \to \infty
\]

where \( 0 < D < 1 \), and \( L_1 \) is a slowly varying function, that is, \( \lim_{t \to \infty} L_1(tx)/L_1(t) = 1 \), for all \( x > 0 \) (constants and logarithms are examples of slowly varying functions). Equivalently (under weak regularity conditions on \( L_1 \)), there is long-range dependence in \( X \) if

\[
f(\lambda) \sim \lambda^{-\alpha} L_2(\lambda) \text{ as } \lambda \to 0
\]

where \( 0 < \alpha < 1 \), \( L_2 \) is slowly varying, and \( f(\lambda) = \sum_k r(k)e^{ik\lambda} \) denotes the spectral density function. From (5.1) we see that long-range dependence is characterized by an autocorrelation function that decays hyperbolically as the lag increases, i.e., \( r(k) \sim k^{-D} \). Furthermore, it is easy to see that (5.1) implies \( \sum_k r(k) = \infty \). This non-
summability of the correlations captures the intuition behind long-range dependence, namely that while high-lag correlations are individually small, their cumulative effect counts and gives rise to features which are drastically different from those of autoregressive moving average (ARMA) processes or, more generally, short-range dependent processes. From the point of view of spectral analysis, short-range dependence is characterized by a spectral density function $f(\lambda)$ which is finite for $\lambda = 0$. On the other hand, (5.1) or (5.2) imply that $f(0) = \sum_k r(k) = \infty$, i.e., long-range dependence requires a spectral density which increases without limit as the frequency tends to zero [55].

At least two claims have been made in the literature regarding the importance of the LRD property in ATM traffic engineering:

- While long-term correlations of LRD processes are individually small, their cumulative effect on the CLR (cell loss rate) is non-negligible.

- The buffer behavior of LRD VBR video traffic can-not be accurately predicted by simple, parsimonious Markov-based (or SRD in general) models.

We know Self-similarity is usually used to describe the scaling behaviour of continuous or discrete process, while LRD focus on the tail behaviour of the autocorrelation of a stationary time series. The heavy-tailed distribution is the root of LRD; data bursts in data network may exhibit the heavy-tailed distribution and result in LRD and self-similarity. Although heavy-tailness is not necessary to generate LDR in aggregate traffic but some empirical measurements provide strong evidence that heavy-tailness is an essential component to induce LRD in network traffic.

The LRD property of traffic fluctuations has important implications on the performance, design and dimensioning of the network. A simple, direct parameter, characterizing the degree of LRD, is the Hurst parameter, $H = 1 - \beta/2$. For Long-range dependent processes the Hurst parameter will be between 0.5 and 1.0. A
number of methods have been proposed to estimate the Hurst parameter. Some of the most popular include aggregated variance time (V/T), Rescaled-range (R/S) and the Higuchi and wavelet-based methods, although there are many others.

5.3.3 Self-Similar Process:

A self-similar phenomenon displays structural similarities across a wide range of timescales. Traffic that is bursty on many or all timescales can be described statistically using the notion of self-similarity. Self-similarity is the property associated with ‘fractals’, which are objects whose appearances are unchanged regardless of the scale. In the case of stochastic objects like time series, self-similarity is used in the distributed sense: when viewed at varying timescales, the object’s relational structure remains unchanged. As a result, such a time series exhibits bursts at a wide range of timescales [50].

The following definition of self-similarity for discrete-time stochastic processes is widely adopted. Assume \( X_k \) be a covariance stationary (sometimes called wide-sense stationary) stochastic process; that is, a process with constant mean \( \mu = E[X_t] \), finite variance and an autocorrelation function which is

\[
\delta^2 = E[(X_t - \mu)^2], \text{ and }
\]

\[
r(k) = E[(X_t - \mu)(X_{t+k} - \mu)] / E[(X_t - \mu)^2] \quad (k = 0, 1, 2 ... ) \text{ that depends only on k.}
\]

Let \( X(m) = (X_k^{(m)} : k = 1,2,3,...) \) denote a new time series obtained by averaging the original series \( X \) over non-overlapping blocks of size \( m \). That is, for each \( m = 1, 2, 3... \) \( X_k^{(m)} \) is given by:

\[
X_k^{(m)} = \frac{1}{m} (X_{kn-m+1} + ... + X_{kn}) , \quad k = 1,2,3...
\]

Note that for each \( m \), the aggregated time series \( X^{(m)} \) defines a covariance stationary process; let \( r^{(m)}(k) \) denote the corresponding autocorrelation function.
The process $X_k$ is called exactly second-order self-similar with self-similarity parameter $H = 1 - \beta/2, (0 < H < 1)$, if it satisfies

$$X = M^{1-H} X^{(m)}$$

Exactly second-order self-similar means that the averaged processes $X^{(m)}$ have identical correlation structure with $X$.

A discrete time stochastic process $X_k$ is called asymptotically second-order self-similar with self-similarity parameter $H = 1 - \beta/2, (0 < H < 1)$ if, for all $k$ large enough and it satisfies

$$r^{(m)}(k) = r(k), \text{ for all } m = 1, 2, \ldots (k \to \alpha)$$

Asymptotically second-order self-similar means that the averaged processes $X^{(m)}$ have a nondegenerate correlation structure, indistinguishable from $X$ as $m \to \alpha$.

In teletraffic modeling the discrete-time stochastic process $X_k$ represents the number of traffic entity (packets, connections/calls) arrivals at time $k$.

### 5.3.3.1 Continuous-time self-similar process:

Self-similar process is a stochastic process, which has scale invariability in statistical. If a continue time stochastic process $X(t)(t=0,1,\ldots)$ has

$$X(t) \overset{D}{=} a^{-H} X(at), \quad \alpha > 0, t \geq 0, \quad 0 < H < 1$$

It is self-similar in statistical, where $D$ means in distribution. If the equation comes to existence for infinite dimensional distributions, $X(t)$ is exactly self-similar, or if equation comes to existence only for mean and variance of the stochastic process, it is asymptotically self-similar [49].

In teletraffic modeling the continuous-time stochastic process $X(t)$ represents the number of traffic entity (packets, connections/calls) arrivals in the time interval.

Self-similar has a number of equivalent effects:

- Variances that decay slowly. In other words, the variance of the sample mean
decreases more slowly than the more usual reciprocal of the sample size.

- Hyperbolic decay of autocorrelation rather than exponential decay:
  \[ r(k) \sim k^{-\beta} \text{ if } k \to \infty \]

- The power spectral density \( S(f) \) that behaves like that of \( 1/f \) noise around the origin:
  \[ \lim_{f \to 0} S(f) \approx f^{-(1-\beta)} \text{ if } f \to 0. \]

5.4 Video Traffic Modeling:

5.4.1 Markov based Model:

A. A. Markov and A. Kolmogorov laid down the foundation to the theory of Markov processes in the early 20th century. A discrete-state stochastic process \( \{X(t)\}_{t \geq 0} \) is called a Markov chain if for \( t_0 < t_1 < \ldots < t_n \), the state transition probability function satisfies the following Markov property:

\[
P\{X(t) = x|X(t_0) = x_0, X(t_{n-1}) = x_{n-1}, \ldots, X(t_0) = x_0\} = P\{X(t) = x|X(t_n) = x_n\}
\]

That is, the current state summarizes all relevant information about past states. Unlike renewal traffic models, Markov traffic models introduce dependence into the random sequence of inter-arrival times. Consequently, they can potentially capture traffic burstiness, due to non-zero autocorrelations in the inter-arrival time process. It may interpret transitions of the Markov chain as signaling arrivals.

5.4.1.1 Discrete time Markov chains:

A discrete-time Markov chain (DTMC) is characterized by state changes that can only take place at discrete times. The DTMC has geometrically distributed state sojourn times. A DTMC can be characterized by the properties of its states. Let \( f_{ij}^{n} \) denote the probability that the first return to state \( E_j \) occurs at the nth step. The probability of at
least one return to state $E_j$ is given by

$$f_{jj} = \sum_{n=1}^{\infty} f_{jj}^n$$  \hspace{1cm} (5.3)

Thus the system is certain to return to $E_j$ if $f_{jj} = 1$. In this case $\mu_{jj}$ defines the mean return (recurrence) time:

$$\mu_{jj} = \sum_{n=1}^{\infty} nf_{jj}^n$$  \hspace{1cm} (5.4)

The states of a discrete-time Markov chain can be classified based on the definition of the first return times as follows:

- A state is transient if $f_{jj} < 1$,
- A state is recurrent (persistent) if $f_{jj} = 1$,
- A recurrent state is null if $\mu_{jj} = \infty$ and nonnull if $\mu_{jj} < \infty$,
- A state is periodic with period $t$ if a return is possible only in $t, 2t \ldots$ steps,
- A recurrent state is ergodic if it is nonnull and aperiodic.

An irreducible DTMC is one in which all states are reachable from all other states. The states of an irreducible DTMC are either all transient or all recurrent nonnull or all recurrent null. An ergodic DTMC is irreducible, recurrent nonnull, and aperiodic. Most of the systems are interested for modeled by ergodic DTMC because this corresponds to well-defined steady state behavior. If the transition probabilities are independent of the time index $n$, the DTMC is homogeneous.

In a DTMC, a set $C$ of states is said to be closed if the system, once it is in one of the states of $C$, will remain there indefinitely. A special example of a close set is a single state $E_j$ with transition probability $P_{jj} = 1$. In this case $E_j$ is called absorbing state.

**5.4.1.2 Continuous time Markov chains:**

A Continuous-time Markov chain (CTMC) is characterized by state changes that can occur at arbitrary points in time. A CTMC can be completely described by an initial state probability vector for $X(t_0)$ and transition probability functions. The CTMC has
exponentially distributed state sojourn times. A CTMC is said to be irreducible if every state can be reached from every other state, with non-zero probability. A state is said to be absorbing if no other state can be reached from it with nonzero probability. The notions of transient, recurrent nonnull, recurrent null carry over from DTMCs. There is no notion of periodicity for CTMCs, however. An irreducible CTMC is ergodic. If the CTMC is ergodic, the equilibrium probability distribution \( \{\pi_i\}_{i \in S} \) can be obtained from a set of linear equations subject to the probability conservation constraint:

\[
\{ \sum_{j \in S} \pi_j q_{ij} = 0 \} 
\]

(5.5)

Where \( \pi = (\pi_1, \pi_2, \ldots, \pi_N) \) and \( Q \) denotes the infinitesimal generator matrix \( Q = (q_{ij}) \) and \( q_{ij} \) denotes the transition rate between states \( i \) and \( j \). The diagonal elements \( q_{ii} \) are determined such that the sum of all elements in each row equals zero.

### 5.4.4 Definition of autoregressive Process:

Consider a linear system with input \( e(n) \) and output \( x(n) \), where \( n \) is the discrete time. The finite AR process is generally expressed as:

\[
x(n) = \sum_{k=1}^{p} a_k x(n-k) + e(n)
\]

(5.6)

Where \( e(n) \) is an uncorrelated process with zero mean and variance \( \sigma^2 \), and is a finite sequence with \( a_p \neq 0 \). Such a process is denoted by AR(p) and \( p \) is called the order of the AR process. The sequence \( \{e(n)\} \) consists of i.i.d random variables, known as the residual (or error process), that gives the AR model its stochastic nature. The residuals are often normally distributed, which implies that \( x(n) \) is also normally distributed, but with different mean and variance. There are a number of methods to estimate the parameters for an AR process given \( x(n) \) [23]. In the present study, the linear prediction method will be described next. Given a set of past samples of the signal \( x(n) \), a linear approximation of the present value of the signal is defined as:

\[
\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k)
\]

(5.7)
By subtracting $\hat{x}(n)$ from the current signal value $x(n)$, obtain the residual signal $e(n)$; that is,

$$e(n) = x(n) - \hat{x}(n)$$

$$= x(n) - \sum_{k=1}^{p} a_k x(n-k)$$

(5.8)

$$\hat{x} = [X(n) - X(n-1) \ldots - X(n-p)]^T$$

(5.9)

$$\tilde{a} = [a_0 - a_1 - a_2 \ldots \ldots - a_p]^T$$

Where $a_0 = 1$. Then Eq. (5.8) can be written in the following form:

$$e(n) = \tilde{X}^T \tilde{a}$$

(5.10)

The most commonly used method to optimize model coefficients $\{a_k\}$ is to minimize the mean-square value $e$ of the error sequence $e(n)$. Based on (5.10), we can write:

$$e = E\{e^2(n)\}$$

(5.11)

$$= \tilde{a}^T \tilde{R} \tilde{a}$$

Where $E\{ . \}$ is the expectation operation, and $\tilde{R}$ is the $(p+1) \times (p+1)$ correlation matrix of the input vector $\tilde{X}$. The prediction model vector $\tilde{a}$ is therefore chosen so as to minimize $e$ subject to the constraint that the first element of $\tilde{a}$ equals 1.

$$\tilde{a} = \frac{\tilde{R}^{-1} \tilde{a}}{\delta^T \tilde{R}^{-1} \delta}$$

(5.12)

Where $\delta$ is an $(p+1) \times 1$ vector of the form

$$\delta = [1 \ 0 \ 0 \ 0 \ \ldots \ 0]^T$$

(5.13)

For a stationary process with mean $\mu$ and variance $\sigma^2$, the autocorrelation function (ACF) of $x(n)$ is defined as

$$\rho_k = \frac{E[(X(n) - \mu)(X(n+k) - \mu)]}{\sigma^2}$$

(5.14)

The ACF of an AR(p) can be written as a difference equation:

$$\rho_k = \sum_{r=1}^{p} a_r \rho_{k-r}$$

(5.15)

Where $\rho_k$ is the ACF at lag k.
5.4.5 Markov Modulated Fluid Models:

Fluid models characterize the traffic as a continuous stream with a parameterized flow rate (such as bits/sec). These models are appropriate in the case where individual units of traffic (packets or cells) have little impact on the performance of the network.

Fluid models are conceptually simple and their simulation has an important advantage over other models. Consider for example, an event simulation for an ATM multiplexer. All models that distinguish between cells and consider the arrival of each cell as a separate event, consume vast amount of memory and CPU resources. In contrast, fluid models characterize the incoming cells by a flow rate. An event is only triggered when the flow rate changes. Since flow rate changes happen much less frequently than cell arrivals, considerable savings in computing and memory resources are achieved. In a queueing context, it is easy to manipulate fluid buffers. Furthermore, the waiting time concept simply becomes the time it takes to serve (clear) the current buffer, and loss probabilities (at a finite buffer) can be calculated in terms of overflow volumes. Because fluid models assume a deterministic service rate, these statistics can be readily computed. Typically, though, larger traffic units (such as coded frames) are of greater interest than individual cells. Modeling the larger units as discrete traffic and their transport as fluid flow would give us the best of both worlds: Loss probability and waiting times can be measured and enjoy savings on simulation computing resources. [30, 10]

A fluid model that is typically used to model traffic is the Markov modulated fluid model.

Typical fluid models assume that sources are bursty - of the “on-off” type. While in the “off” state, traffic is switched off, whereas in the “on” state traffic arrives deterministically at a constant rate $L$. For analytical tractability, the duration of “on” and “off” periods is assumed to be exponentially distributed and mutually independent. In this model, the current state of the underlying Markov chain determines the flow (traffic) rate. While in state $S_k$, traffic arrives at a constant rate $\lambda_k$. 
This model is a Markov modulated constant rate model and is used to model VBR video sources. [30], [10].
CHAPTER-6

6. Packet scheduling in IP routers:

Packet scheduling is an essential traffic control because it recognizes that packets have different forwarding requirements. A general principle of packet scheduling is that the average waiting time for all packets does not change under different schedules. Packet scheduling is one kind of multiplexing where a scheduler decides how a number of packet sources (usually queues) will share a single output channel. A packet scheduler is naturally implemented as a pull element with multiple inputs and one output. This element reacts to requests for packets by choosing one of its inputs, pulling a packet from it, and returning that packet (If the chosen input has no packets ready, the scheduler will usually try other inputs) [40].

In packet switched networks, scheduling is done in the network’s intermediate nodes like routers and switches. Each intermediate node consists of the following components: Input buffers, output buffers, and switching fabrics [34]. Fig. 6.1 shows the architecture of a network switch.

![Diagram of a network switch](image)

Figure 6.1: General architecture of a network switch [34].

Scheduling algorithms are located on the output interface of the network switches. Each interface has its own instance of the scheduler where different scheduling algorithms can be used on different interfaces. Fig 6.1 shows the architecture of a scheduler where each scheduler consists of two main components: classifier and
scheduler. The classifier is in charge of allocating packets into different queues according to the scheduling classifier scheme. The scheduler selects the next best packet from the queues according to its appropriate scheduling algorithm.

Figure 6.2: The model of a scheduling algorithm [34]

For designing scheduling schemes, different trade-offs can be considered in terms of the following five requirements which are Complexity, Fairness, Isolation (protection), Efficiency and Performance. Depending on the specific situation, some of these requirements may be more important than others and the decision for the best choice is made given the particular situation. The next generation of Internet needs scheduling disciplines in order to support:

- Per-connection delay, bandwidth, and loss bounds needed for guaranteed-service applications.
- Fair resource allocation needed for best-effort applications.

Packet scheduling mechanisms are classified into two categories: work conserving and non-work-conserving. A work-conserving scheduler is never idle if there is at least one backlogged queue in the system. It is an important feature to utilize the system resources. In a non-work-conserving scheduler, each packet is assigned a time when it has to be sent to the output interface. The scheduler remains idle and no packet will be transmitted until the next packet is eligible for transmission.
Some well-known scheduling disciplines have been discussed such as First in First out (FIFO), Priority Queuing (PQ), Weighted Fair Queuing (WFQ), Generalized Processor Sharing (GPS), Low-Latency Queuing (LLQ) and Weighted Round-Robin (WRR).

6.1 First in First out Queuing (FIFO):

First in First out (FIFO) is one of the simplest scheduling algorithms. In FIFO queuing, all packets are treated equally by placing them into a single queue, and then servicing them in the same order that they were placed into the queue. This Scheduling method processes packets according to their arrival order to the switching node. The packets are placed in queue order by the multiplexer according to arrival time as shown below.

FIFO is a work conserving algorithm and has a very low complexity, so it is one of the most commonly implemented algorithms in the networks. There are some limitations for FIFO as follows [34].

- It is not able to provide fairness in resource allocation to different flows. Nevertheless, this limitation is not very important for best effort applications.
- It cannot provide any performance guarantees in terms of delay, delay jitter, or throughput to the real time applications.

Consequently, multimedia applications do not work well with FIFO schedulers. One way to provide a delay bound is to limit the buffer size, so that the packets are
guaranteed to be sent in less than the time it takes to serve a full queue. A disadvantage of this solution is that it increases the packet loss probability, which is a consequence of the high buffer overflow probability.

6.2 Priority Queuing (PQ):

Priority Queuing (PQ) is providing differential treatment to flows is to use multiple with associated priorities. It is a powerful and strict form of congestion management. PQ allows the network administrator to define up to four queues for network traffic. These queues are the High, Medium, Normal, and Low priority queues. The router processes the queues strictly based on their priority. If there are packets in the high priority queue, this queue will be processed until it is empty, independently of the state of the other queues. Once the high priority queue is empty, the router moves to the medium queue and dispatches a single packet. Immediately the router checks the high queue to ensure it is still empty. If it is, it will go to the medium queue, then the normal, then the low. All three, high, medium, and normal, must be completely empty before a single packet is dispatched out of the low queue. Every time a packet is dispatched the router checks the high queue.

Priority queuing gives network administrator’s tremendous control over network traffic. And it also gives the network administrator enough power to deny low priority traffic the chance to be transmitted at all. The admission control is necessary to maintain desired QoS to the offered services. It is especially important in the case of real-time services like call blocking, either new call.

6.3 Weighted Fair Queuing (WFQ):

Weighted Fair Queuing (WFQ) is a complex scheduler used for various size packets. WFQ offers fair queuing that divides the available bandwidth across queues of traffic based on weights. Each flow or aggregate there of is associated with an independent queue assigned with a weight, so as to ensure that important traffic gets higher priority over less important traffic. In times of congestion the traffic in each queue (a single flow or an aggregate of them) is protected and treated fairly, according to its
weight. Each queue shares the transmission service proportionally to the associated weight.

WFQ has the following important goals and objectives:

- Divide traffic into flows.
- Provide fair bandwidth allocation to the active flows.
- Provide faster scheduling to low-volume interactive flows.
- Provide more bandwidth to the higher-priority flows.

Since the WFQ is a version of GPS, it guarantees to obtain a certain fraction of system capacity for both short and long-term scale. As it works with the same principles of GPS, WFQ is work-conserving discipline. The packet delay variability in WFQ is lower than FIFO and PQ, because it isolates the source classes which provide at least capacity for each class.

6.4 Weighted Round Robin (WRR):

Weighted Round Robin (WRR) is a simple modification to round robin. Also it is a simple emulation of GPS. WRR serves a certain amount of data instead of sending an infinitesimal amount of data from the queues [26]. The served data can be in the form of packets or bytes. In this algorithm each queue has a weight that allows sending a certain amount of data from each nonempty queue. The weight is usually a percentage of the whole bandwidth. This algorithm is a closer approximation for GPS when all connection flows have equal weights and all the packets have the same size (in case of packet WRR) [26]. When different traffic flows have different weights, the WRR algorithm serves the flows in proportion to their weights. In cases where there are different packet sizes for different flows in order to achieve a normalized set of weights for the flows, the WRR algorithm divides each flow’s weight by the average packet size of that flow. One of the advantages of WRR is its simplicity. A WRR scheduler can be efficiently implemented in hardware. Also, once the weights are selected the runtime overhead of WRR is small.
There exist two problems that cause WRR not to emulate GPS correctly [26]:

- In practice the source’s packet sizes may not be predictable, so a WRR algorithm cannot allocate bandwidth fairly to different flows.
- At time scales shorter than a round trip time, the algorithm is not a fair algorithm since some flows may get more service than the others. WRR tends to be fair only at larger time scales.

A limitation of the WRR algorithm is that its performance depends on the packet arrival pattern. For example, when a packet arrives to a queue just after the queue has been served. It has to wait in the queue for a whole round time before getting served, no matter how important the packet’s flow is.

### 6.5 Generalized Processor Sharing (GPS):

Generalized Processor Sharing (GPS) is an ideal work-conserving scheme that is capable of achieving max-min fair share and also the best approximation to the Fair Fluid Model. In simple term, GPS assumes that each flow is kept in a separate logical queue. It serves an infinitesimal amount of data from each queue such that in a finite time interval it visits every nonempty queue. Each queue can have associated weight and can be served in proportion of its weight.

Following are the variables used in a GPS algorithm:

- $\phi_i$: The share of bandwidth reserved by flow $i$.
- $W_i(t_1, t_2)$: amount of traffic served from flow $i$ during the time period $(t_1, t_2)$.

A connection flow is defined as backlogged if it has packets that are either receiving service or waiting for service in its queue [26]. Thus, GPS serves each backlogged connection with minimum rate equal to its reserved rate at each instant. The extra bandwidth not used from other connection flows is distributed among all the backlogged connections in proportion to their reservation. In other words, during an arbitrary interval $(t_1, t_2)$, for any pair of backlogged connection flows $i$ and $j$ the
following equation holds [26]:

\[ \frac{W_i(t_1, t_2)}{W_j(t_1, t_2)} = \frac{\phi_i}{\phi_j} \]

Or

\[ \frac{W_i(t_1, t_2)}{\phi_i} = \text{Constant}. \]

Because GPS posse the properties of ideal fairness and complete isolation. However, GPS is not implementable because serving an infinitesimal amount of data from each non-empty queue is not possible. Thus, various emulations of GPS have been proposed in the literature.

6.6 Low Latency Queuing (LLQ):

Low-Latency Queuing (LLQ) is an extension of CBWFQ. In fact, the only real difference between the two is how the bandwidth is allocated to the class maps in the policy map. LLQ can instruct one or more class-maps to direct traffic into a priority queue. LLQ includes a strict-priority queue that is given priority over other queues, which makes it ideal for delay and jitter-sensitive real time applications. Unlike the plain old PQ, whereby the higher-priority queues might not give a chance to the lower-priority queues and effectively starve them, the LLQ strict-priority queue is policed. This means that the LLQ strict-priority queue is a priority queue with a minimum bandwidth guarantee, but at the time of congestion, it cannot transmit more data than its bandwidth permits. If more traffic arrives than the strict-priority queue can transmit (due to its strict bandwidth limit), it is dropped. Hence, at times of congestion, other queues do not starve, and get their share of the interface bandwidth to transmit their traffic.
7. Video source models:

Due to analyze data flow which requires a very high bandwidth, video traffic has been chosen for this work. Video files which are encoded by MPEG-4 are used here because MPEG-4 encoded video is expected to account for a large portion of the traffic in future wire line. The frame size traces have been generated from MPEG-4 encoding of over 10 video sequences of 60 minutes length each. MPEG-4 provides very efficient video coding covering the range from the very low bit rate of wired communication to bit rates and quality levels beyond high definition television (HDTV).

For each encoded video and quality level, verbose trace files have been taken from http://www.tkn.tu-berlin.de/research/trace/ltvt.html web site. The verbose trace files give Frame number, Frame Type (I, P or B), Display time (in ms), and Frame size (in bytes) in ASCII format with one frame per line. Point to be noted here that in the verbose files the frames are ordered in the sequence IPBBPBBPBBBIBBP...... This is because for decoding a B frame the decoder needs both the preceding I (or P) frame and the succeeding P (or I) frame. The frames are therefore emitted in the sequence IPBB... by the encoder. The Group of Picture (GoP) pattern was set to IBBBPBBPBB. Thus twelve frames are considered to form a GoP. Medium and high quality levels of encoded video are used which consists of more pixels than low resolution videos, come of much help to test the network. It is assumed that video sources are generating 25 frames/sec.

Maglaris at el.[5] have developed a simulator, which calculates GoP size and analyze each frame individually. After melting down frames in bit in the network, packets are started counting and pass it to the buffer. Appropriate queueing method is then applied to transmit packets to the receiver. The focus of this work is on the transmission channel through which all packets are transmitted. Applications of digital video traffic such as video conferencing, video telephone, and switched TV,
impose very large bandwidth requirements on public and private networks. The installation of optical fibers down to the subscriber local loop will encourage the proliferation of video services, but as more bandwidth becomes available new services will tend to overload them. Applications that require timely and synchronous delivery, such as telephone voice and real-time video, currently use dedicated circuit switching. This may change with the new concept of fast packet switching [39] that promises to convey large volumes of packetized digital voice. The asynchronous packetized transmission of conversational voice requires very stringent end-to-end delay; voice packets exceeding a given time threshold may not be used to synthesize the continuous voice stream at the receiver. Late packets thus result in voice losses that can be tolerated up to a certain degree. A similar problem must be considered in full motion video and particular attention must be paid to recovery from packet losses.

Data are generated in the course of this work using a modified version of the conditional replenishment scheme proposed in [37]. Two Markov models that match their basic statistics have been introduced. The first model (Model A) is a continuous state autoregressive discrete-time Markov process which is used for our simulation experiments. The second model (Model B) is a quantized-state continuous-time Markov process which is used in our fluid-flow queueing analysis. The analytic model is used to analyze the queueing behavior of statistical multiplexing of several independent, identically distributed video sources. The results show that the probability of buffering or delaying data beyond a certain threshold decreases dramatically as the number of multiplexed video sources increases.

Variable bit rate video coding exhibits the type of statistical variations that made statistical multiplexing and packet switching attractive for bursty computer communications. These network architectures can dynamically support variable rate sources by smoothing out the aggregate of several independent streams in common buffers within the network. It is expected that such technologies may provide efficient video transport without the unpleasant variations in quality of multimode coders. In statistical or asynchronous time division (ATD) multiplexing, several independent sources share a line of capacity less than the sum of their peak rates. Instead of using individual buffers, all sources feed a common buffer and their cumulative bit rate
tends to smooth out around the average rate as indicated by the law of large numbers. Packet switching is a network extension of statistical multiplexing, with data from individual sources segmented into small packets that are stored and forwarded from switch to switch toward their destination. The statistical smoothing of variable rate sources is achieved as packets are buffered in the network switches. Both statistical multiplexing and packet switching introduce variable delays in delivering data due to the buffering stages.

Both statistical multiplexing and packet switching introduce variable delays in delivering data due to the buffering stages. In addition, they may introduce data losses due to buffer overflow. Thus, it is very important to select parameters and line speeds to minimize these effects while maintaining the efficiency of statistical averaging.

7.1 The modeling process:

Traffic models reflect the best knowledge of traffic behavior. Traffic is easier to characterize at sources than within the network because flows of traffic mix together randomly within the network. When flows contend for limited bandwidth and buffer space, their interactions can be complex to model. The “shape” of a traffic flow can change unpredictably as the flow progresses along its route. On the other hand, source traffic depends only on the rate of data generated by a host independent of other sources. Knowledge is gained primarily from traces (measurements) of past traffic consisting of packet arrival times, packet header fields, and packet sizes. A trace of Star Wars IV encoded by MPEG-4 at medium quality is shown in Figure 7.1 (frame size). Traffic can be measured by packet sniffers or protocol analyzers.
One of the practical difficulties in traffic modeling is collecting and analyzing large sets of traffic measurements. Traffic can be highly variable, even between two similar types of sources. For example, one video might have many scene changes reflected by frequent spikes in the source rate, while another video might have few scene changes resulting in a smoother source rate. Therefore, it is good practice to collect measurements from many sources or over many time periods, and then look for common aspects in their behavior. In probabilistic terms, each traffic trace is a single realization or sample of the traffic behavior.

It is assumed that video sources are generating 25 frames/s. Each frame consists of approximately 250000 pixels that are digitally coded. The conditional replenishment compression algorithm encodes and transmits the difference between pixel levels of subsequent frames if this difference exceeds a given threshold. As in [36] it is assumed that N independent video sources are multiplexed into a high-speed trunk. The un-buffered coded bits from each source are first stored in separate pre buffers and then join a common buffer. The pre buffers may perform a pre smoothing of the source data over a frame period as will be explained later. The multiplexer assembles the data in the common buffer into blocks that are transmitted over the high-speed communication line. The blocks may be asynchronous time frames that combine portions of data from each source. The frame length depends on the instantaneous amount of data from the sources. Frame delimiters and source identification...
information must be included to enable the de-multiplexing at the destination.

![GOP interms of Bit Rate for Source(Star Wars IV) vs Number of GOPs](image)

Figure 7.2: Number of GOP in terms of bit rate for source movie (Star Wars IV).

A block may also be a packet of data assembled from a single source. Packets are stored and forwarded in a FIFO mode in the same order as they are assembled. The packet length may be variable due to temporal changes in the source bit rate, or fixed, in which case the variable source rates will cause varying packet arrival rates. Each packet needs a header identifying the source, the destination, its sequence number, and possibly a time stamp to alert the network in case of excessive buffering delays. Error detection and correction overhead bits may be added. Statistical multiplexer will be modeled as a queue that receives the encoded bit streams from the pre-buffers; its service rate is determined by the speed of the channel. Issues related to the specific multiplexing technique, e.g., frame or packet formats and related protocols, are beyond the scope of this study. This analysis is based on a continuous fluid-flow approximation of the traffic that does not take into account the discrete nature of packets or frames. The queue behaves like a reservoir of water that is fed from water supplies of time varying rates and empties through a fixed-rate sink. The fluid flow approximation is a powerful analytic tool which is able to capture correlated input models; it is accurate if the packet size is small compared to the speed of service [38]. The analysis agrees with our simulation model, in which bit streams from individual sources are assembled into packets of fixed length. Packets join the common buffer and are served in a FIFO order. An accepted scheme is the fluid-flow model with ON/OFF sources. In this model, traffic is considered as a continuous flow and the
packet concept disappears. Therefore, the aggregated traffic is a consequence of the amount of fluid generated by a set of independent ON/OFF sources. The main difficulty in accurately modeling real traffic by employing a fluid-flow model with ON/OFF sources is the selection of the appropriate collection of ON/OFF traffic sources.

7.2 Source Model Inferences from Experimental Results:

For a smooth transition of asynchronous discrete time division multiplexed data packets a model is used in order to analyze the common buffer statistics, a model is needed for the coded source rate. The rate depends on the compression algorithm and the nature of the video scene. For a scene without abrupt movement, such as the head of a person in a picture phone, a bell-shaped stationary probability density have been expected and to exhibit significant correlations for an interval of several frames. It has been verified this behavior by experimenting with a test image sequence depicting the head of a talking person. The sequence had duration of 10 s or 250 frames. The instantaneous bit rate $\lambda(t)$ have been measured in bit/pixel. Recall that there are about 250000 pixels per frame and 25 frames/s, thus 1 bit/pixel corresponds to 6.25 Mbits/s.
Figure 7.3: Coding Bit rate of five different movies

- Bit Rate for source (Silence of the Lambs)
- Bit Rate for source (Soccer)
- Bit Rate for source (Mr. Bean)
- Bit Rate for source (Star Wars IV)
- Bit Rate for source (Jurassic Park I)
A key parameter in evaluating the bit rate is the number of pixels, $N_p$, which, combined in a group, give rise to a coded bit stream of length $N_b$ into the prebuffer. The average bit rate is then equal to $N_b/N_p$ bit/pixel. A large $N_p$ leads to more averaging at the prebuffering stage. It requires large individual prebuffers and introduces long delays. A small $N_p$ preserves the variations of individual source rates at the input of the multiplexer. Statistical multiplexing in the common buffer will then average the overall bit rate in a more efficient manner. On the other hand, too small $N_p$ values will introduce quasiperiodic variations with the period equal to frame duration. For example, if $N_p$ is the number of pixels in a horizontal scan line, $N_b$ will be influenced by the variable activity of the lines and will tend to be repeated after a frame for the same line. These periodic variations are not worth exploiting in statistical multiplexing, and disappear with prebuffering of a whole frame. The prebuffers will introduce a one-frame delay of $1/25$ s at the encoder side. Therefore, $N_p$ have chosen which equal to the number of pixels in a frame. This choice also makes $\lambda(t)$ dependent only on the varying activity of the frame sequence, and eliminates complicated properties from its statistics. Note that $N_p$ is not necessarily related to the packet size, as there may be several packets formed from an $N_b$ bit stream, and the packetization mechanism need not be synchronized with the frame sequence. Fig. 7.3, shown $\lambda(t)$ as the bit rate (averaged for each frame) for all 200 frames of the test sequence measured in bit/pixel. Although the measured rate is fixed for the duration of a frame ($1/25$ s), it is treated as a continuous-time function since the frame period is very small compared to the standard time scale. The average value $\mu$ over all 7499 GOP frames and the standard deviation $\sigma$ were found to be $E(\lambda) = 0.149$ bit/pixel. The maximum value of the bit rate was 0.012 bit/pixel. Note that the rate cannot be negative, and that the density function is not exactly symmetric around its average $E(\lambda)$.

The autocovariance $C(\tau) = E[\lambda(t)\lambda(t+\tau)] - \mu^2$ of the sequence was evaluated and is
shown in Fig. 7.4 as a function of frame (n) difference. The desire curve corresponds to an exponential fit of the form \( \hat{C}(\tau) = \sigma^2 e^{-a\tau} \), with \( a = 0.03 \text{s}^{-1} \). It is partly due to truncations in computing \( C(\tau) \) and would be eliminated if the sequence were much longer. Interestingly, it may partly be due to the fact that the head of the talking person in the scene was moving in a roughly periodic pattern.

![Figure 7.4: ACF Vs Lags for GOP](Image)

We expect the average rate and its variance to be quite different in another test sequence, depending on the amount of motion.

We will present in the sequel two models of the encoded bit rate of a video source. The first model is suitable for queueing simulations, while the second leads to a simple queueing analysis. In both cases, the first- and second-order statistical properties of the measured data have been match as well as some features of the steady-state distribution. In particular, models have the same mean and variance as the experimental data. The queueing behavior at the multiplexer is not very sensitive to the specific nature of the distribution as results will show.

A key feature of the models is that they match the exponential autocovariance fit
mentioned above. Apart from experimental evidence in Fig. 7.4, the exponential nature of the autocovariance is also supported from early results, obtained for video scenes.

7.3 Source Model A: Continuous-State Autoregressive Markov Model:

We first model the coder rate as a continuous-state, discrete time stochastic process. Let $\lambda(n)$ represent the bit rate of a single source during the $n^{th}$ frame. A first-order autoregressive Markov process $\lambda(n)$ is generated by the recursive relation

$$\lambda(n) = a\lambda(n-1) + bw(n)$$  \hspace{1cm} (7.1)

Where $w(n)$ is a sequence of independent Gaussian random variables and $a$ and $b$ are constants. Assume that $w(n)$ has mean $\eta$ and variance 1. Further, assume that $|a| < 1$; thus, the process achieves steady state with large $n$. The steady-state average $E(\lambda)$ and discrete autocovariance $C(n)$ are given by [42]

$$E(\lambda) = \frac{b}{(1-a)} \eta$$  \hspace{1cm} (7.2)

$$C(n) = \frac{b^2}{1-a^2} a^n \quad n \geq 0$$  \hspace{1cm} (7.3)

The autocovariance is exponential and can fit the experimental data. The steady-state distribution of $\lambda$ is Gaussian with mean $E(\lambda)$ and variance $C(0)$. Assuming that the negative tail of the density of $\lambda$ is very small, there is a reasonable matching of experimental data and the autoregressive model. From the measured data it is obtained

$$E(\lambda) = 0.149 \text{ bit/pixel}$$ 

$$C(n) \simeq 0.75 \times (e^{-0.03})^n \text{ (bits / pixel)}^2$$

The discrete autocovariance $C(n)$ is obtained from the experimental fit $C(\tau) = 0.75 \times e^{-0.03}$, by sampling at $n/\tau = 25$ frames/s. Matching (7.2) and (7.3) with the measured data, have that $a \simeq 0.9997$ $b \simeq 0.0198$  \hspace{1cm} (7.4)
The continuous-state autoregressive model provides a rather accurate approximation of the bit rate. A better matching may be achieved if increases its order to include the influence on $\lambda(n)$ of several past values $\lambda(n-k)$, $k \geq 1$. In this case, the autocovariance will be the sum of several exponentials. Nevertheless, since a single exponential fit was found to dominate the decay of the autocovariance, here the first-order model is trusted.

Table 7.1: Characteristics Coefficient values for different video traces.

<table>
<thead>
<tr>
<th>Video Trace File</th>
<th>Simulation</th>
<th>C(0)</th>
<th>Coefficient of Existing Curve</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>Star Wars IV</td>
<td>GOP</td>
<td>0.0061</td>
<td>0.7594 -0.03051</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>2.06E+05</td>
<td>0.1895 -0.00575</td>
</tr>
<tr>
<td>Jurassic Park I</td>
<td>GOP</td>
<td>0.0865</td>
<td>0.7655 -0.02491</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>1.2732e+006</td>
<td>0.4597 -0.005</td>
</tr>
<tr>
<td>Soccer</td>
<td>GOP</td>
<td>0.4921</td>
<td>1.006 -0.1892</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>5.24E+06</td>
<td>0.6821 -0.01733</td>
</tr>
<tr>
<td>Mr. Bean</td>
<td>GOP</td>
<td>0.0315</td>
<td>0.5889 -0.01642</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>8.01E+05</td>
<td>0.2488 -0.00573</td>
</tr>
<tr>
<td>S_Lambs</td>
<td>GOP</td>
<td>0.1105</td>
<td>0.6791 -0.00728</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>1.12E+06</td>
<td>0.6643 -0.00315</td>
</tr>
</tbody>
</table>
Table 7.2: Model parameter values for different video traces.

<table>
<thead>
<tr>
<th>Video Trace File</th>
<th>Simulation</th>
<th>Steady state average $E(\lambda)$ for all frames</th>
<th>Steady state average $E(\lambda)$ for all GOP</th>
<th>Model Parameter (a)</th>
<th>Model Parameter (b)</th>
<th>Slope of autocorelation (To the base 10)</th>
<th>Slope of autocorelation (To the base $e$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars IV</td>
<td>GOP</td>
<td>0.273050</td>
<td>0.149227</td>
<td>0.999709</td>
<td>0.019884</td>
<td>0.000126</td>
<td>0.000291</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>0.273050</td>
<td>0.149227</td>
<td>0.999874</td>
<td>0.012938</td>
<td>0.000055</td>
<td>0.000126</td>
</tr>
<tr>
<td>Jurassic Park I</td>
<td>GOP</td>
<td>0.940571</td>
<td>0.514049</td>
<td>0.999004</td>
<td>0.039038</td>
<td>0.000433</td>
<td>0.000996</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>0.940571</td>
<td>0.514049</td>
<td>0.999800</td>
<td>0.013599</td>
<td>0.000087</td>
<td>0.000200</td>
</tr>
<tr>
<td>Soccer</td>
<td>GOP</td>
<td>3.887757</td>
<td>2.124762</td>
<td>0.992461</td>
<td>0.122932</td>
<td>0.003287</td>
<td>0.007568</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>3.887757</td>
<td>2.124762</td>
<td>0.999931</td>
<td>0.009724</td>
<td>0.000030</td>
<td>0.000069</td>
</tr>
<tr>
<td>Mr. Bean</td>
<td>GOP</td>
<td>0.646191</td>
<td>0.353117</td>
<td>0.999343</td>
<td>0.027804</td>
<td>0.000285</td>
<td>0.000657</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>0.646191</td>
<td>0.353117</td>
<td>0.999771</td>
<td>0.010678</td>
<td>0.000100</td>
<td>0.000229</td>
</tr>
<tr>
<td>S_Lambs</td>
<td>GOP</td>
<td>0.615105</td>
<td>0.336173</td>
<td>0.999709</td>
<td>0.019884</td>
<td>0.000126</td>
<td>0.000291</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>0.615105</td>
<td>0.336173</td>
<td>0.999874</td>
<td>0.012938</td>
<td>0.000055</td>
<td>0.000126</td>
</tr>
</tbody>
</table>

7.4 Queueing Simulation Using Model A:

The model in (7.1) with the parameters (7.4) was used to generate the bit rates of each source in queueing simulation experiments. Instances of negative $\lambda(n)$ were corrected to zero. This occurred rarely enough and did not significantly alter the measured moments of the simulated rates. In the simulation model, N sources generate independent bit streams with rate $\lambda(n)$ for the duration of their nth frame. It’s assumed that the sources need not be synchronized in their frame sequences. Thus, the first frame Occurrence of the N sources is randomized over the interval of a frame. Once initialized, the sources keep their individual frame synchronizations and their rate is generated according to the autoregressive formula (7.1). Bits generated from a single source over a frame period join a prebuffer. At the end of the frame, the bits in the prebuffer are packetized into fixed length packets that proceed into the common queue. Packets in the common buffer are served in a FIFO order. Packet transmission times are fixed, proportional to the packet length.
We collect statistics (as time averages) of the common buffer length to estimate the average and variance of the queue length, and the probability that the queue exceeds certain levels. Results are presented in next section.

**7.5 Source Model B:**

**7.5.1 Discrete-State, Continuous-Time Markov Process:**

Most researchers try to describe the autocorrelation function (ACF) of a video trace (be it on a frame or GOP level) and take this as the starting point towards building their own video source model. The ACF of an MPEG encoded video has an inherently strong periodic behavior, related to the periodic alternation of the three frame types. Most often these periodic peaks are smoothened by using a rate shaper embedded in the encoder. There is longer term correlation as well, due to the alternation of easy and difficult scenes. Therefore, modeling the source behavior at a higher granularity, e.g., the GOP level is justified. Model A is easy to simulate but cannot lead to a queuing analysis of manageable complexity. Even a continuous flow approximation of the queuing process with an input leads to two-dimensional diffusion partial differential equations, with reflecting barriers at zero for both the input rate and the queue size. Such problems are encountered in stochastic storage theory; the case of queues with continuous correlated inputs is of particular analytic complexity [41].

We developed instead, a discrete-state Markov model that allows simple analytic tools to be employed. In Model B the bit rate is quantized into finite discrete levels. Transitions between levels are assumed to occur with exponential transition rates that may depend on the current level. Thus, unlike Model A, the rate will approximate by a continuous-time process \( \lambda(t) \) with discrete jumps at random Poisson times. The rate of these Poisson times and the probability of the jump size may change depending on the level of the bit rate. Model B can be obtained from the continuous state bit rate sampling the latter at random Poisson time instances (with appropriate rates) and quantizing the state at these points.
Figure 7.6: Fluid-flow model for a finite-buffer switching node under ON/OFF traffic sources [44].

Model B is a discrete finite-state, continuous-time Markov process. Its state space is the set of the quantized levels up to a maximum level. The quantization step, the number of states, and the transition rates can be tuned to fit the average variance and autocovariance function of the measured data as before. Model B will be used to analyze the statistical multiplexer as a continuous-state queue (a fluid reservoir) that is filled from N variable rate sources each with rate $\lambda(t)$. Thus, parameters are tuned to the aggregate instantaneous input rate $\lambda(N)$, rather than the single source bit rate $\lambda(t)$. The total rate is

![Graph showing the number of GOPs in terms of bit rate for Aggregate movie source (Star Wars IV).](image)

Figure 7.7: Number of GOP in terms of bit rate for Aggregate movie source (Star Wars IV)

The sum of N independent random processes each with mean $E(n)$ and autocovariance $C(\tau) \approx C(0)e^{-\alpha\tau}$ steady state. The steady-state mean and autocovariance of $\lambda(N)$ will be
\[ E(\lambda_N) = N \times E(\lambda) \equiv 0.149 \times N \text{ bits/pixel} \]
\[ C_N(\tau) = N \times C(0) \times e^{-\tau} \equiv 0.0061 \times N \times e^{-0.03\tau} \text{ (bit/pixel)}^2 \]  

(7.5)

There are an infinite number of choices of Markov models that can fit the parameters in (7.5). The actual choice should be guided by further examination of the video coder output and the complexity of the corresponding queuing analysis. As an example, if the coding rate involves sudden increases due to error recovery procedures or abrupt change of scene, this can be modeled by allowing big jumps in \( \lambda_N(t) \) with a certain transition rate. In this thesis, the phenomena are not considered. In the recorded data, Fig. 7.7, discontinuities and very steep changes were not observed. The state \( \lambda_N(t) \) of the process represents the quantized level of the aggregate bit rate of \( N \) sources. It is assumed that uniform quantization step \( A \) bit/pixel, and \( M + 1 \) possible levels, \((0, A, \ldots, a, MA)\). The exponential transition rates \( r_{ij} \) from state \( iA \) to state \( jA \) are given by

\[ r_{ij} = \begin{cases} 
(M - i)\alpha, & i < M \\
i\beta, & i > 0 \\
0, & \text{otherwise}
\end{cases} \]

(7.6)

It can be easily shown that \( \lambda_N(t) \) at steady state will have a binomial distribution with mean \( E(\lambda_N) \), variance \( C_N(0) \), and exponential autocovariance \( C_N(\tau) \),

\[ P\{\lambda_N(t) = kA\} = \left(\frac{\alpha}{\alpha + \beta}\right)^k \left(1 - \frac{\alpha}{\alpha + \beta}\right)^M, \quad p = \frac{\alpha}{\alpha + \beta} \]  

(7.7)

\[ E(\lambda_N) = MAp \]  

(7.8)

\[ C_N(0) = MA^2 p(1 - p) \]  

(7.9)

\[ C_N(\tau) = C_N(0)e^{-(\alpha + \beta)\tau} \]  

(7.10)

The parameters of the model \( M, A, \alpha, \text{ and } \beta \) are obtained by matching (7.7)-(7.10) with the measured values in (7.5). With the number of quantization levels \( M \) as a parameter, and for a given number of multiplexed sources \( N \), they are
\[ \beta = a \left( 1 + \frac{N \times E^2(\lambda_N)}{M \times C_N(0)} \right) \]

\[ \alpha = a - \beta \]

\[ A = \frac{C_N(0)}{E(\lambda_N)} + \frac{E(\lambda_N)}{M} \] 

(7.11)

The bit rate can span the interval \( 0 \leq \lambda_N(t) \leq MA \); \( M \) should be selected large enough to span all likely bit rate values. Thus, for a finite number of sources \( N \), the binomial model does not converge to the continuous-state Model A by decreasing the step size. Nevertheless, it is found that with \( M = 20 \times N \), the analytic results using Model B are in close agreement with simulations that use Model A.

Table 7.3: Peak rate values and Markov source transition parameter values for different video traces

<table>
<thead>
<tr>
<th>Video Trace File</th>
<th>Simulation</th>
<th>Steady state average ( E(\lambda) ) over 8 sources</th>
<th>OFF to ON transition rate (( \alpha ))</th>
<th>ON to OFF transition rate (( \beta ))</th>
<th>Peak Rate (P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars IV</td>
<td>GOP</td>
<td>0.188399</td>
<td>1.097774</td>
<td>1.105054</td>
<td>1.75100</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>0.188399</td>
<td>0.016723</td>
<td>0.019873</td>
<td>1.75100</td>
</tr>
<tr>
<td>Jurassic Park I</td>
<td>GOP</td>
<td>0.648986</td>
<td>0.538114</td>
<td>0.563024</td>
<td>2.10400</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>0.648986</td>
<td>0.001156</td>
<td>0.006156</td>
<td>2.10400</td>
</tr>
<tr>
<td>Soccer</td>
<td>GOP</td>
<td>2.682512</td>
<td>1.625459</td>
<td>1.814659</td>
<td>2.14400</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>2.682512</td>
<td>0.001081</td>
<td>0.000652</td>
<td>2.14400</td>
</tr>
<tr>
<td>Mr. Bean</td>
<td>GOP</td>
<td>0.445810</td>
<td>0.498850</td>
<td>0.515270</td>
<td>2.27100</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>0.445810</td>
<td>0.011153</td>
<td>0.016883</td>
<td>2.27100</td>
</tr>
<tr>
<td>S_Lambs</td>
<td>GOP</td>
<td>0.424418</td>
<td>0.571962</td>
<td>0.579242</td>
<td>1.89500</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>0.424418</td>
<td>0.006395</td>
<td>0.009545</td>
<td>1.89500</td>
</tr>
</tbody>
</table>
In the binomial model, the rate $\lambda_t(t)$ can be thought of as the aggregate rate from $M$ independent minisources, each alternating between transmitting 0 bit/pixel (called the OFF state) and A bit/pixel (the ON state) according to a Bernoulli distribution. A minisource turns ON with exponential rate $\alpha$ and OFF with rate $\beta$. The ON state represents active talkspurts and the OFF period to silence intervals, both assumed exponentially distributed. The aggregate rate out of $M$ voice sources corresponds exactly to our quantizing the aggregate bit rate of $N$ video sources into $M$ levels.

From the fig 7.9 the observation follows the same conclusion we got in the previous
one. It is observed from the plot above that the movies Mr. Bean and Jurassic Park I responds late at the congestion state. Soccer and Silence of the Lambs both shows a very tolerant characteristic for this increment of buffer size in course of time. The best result is obtained by Star Wars IV because of its least change in between GOP frames.

From fig 7.10 Performance measures of QoS is the packet loss probability of the system. With the same buffer size of 5 and link capacity 11mbps the different movie loss characteristic is shown below.

Figure 7.10: Buffer size Vs Loss probability in different sources

Due to rapid change of frames Mr. Bean movie has got higher loss rate. It is seen that the average loss is 3.5 at initial buffer size 0. But it decreases with the increase of buffer size and reaches at a steady state at buffer size 5 and finally converges at higher buffer size. Again Star wars IV has achieved the lowest loss rate than the other four where Mr. Bean has the highest loss probability. Frequent frame changes occur most in the two movies, soccer and silence of the lambs, have maintained an average of 3.5.
The overflow characteristic of different movies seems quite interesting. Severe congestion occurs at buffer size 0-4 that is why packets are over flown at higher rate at that time. But with the increase of buffer size they all don’t get to a steady state. They keep getting better at a higher buffer size.

Figure 7.11: Buffer Size Vs Overflow in different sources
CHAPTER-8

8. Analytical Framework for QoS evaluation:

To quantify the number of IPTV connections being supported with satisfactory QoS and determine the network performance with very bursty IPTV traffic, an analytical framework based on the fluid flow approach is developed in this chapter. The approach could be applied specifically for network scenarios, considering the properties of the video traffic and the accompanying network characteristics.

8.1 Multiplexed Queue System:

For the analysis, a simple FIFO queue management scheme serves a number of independent video sources, and was used for the aggregate multiplexed video traffic. In this case used 8 numbers of input video sources, and individual VBR video source was modeled by using a same type of source model. All video sources used the same model data or real data, hence they were homogeneous sources. Each source is stored in a separate prebuffer, which assembles the data into blocks and packetized the blocks. The packets from the all the prebuffers join a common buffer in the multiplexer, where the packets are queued for transmission over a high-speed communication line [17]. The departure process of the buffer was deterministic with FIFO discipline. Since all used sources were video sources, no source had any priority over each other.
8.1.1 Fluid flow model:

Fluid flow models assume that the packet/Frame arrival process at a multiplexer occurs continuously in time and may be characterized by continuous random fluctuations in the arrival. The computational model affords the estimation of the delay, loss and overflows distributions in multiplexers fluid sources and served at constant rate.

The FIFO multiplexer is offered traffic from K classes of ON/OFF sources. An ON/OFF source alternates between an active ON state, when it transmits information at a peak rate, and a silent OFF state. The durations of the ON and OFF periods are assumed to be exponentially distributed. A traffic class $i$ is described by four parameters:

- Number of sources: $N_i$
- Peak bit rate: $p_i$ [Mbps]
- ON to OFF transition rate: $\beta_i$ [S$^{-1}$]
- OFF to ON transition rate: $\alpha_i$ [S$^{-1}$]
The intensity (rate) of transitions from the ON (OFF) state to the OFF (ON) state for a class i source is \( \beta_i(\alpha_i) [S^{-1}] \).

The mean rate for class i is given by \( m_i = p_i \frac{\alpha_i}{\alpha_i + \beta_i} [bps] \).

The data transmitted by the \( \sum N_i \) source is received by a finite buffer of size B Mbits with a maximum output rate of C bps. The buffer is modeled as a fluid reservoir. If \( p_{\text{in}} \) is the input rate, and \( x \) is the buffer content, then the change in buffer content is given by

\[
dx{t} = \begin{cases} 
0 & \text{when } x = 0 \text{ and } p_{\text{in}} < C \\
p_{\text{in}} - C & \text{when } 0 < x < B \\
x = B \text{ and } p_{\text{in}} < C \text{ and } x = B \\
0 & \text{when } x = B \text{ and } p_{\text{in}} > C
\end{cases}
\]

The average input rate is \( \sum_{i=1}^{c} N_i p_i \frac{\alpha_i}{\alpha_i + \beta_i} \) and the load on the output is then given by

\[
p = \frac{\sum_{i=1}^{c} N_i p_i \frac{\alpha_i}{\alpha_i + \beta_i}}{C} \tag{8.1}
\]

The average input rate to be smaller than the output capacity, i.e. \( p < 1 \).

Let \( k_i \) denote the number of active sources in class i, \( k=(k_1,\ldots,k_c) \) the state vector which the sources are in. Let

\[
S := \{k = (k_1,\ldots,k_c) : 0 \leq k_i \leq N_i, i = 1,\ldots,c\} \tag{8.2}
\]
Denote the state space for the sources. S is of cardinality $N_1 + 1 \ldots \ldots (N_e + 1)$.

Let $F_i(x)$ denote the probability that the queue length is less than $x$, $\{F_k(x)\}_{k=x}$ denote the stationary buffer distribution column vector, where $F_i(x) = P_r(\sum = k, Q \leq x), k \in S_1, 0 \leq x \leq B$. The equilibrium queue length distribution at the bottleneck link is subject to

$$D \frac{d}{dx} F(q) = MF(x), 0 < x < B,$$  \hspace{1cm} (8.3)

Where $D$ is a diagonal matrix with entry $(k, k)$ equal to

$$d_k = (\sum_{i=1}^k p_{ki} - C)$$  \hspace{1cm} (8.4)

And where entry $(k, n)$ in $M$ looks as follows:

$$m(k, k) = -\sum_{i=1}^k (N_i - k_i)\alpha_i + k_i\beta_i, \text{ for } k \in S$$  

$$m(k, k_1 \ldots \ldots, k_{i-1}, \ldots, k_j, \ldots, k_k) = (N_i - \beta_i + 1)\alpha_i, \text{ for } k \in S$$  \hspace{1cm} (8.5)

$$m(k, k_1 \ldots \ldots, k_{i-1}, \ldots, k_j, \ldots, k_k) = (k_i + 1)\beta_j, \text{ for } k \in S$$

### 8.2 Solution:

In the infinite buffer case with identical sources, the initial condition is different, and it can be formulated, and an analytical solution exists. However, in the finite buffer case with a more general input stream, no such approach seems possible, and the equation must be solved numerically. The solution $F(x) = (F_k(x))_{k=x}$ is obtained from a spectral expansion:

$$F_k(x) = \sum_{n=\infty} a_n \exp(z(n)\varphi_n)k$$  \hspace{1cm} (8.6)

Hence, the $\{z(n)\}$ are the solution to the generalized eigenvalue problem $zD\varphi = M\varphi$. In practice, the eigenvalue for state $k$ is found by solving a non-linear algebraic equation. The eigenvector $\varphi_n$ corresponding to the eigenvalue $z(n)$ is given by the coefficients of certain polynomial in K variables. The coefficients $a_k$ are found by
means of the initial condition.

\[ F_k(0) = 0, \quad a(k) > C \]
\[ 0 = u_k = \pi(k) - \lim_{x \to B} F_k(x), \quad a(k) = C \]  

(8.7)

Where \( \pi(k) \) denotes the overall probability of sources being in state \( k \). This probability is found from the multi-binomial distribution

\[ \pi(k) = \prod \left( \frac{N}{k!} \left( \frac{\alpha_i}{\alpha'_i + \beta'_i} \right)^{k_i} \left( \frac{\alpha'_i}{\alpha_i + \beta_i} \right)^{s_i} \right) \]  

(8.8)

Given \( B \) and \( D \), using the fluid flow approach, the queue distribution can be obtained. Given the QoS requirements of IPTV traffic, including the loss rate and delay bound, the number of connections can be limited and choose an appropriate buffer size can be chosen. For the delay budget in a network, the required queue length can be determined and thus the buffer size that can be supported with a guaranteed loss rate due to buffer overflow.

### 8.3 Performance analysis with the Fluid Flow approach:

In this section, the fluid flow based performance analytical framework is developed. Given the queueing system, i.e., the generating matrix of the arrival rate and service rate at each state, the queue distribution can be obtained by solving a uniform derivative equation. The fluid buffer overflows, loss probability, and mean waiting time in the queue.

#### 8.3.1 Fluid Overflow Probability:

The survivor function \( G(x) \), which represents the probability of buffer overflow, is the complementary distribution of \( F(x) \). Variations in the traffic rate and channel bandwidth have opposite effects on the admission region. The input rate of the data traffic, for example, from the IPTV traffic point of view, can be considered as a deduction from the available capacity of the output link. The buffer overflow
probability G(x) = Pr (Q>x) it can be described by the following equation:

\[ G(x) = 1 - \sum_{k=0}^{\infty} F_k(x) \quad (8.9) \]

### 8.3.2 Fluid Loss Probability:

The overall loss probability \( P_{\text{loss}} \) is defined by fluid generated by the sources is greater than the fluid than can be drained by the channel (\( BS > C \)) and only in these states can the fluid be lost. This probability only computes the percentage of time that the system is losing information, and it can be readily described by the following equation:

\[ P_{\text{loss}} = \frac{\sum_{\{k.f > C\}} (k.f - C)u_k}{\sum_{i=1}^{k} N_i P_i \left(\frac{\alpha_i}{\alpha_i + \beta_i}\right)} \quad (8.10) \]

The fluid loss probability \( P_{\text{loss}} \) for sources in class \( j \) is the fraction between lost class \( j \) information to the offered class \( j \) information, and is therefore given as

\[ P_{\text{loss}}(i) = \frac{\sum_{\{k.f > C\}} k_i P_i (k.f - C)u_k}{\sum_{i=1}^{k} N_i P_i \left(\frac{\alpha_i}{\alpha_i + \beta_i}\right)} \quad (8.11) \]

### 8.3.3 Fluid Mean Delay Probability:

The delay can be described by the mean delay or the (1- \( \alpha \)) quintile of the delay distribution. The packetization delay is the time it takes to accumulate enough bits to fill a packet. The queue distribution can be for IPTV traffic, including the delay bound, the number of connections can be limited and an appropriate buffer size can be chosen. For instance, given the delay budget in a home network, the required queue length are determined and thus the buffer size.
The mean queue length is given by

$$\overline{Q} = \int_0^\beta x \frac{d \Pr(Q \leq x)}{dx} dx$$

$$= \sum_{k=x} \left[ xF_k(x) \right]_0^\beta - \int_0^\beta F_k(x) dx + Bu_k \right] \right)$$

(8.13)

Where $F_k(B-)$ is defined as $\lim_{x \to B} F_k(x)$. The mean queuing delay for class $j$, $\overline{W}_j$, is obtained from Little’s formula

$$\overline{W}_j = \frac{\overline{Q}}{N_j m_j (1 - Ploss(i))}$$

(8.14)

Where $m_j$ denotes the mean arrival rate of a source from class $j$.

$$m_j = p_i \frac{\alpha_i}{\alpha_i + \beta_i}$$

(8.15)
CHAPTER-9

9. Simulation framework for QoS evaluation:

9.1 Discrete-Event Simulation:

Discrete-event simulation [33] is a way to build a model, so that the dynamic (time based) behavior of the system can be observed. In the system each event occurs at an instant in time and marks a change in the state of the system. During the experimental phase the Discrete event model is executed (run over time) in order to generate results. The results can then be used to provide insight into a system and forms a base to make decisions on.

The general steps involved in the development of a DES model starts by

- Determining the Goals of the system to be developed
- Building of a conceptual model.
- Converting it into a specification model.
- Followed by converting the specification model into a computational model.
- Verifying the system developed in the previous step and finally the validation (computational model being consistent with the system being analyzed) of the system.

To evaluate the performance and the behavior of the scheduling algorithms, a simulation for the thesis work was developed using C programming language with Window Xp/Linux platform. The network simulation used Markov model for channel simulation, On/Off model for traffic generation and the scheduling algorithms for scheduling purposes. The simulation was based on discrete event simulation (Markov model) using fluid flow model (On/Off), at the network layer, to asses and analyzes the performance of scheduling algorithms (FIFO).
Discrete event simulation uses discrete sequence of events to formulate the system. It has five following components;

- **Clock**: It is the time keeper of the system. It keeps track of the time regarding which time interval is occurring and for how long etc.

- **Random number generator**: The simulation needs to generate random variables of various kinds, depending on the system model. This is accomplished by one or more pseudorandom number generators.

- **Statistics**: It keeps the system statistics for study and analysis.

- **Ending condition**: It is the condition specified for the termination of the system’s run. It can be some time value or the number of events.

- **Event list**: The simulation maintains a list for the simulation events. It has all the events which in turn have a start time, end time, their type and the next event to come. The event list acts as the backbone of the simulation.

Every event occurs at a specific time interval and marks the change of system state. An event is taken as the wholesome of a burst arrival time, its service type and its departure time. Discrete event simulations can be implemented in one of the following four ways; event based, process based, activity based and the three phase approach based. [33]

### 9.2 Fluid flow simulation model for FIFO Multiplexer:

Network performance evaluation through traditional packet level simulation is becoming increasingly difficult as today’s networks grow in scale along many dimensions. As a consequence, fluid simulation has been proposed to cope with the size and complexity of such systems. The “simulation event” rate has been used to measure the computational effort of the simulators and show that this measure is both adequate and accurate.
Flow aggregation is considered as a technique to reduce the impact of the “ripple effect” in fluid simulation. The results show that the tradeoffs between parameters of a network model, determines the most efficient simulation approach. Another modeling technique making simplified assumptions about the real system is the fluid model, which was first proposed by Anick et al. in [32] to model data network traffic. In the fluid simulation paradigm, network traffic is modeled in terms of a continuous fluid flow, rather than discrete packet instances.

A fluid simulator keeps track of the fluid rate changes at traffic sources and network queues. An equivalent packet-level simulator would keep track of all individual packets in the network. In fluid simulation, the higher level of abstraction suggests that less processing might be needed to simulate network traffic. Intuitively, this is not surprising as a large number of packets can be represented by a single fluid chunk. For simple network components, where traffic flows do not compete for resources, the fluid simulator outperforms the packet-level simulator. An example would be a link that connects two nodes and never experiences queueing this component only introduces a constant propagation delay.

The simulator used for this work uses one class of traffic needed for the model. Short term burst level traffic was used as the traffic mode. Fluid flow in and out of the user equipments is represented in three ways, which are; inflow, outflow and net flow. Inflow is the amount of fluid traffic flowing into the user equipment. Outflow is the amount of fluid traffic flowing out of the user equipment and net flow is the difference of the traffic between inflow and outflow. The time period for traffic is exponentially distributed in both forms of fluid flow, inflow or outflow. The buffer is termed as the reservoir for the fluid traffic. The congestion parameters, overflow, is measured on the bases on inflow and outflow. If inflow is more than the outflow then the buffer overflows with network traffic and data is lost. Since the inflow is basically the traffic coming via downlink into the user equipment, which can assume to constant and focus on the outflow which deals with the uplink. The ups and downs in the rate are because of the outflow in our case.
The performance of the selected algorithms was measured and evaluated based on the congestion parameters i.e. loss, overflow and wait for users. The data transport performance was considered based upon these congestion parameters i.e. the lower the loss, overflow and wait the better the data transport performance and the better the algorithm and vice versa.

Figure 9.1: Sequence flow of the Simulation

The tasks performed by each of these modules are stated as under;

**Initialize network**: Initiate the network by setting the parameters specified in the file.

**Start up network**: To start the network with some traffic i.e. events and to start the event list.

**Generate traffic**: Generates the next traffic event and calls the scheduler.

**Scheduling**: Applies the scheduling algorithms to schedule the user traffic.

**Congestion control**: Calculation of the congestion parameters for each user, i.e. loss, wait, overflows.

The following arguments are entered by the user in order to run the simulation:
• Simulation Executable name.
• Number of source cells.
• Scheduling algorithm name.
• Simulation number (Different simulations have different buffer sizes)

The parameters for the simulation were saved in a parameter file and stored in the folder with the executable file. They were then fetched from the file by the program when it was run, which were used to initialize the network. These initialization parameters are:

• Cell number.
• Number of sources.
• Number of Markov Model states.
• Burst traffic parameters (For On-Off model).
• Link capacity.
• Initial buffer size.

The network is started off with the event generation. The simulation was run for 20 iterations for each buffer size from 1 to 10 using different types of traffic. The event list is the backbone of the program as it provides a means for the traffic. An event structure was used to store all the properties of individual events and then it was linked by a linked list. The events born by the current event are also inserted to the event list with all their properties. The random number generator of the discrete event model (Markov) is used to decide the time of each event’s occurrence. Each current event is passed to the scheduler to schedule the users generating the traffic and allow transmissions. The scheduler calls on the selected algorithms and makes the scheduling decision based on the current channel conditions, user conditions and traffic conditions. The transmission states of the scheduled users are then turned on using the On-Off model and the non-scheduled users are switched off. Then the congestion parameters are calculated to obtain the performance measures. The wait, loss and overflow are calculated as the mean of the iterations for each specific buffer size. These output congestion parameters are then used to analyze the behavior and performance of the selected scheduling algorithms. The results are then plotted in graphs to obtain a graphical study mechanism.
Pseudo Code:
The overall simulation program can be summed up into the following main steps;

Analyze Video Trace Files();
//Markov Model is used for DTM &CTM
Calculate Smoothing Parameters for simulation();
Initialize Network();
Make selections based on input arguments.
Generate Traffic();
For all buffer sizes Begin
  Establish traffic parameters.
  Initialize simulation parameters.
  Initialize the event list.
  Simulate events.
End

For all events Begin
  Apply scheduling algorithm.
  Schedule users
  Calculate congestion parameters.
End
End

Multiplexer: An important characteristic of a multiplexer is its scheduling discipline. The Fluid Flow simulation evaluates the performance for FIFO scheduling methods for a single multiplexer by burst level simulation. The performance is measured in terms of fluid overflow probability, fluid loss probability and mean fluid delay. The multiplexer is described by the selected bandwidth scheduling method (FIFO), the per-class buffer capacity (in Mbit), per-class GPS weights, and total link capacity (in Mbit/s).
In the FIFO fluid model [30], the dynamics of the multiplexer are more subtle. First, note that fluids from two different sources are distinct and do not mix in the queue, just as packets transmitted by two sources are differentiated within the queue. Fluids from different sources may arrive simultaneously at a queue (which cannot occur in the packet-level model) and the FIFO fluid queue can serve fluids from multiple sources simultaneously. The fluid departure rate can also be described of the different sources as a function of the fluid arrival rates. Note that the departure rate of a fluid flow depends not only on its own arrival rate, but also other flows as well. A departing fluid flow shares the queue’s service capacity with all other flows currently being FIFO-served in proportion to their arrival rates.
### 9.3 Simulation flow chart:

**START**
- Initialize Variable
- Read Input Parameters

**Arrival**
- Choose Sub-Event
- Schedule And Then Arrival & Departure
- Decide TDM
  - Yes: Discrete TDM
  - No: Continuous TDM
  - Continuous TDM
    - Aggregate Source
    - Find Parameters \( \alpha, \beta, A \)
  - Discrete TDM
    - Find Coefficient
    - Fit the curve to the desired one for smoothing
- Decrement Queue Size

**Departure**
- Advanced Queue Forward
- Frame-Time > Tmax
- Frame > Fmax
- Generate Output Report (Cell Loss Probability)
- STOP

**Queue = Qmax**
- Record Loss
- Increment Queue Size

**Doted line is our proposed model**
CHAPTER-10

10. Numerical Analysis:

In this section a performance evaluation of the proposed model is performed by investigating its suitability to model the queueing behavior of self-similar traffic. A system fed is simulated by different type of traffic traces and geometrically distributed service times. The observed steady state queue size distribution is compared to the one predicted by an analytical program fitted to the parameters of the traffic traces.

Given the queueing system, i.e., the generating matrix of the underlying Markov chain, arrival rate and service rate at each state, the queue distribution is obtained. The loss rate of traffic with a given queue length is then determined. In other words, the maximum buffer size can be used to support IPTV connections with guaranteed QoS. By quantifying queue performance analytically, the admission region of IPTV traffic is obtained. It can help service providers improve the design and deployment of networks to support IPTV traffic and choose a proper resource (buffer and/or bandwidth) allocation scheme. In addition, it provides important insights into which system parameters and/or traffic characteristics affecting the admission region of networks.

We have designed two experiments which reflect the performance of the proposed model. Experiment A is set to show the improvement of the bottleneck problem addressed by the traditional IPTV network where the data smoothing of the transmission channel is not considered. The three QoS evaluation parameters, loss, overflow and mean waiting time are considered as usual.

The movie considered for testing the discrete time division multiplexing is Star Wars IV. It has assumed to have the least amount changes in consequent frames and black background.
10.1 Expiment A:

10.1.1 QoS evaluation parameter analysis:

Figure 10.1: Buffer size Vs Loss Probability with statistical smoothing and without Smoothing.

It is observed that the traditional network responds reluctantly with the increment of buffer size. Packet loss remain constant at the continuous congestion state. Whereas the integration of the statistical multiplexing model A for discrete TDM shows a lot better performance over the packet loss. From the very beginning it turns to reach the minimum loss and achieves the steady loss free state by using least buffer size of 5 where the traditional model without statistical smoothing continues losing data and needs a comparatively higher size of buffer to reach a steady loss free state. The result is indeed very optimistic for DTDM. From the table below it can be seen that the data smoothing technique incredibly minimises the loss to 0 long before the traditional model. The simulator have been tested with buffer size 5 and link capacity 11.
Table 10.1: QoS evaluation parameter values for before and after smoothing

<table>
<thead>
<tr>
<th>Simulator</th>
<th>Trace File</th>
<th>OFF to ON transition rate (α)</th>
<th>ON to OFF transition rate (β)</th>
<th>Peak Rate (P)</th>
<th>Overall Loss</th>
<th>Overall Overflow</th>
<th>Overall Mean Waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Smoothing</td>
<td>N/A</td>
<td>0.5</td>
<td>0.5</td>
<td>2.0</td>
<td>0.02193</td>
<td>0.0772</td>
<td>0.0477</td>
</tr>
<tr>
<td>With Smoothing</td>
<td>Star Wars IV</td>
<td>1.0977</td>
<td>1.1050</td>
<td>1.751</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.00335</td>
</tr>
</tbody>
</table>

Figure 10.2: Buffer size Vs Overflow with statistical smoothing and without Smoothing.

When the total inflow exceeds the link capacity then overflow of the data packets occurs. In DTDM it happens for a good number of buffer sizes, thing to check out here is how the improved model works over the traditional model. For the same movie Star Wars IV the model without statistical smoothing shows up with a high overflow rate over increasing buffer size. It is seen that it takes exponential shape over time and takes a very long time to get to a steady state. It is quite surprising that the new proposed method reaches the steady state at the very first buffer. That means it uses the minimum available buffer very efficiently. The starting overflow rate is...
also quite significant. Improved model has a very low overflow rate at the beginning.

Mean delay of the data in the buffer is measured for both the model. For a smooth transition of data packets the delay is minimized to a negligible amount of time. It does not take more than 0.005s and handles the congestion by that time. On the other hand traditional model takes a long to get to a steady state.

10.1.2 Comparison of results with analytical simulator:

To verify the proposed network model and demonstrate the effectiveness of the proposed video model in network simulations and analysis, the analytical results are compared with simulation results with real video traces. In the following section, explain the approach to determine the model accuracy for an analytical algorithm.

This analytical program calculates, based on an approach described in S.B. Jacobsens[8] paper, the time congestion, overall loss ratio and individual loss ratio in a finite buffer receiving cells from up to three different classes of on/off sources. The approach is an eigenvalue and eigenvector approach and in the case with very many (100 or more) sources in one class the program is unstable. However, it is quite insensitive to a high fraction: large buffer capacity to average number of cells in an
average on period.

The below table shown that comparison between the improved simulator and the analytical program on the QoS evaluation parameters.

Table 10.2: QoS evaluation parameter values for Simulator and Analytical program

<table>
<thead>
<tr>
<th>Video Trace File</th>
<th>Number of Source</th>
<th>Buffer Size</th>
<th>Link Capacity</th>
<th>Simulator Type</th>
<th>Overall Loss</th>
<th>Overall Overflow</th>
<th>Overall Mean Waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars IV</td>
<td>8</td>
<td>5</td>
<td>11</td>
<td>Simulator</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00335</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Analytical</td>
<td>2.20829e-010</td>
<td>8.34056e-010</td>
<td>0.00355</td>
</tr>
</tbody>
</table>

Figure 10.4: Buffer size Vs Loss Probability in two different simulators

Figure 10.5: Buffer size Vs Overflow Probability in two different simulators

The result clearly depicts that new improved model for continuous TDM performs no worse than the approach taken by Dittman and Jacobsen [8]. However, the loss and delay probability is slightly better found in the proposed model.

10.2 Experiment B:

We now test new improved model for different types of movies. The reason behind is that the change in subsequent frames varies on the basis of different scenarios. Theoretically the movies which have fixed dark background like Star Wars IV has least change of bits per frame than the movies where changes occur very frequently like Mr. Bean. Mew improved model is tested on all these sort of movies. The
measurement parameters remain same. The model has been simulated three times namely sim1, sim2 and sim3. In Sim1, dynamic buffer size is tested which ranged 0-10 where in sim2 link capacity is varied from 6-16. It is also tested that how the model respond when the number of sources increase with time. Sim3 contains the test result for the different number of sources.

10.2.1 Simulation-1: Investigation of variable buffer size for GOP:

Table 10.3: QoS evaluation parameter values for simulation-1 in GOP

<table>
<thead>
<tr>
<th>Video Trace File</th>
<th>Number of Source</th>
<th>Buffer Size</th>
<th>Link Capacity</th>
<th>Overall Loss</th>
<th>Overall Overflow</th>
<th>Overall Mean Waiting time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr. Bean</td>
<td>8</td>
<td>5</td>
<td>11</td>
<td>0.443026</td>
<td>1.56985</td>
<td>0.15209</td>
</tr>
<tr>
<td>Jurassic Park I</td>
<td>0.04908</td>
<td>0.14630</td>
<td>0.06602</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_Lambs</td>
<td>0.00089</td>
<td>0.00393</td>
<td>0.01976</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Soccer</td>
<td>0.000007</td>
<td>0.00002</td>
<td>0.02156</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Wars IV</td>
<td>0.000000</td>
<td>0.00000</td>
<td>0.00335</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10.6: Buffer size Vs Loss Probability for different trace files in GOP
Analysis: If we have a look at the loss probability of the movies, we see the reflection of the expected behavior. Mr. Bean has got the highest loss probability than the other movies and Star Wars IV has got the minimum. Thing to be noticed is that all loss of traces starts almost a same loss interval. Loss decreases over the increase of buffer size. As for the overflow of the whole network the proposed model does a very impressive job. From the very first buffer allocation most of the movies achieve a steady rate of improvement. The worst case is for the movie Mr. Bean which comes
down to 15 from 35 of overflow in the first buffer and soon reaches to the steady state of no overflow. When it comes to mean waiting time, Mr. Bean and Jurassic Park have got the highest waiting time than the other movies. This waiting time increases with the increment of number of buffer size. Soccer and Silence of the Lambs both shows a very tolerant characteristic for this increment of buffer size in course of time. The best result is obtained by Star Wars IV because it takes less waiting time than the other movies.

10.2.2 Simulation-2: Investigation of variable link capacity for GOP:

Table 10.4: QoS evaluation parameter values for simulation-2 in GOP

<table>
<thead>
<tr>
<th>Video Trace File</th>
<th>Number of Source</th>
<th>Buffer Size</th>
<th>Link Capacity</th>
<th>Overall Loss</th>
<th>Overall Overflow</th>
<th>Overall Mean Waiting time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr. Bean</td>
<td>8</td>
<td>5</td>
<td>16</td>
<td>0.00003</td>
<td>0.00012</td>
<td>0.00066</td>
</tr>
<tr>
<td>Jurassic Park I</td>
<td></td>
<td></td>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00011</td>
</tr>
<tr>
<td>Soccer</td>
<td></td>
<td></td>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00005</td>
</tr>
<tr>
<td>S_Lambs</td>
<td></td>
<td></td>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Star Wars IV</td>
<td></td>
<td></td>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

Figure 10.9: Link capacity Vs Loss probability for different trace file in GOP
Analysis: An optimistic result is found regarding the loss probability for variable link capacity. Four movies out of five reaches become loss free before link capacity 15. Star Wars IV converges more quickly than the others where Mr. Bean has got the maximum link capacity needed for transmission. Similar scenario can be seen in the
overflow probability plot. Soccer and Silence of the Lambs have been handled in almost the link capacity 10 though soccer is more computationally expensive than the other one. Jurassic park I movie has sudden changes in between frames with changing background. This is why it takes a more link capacity but also has got the similar characteristic curve like other movies. Mr. Bean has got the maximum overflow rate but it dramatically drops down in first 10. It is observed that in the first 7 link capacity increment have got the fast drop of the overflow and it then steadily drops till 10. It mostly reflects on Mr. Bean movie.

10.2.3 Simulation-3: Investigation of variable number of sources for GOP:

Table 10.5: QoS evaluation parameter values for simulation-3 in GOP

<table>
<thead>
<tr>
<th>Video Trace File</th>
<th>Number of Source</th>
<th>Buffer Size</th>
<th>Link Capacity</th>
<th>Overall Loss</th>
<th>Overall Overflow</th>
<th>Overall Mean Waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars IV</td>
<td>16</td>
<td>5</td>
<td>11</td>
<td>42.6334</td>
<td>62.7534</td>
<td>1.15208</td>
</tr>
<tr>
<td>Jurassic Park I</td>
<td></td>
<td></td>
<td></td>
<td>33.224</td>
<td>81.823</td>
<td>0.43700</td>
</tr>
<tr>
<td>Soccer</td>
<td></td>
<td></td>
<td></td>
<td>32.1406</td>
<td>78.3299</td>
<td>0.44564</td>
</tr>
<tr>
<td>Mr. Bean</td>
<td></td>
<td></td>
<td></td>
<td>38.4852</td>
<td>90.312</td>
<td>0.44407</td>
</tr>
<tr>
<td>Silence of the Lambs</td>
<td></td>
<td></td>
<td></td>
<td>27.1173</td>
<td>76.7824</td>
<td>0.42477</td>
</tr>
</tbody>
</table>

Figure 10.12: Number of sources Vs Loss probability for different trace files in GOP
Analysis: The proposed model shows quite a sensitive response to the increment of number of sources. Loss probability increases linearly with the increment of number of sources. Interesting thing is almost all the movies reaches to steady loss of packets on the source 16. The same thing happened in the overflow measurement. A bell shaped increment of overflow packets is seen. Star Wars IV has got the minimum overflow where Mr. Bean has got the maximum. As it comes to mean waiting time, a very strange phenomenon has been observed. The movie Star Wars IV has the packets in the buffer due to the occurrence of congestion. The mean delay unlike the other
movies increases at a steady rate till 12th number of sources. But after that it dramatically takes more time than usual. It even takes more than a second to process frames for 16 sources. For the same type movies like Star Wars IV the dark background pixel values probable take more space to buy excess time for transmission unlike other movies which almost all got to a steady increase of waiting time till 16.

10.2.4 Simulation-1: Investigation of variable buffer size for Frame:

Table 10.6: QoS evaluation parameter values for simulation-1 in Frame

<table>
<thead>
<tr>
<th>Video Trace File</th>
<th>Number of Source</th>
<th>Buffer Size</th>
<th>Link Capacity</th>
<th>Overall Loss</th>
<th>Overall Overflow</th>
<th>Overall Mean Waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soccer</td>
<td>8</td>
<td>5</td>
<td>11</td>
<td>9.44385</td>
<td>36.2288</td>
<td>0.3560</td>
</tr>
<tr>
<td>Mr. Bean</td>
<td></td>
<td></td>
<td></td>
<td>1.97263</td>
<td>7.77219</td>
<td>0.18440</td>
</tr>
<tr>
<td>S_Lambs</td>
<td>8</td>
<td>5</td>
<td>11</td>
<td>0.39030</td>
<td>2.6661</td>
<td>0.06665</td>
</tr>
<tr>
<td>Star Wars IV</td>
<td></td>
<td></td>
<td></td>
<td>0.22057</td>
<td>0.89764</td>
<td>0.03660</td>
</tr>
<tr>
<td>Jurassic Park I</td>
<td></td>
<td></td>
<td></td>
<td>0.01721</td>
<td>0.02601</td>
<td>0.00351</td>
</tr>
</tbody>
</table>

Figure 10.15: Buffer Size Vs Loss probability for different trace files in Frame
Figure 10.16: Buffer Size Vs Overflow for different trace files in Frame

Figure 10.17: Buffer Size Vs Waiting time for different trace files in Frame

Analysis: Now analyze the frame by frame analysis for the QoS parameters. From the table above we see that the movie which has got the sudden changes in frames has produced worse result in other words has maximum loss, overflow and delay. The loss probability curve shows that the movie Soccer has the maximum loss probability where Jurassic Park I has the minimum. Sudden change causes more pixels in resultant frame transmission. That is why it needs more bandwidth allocation for transmission. For a limited resource it produces a high loss of packets. Mr. Bean then
shows the higher loss probability than the rest. The maximum loss stays between 9 and 10. The overflow tells us the same story with the Soccer with the highest overflow and Jurassic Park with the lowest. But unlike loss the overflow probability decreases as the buffer size increases. Mean delay analysis shows the Soccer and the Jurassic Park movies responds totally opposite. Delay increases linearly with buffer size for Soccer and Jurassic Park stays with almost the zero line. The conclusion can be drawn from the frame analysis that this model helps the movies to maintain a steady and optimized transmission. Diversity between the trace files distinguishes them from one another. Movies which are sensitive to sudden scene change takes more transmission capacity but still this model manage to get them steady transmission.
CHAPTER-11

11. Conclusion and Future work:

11.1 Conclusion:

In this thesis, models and results have been presented that assess the performance of numerical multiplexing of independent video sources. Our results indicate that the probability of loss, overflow and mean waiting time based on different video trace file beyond an acceptable limit drops dramatically as the number of multiplexed sources increases beyond one.

In this work two sources Markovian traffic model for video sources are presented. Model A is an autoregressive continuous-state, discrete-time Markov process, which was used to generate source data in simulation experiments. Model B is a discrete-state, continuous-time Markov process that is used in deriving a fluid-flow queueing analysis. The bursty HDTV video traces have been investigated to demonstrate the accuracy of the proposed traffic model for studying network performance. The model is simple and can easily be incorporated in network simulators, and can also be used to quantify FIFO queue performance analytically. Therefore, the model can be an effective for IPTV performance evaluation via analysis and/or simulations.

Two experiments have been conducted to watch the behavior model of the IPTV data transition. It is found that the proposed statistical data smoothing has an impressive impact on the overall outflow. Other methods seen in commercial establishment for data smoothing are simply out performed by this proposed Markovian fluid flow analysis model. Five different movie traces are used for extreme evaluation which helps this work achieve the accurate measure for analysis.
11.2 Further Work Issues:

To ensure the success and QoS of IPTV services, there are many related research issues beckon for further investigation. There are four main open issues that need to be addressed:

- This proposed model can be extended to long-range dependency traffic model which can be used for further investigation in IPTV network.

- Reactive traffic control based on flow and congestion control can be applied for network performance evaluation in IPTV network rather than preventive control.

- In addition, the proposed traffic model has been proven to be effective and accurate for FIFO queue performance analysis and simulations. Therefore, different queueing method like priority, GPS and LLQ can be exploited for network performance and simulation.

- This work is tested on one class of data and Markov model for fluid flow analysis. More than one classes is general scenario, this network can be tested for those data classes and for other model like Pareto.

Based on the above conclusions, the proposed video traffic model and the analytical framework for IPTV systems are promising to determine FIFO queue distribution of IPTV connections. Many opportunities for further improvement still exist. A few key issues have been outlined in this chapter in the hope of inspiring further research interest in this important area.
Reference:

[1] Emad Shihab and Lin Cai “IPTV Distribution Technologies in Broadband Home Networks” University of Victoria, Canada.


[15] FG IPTV-C-0315 “Proposal on requirements for multicast capability” ITU,
Mountain View, USA, 22-26 January 2007.


[18] FG IPTV-C-0978 “Open IPTV Service Model and Scenario using NGN Open Service Environment” Telecommunication Standardization Sector ITU.


[22] Patrick Seeling And Martin Reisslein “evaluating multimedia networking mechanisms using video traces”.

[23] A. Alheraish, S. Alshebeili and T. Alamri “Regression Video Traffic Models in Broadband Networks” King Saud University, Saudi Arabia.


[40] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek “The Click modular router”, MIT Laboratory for Computer Science


[44] Esteve Pallarés-Segarra, Joan Garc‘ia-Haro “Fluid-flow approach to evaluate the information loss probability in a finite buffering switching node under heterogeneous ONOFF input traffic sources”, Polytechnic University of Catalonia (UPC), Spain


[47] Cisco IOS 12.0 quality of service By Cisco Systems, Inc


[52] Bruce Chen, “Simulation And Analysis Of Quality Of Service Parameters In Ip Networks With Video Traffic”, Simon Fraser University, May 06, 2002


Appendix

Data flow diagram of the proposed model: