WASHINGTON UNIVERSITY IN ST. LOUIS
School of Engineering and Applied Science
Department of Computer Science and Engineering

Dissertation Examination Committee:

Raj Jain, Chair
Chenyang Lu
Chris Gill
Ron Cytron
Yixin Chen
Joseph A. O'Sullivan
Tarek Abdelzaher

MODELING AND DYNAMIC RESOURCE ALLOCATION FOR HIGH DEFINITION AND MOBILE VIDEO STREAMS

by

Abdel-Karim Al-Tamimi

A dissertation presented to the
Graduate School of Arts and Science
of Washington University in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2010
Saint Louis, Missouri

ABSTRACT OF THE DISSERTATION

MODELING AND DYNAMIC RESOURCE ALLOCATION FOR HIGH DEFINITION AND MOBILE VIDEO STREAMS

by

Abdel-Karim Al-Tamimi

Doctor of Philosophy in Computer Engineering
Washington University in St. Louis, 2010
Research Advisor: Prof. Raj Jain

Video streaming traffic has been surging in the last few years, which has resulted in an increase of its Internet traffic share on a daily basis. The importance of video streaming management has been emphasized with the advent of High Definition (HD) video streaming, as it requires by its nature more network resources.

In this dissertation, we provide a better support for managing HD video traffic over both wireless and wired networks through several contributions. We present a simple, general and accurate video source model: Simplified Seasonal ARIMA Model (SAM). SAM is capable of capturing the statistical characteristics of video traces with less than 5% difference from their calculated optimal models. SAM is shown to be capable of modeling video traces encoded with MPEG-4 Part2, MPEG-4 Part10, and Scalable Video Codec (SVC) standards, using various encoding settings.

We also provide a large and publicly-available collection of HD video traces along with their analyses results. These analyses include a full statistical analysis of HD videos, in addition to modeling, factor and cluster analyses. These results show that by using SAM, we can achieve up to 50% improvement in video traffic prediction accuracy. In addition, we developed several video tools, including an HD video traffic generator based on our model. Finally, to improve HD video streaming resource management, we present a SAM-based delay-guaranteed dynamic resource allocation (DRA) scheme that can provide up to 32.4% improvement in bandwidth utilization.

Acknowledgements

I am grateful to Prof. Raj Jain for the advice, the support, the availability and the enthusiasm he has always offered. I am also grateful to all of Prof. Chris Gill, Prof. Ron Cytron, Prof. Chenyang Lu, Prof. Yixin Chen, Prof. Joseph A. O'Sullivan and Prof. Tarek Abdelzaher, for accepting to serve on my dissertation committee.

Thanks to Yarmouk University in Jordan for providing me with the financial support to get both my master and Ph.D. degrees. A part of this dissertation work has been partially supported by a grant from the Application Working Group of WiMAX Forum[1].

I am thankful for all my friends for their encouragement and great memories.

I am in a great debt to my parents: my father Rihab Al-Tamimi, and my mother Elham Abu Monshar for their continuous love encouragement, and support.

Again, to my family and friends, thank you!

Abdel-Karim Al-Tamimi

Washington University in St. Louis
August 2010

[1] WiMAX, Mobile WiMAX, Fixed WiMAX, WiMAX Forum, WiMAX Certified, WiMAX Forum Certified, the WiMAX Forum logo, and the WiMAX Forum Certified logo are trademarks of the WiMAX Forum.

Table of Contents

Acknowledgements iii

List of Tables vii

List of Figures ix

List of Abbreviations xii

List of Symbols xvi

1 Introduction 1

1.1 Dissertation Main Contributions 3

1.2 Organization 5

2 Introduction to WiMAX Networks 7

2.1 WiMAX System Model 8

2.2 WiMAX Quality of Service (QoS) Classes 10

2.3 WiMAX Application Classes 12

2.4 Quality of Experience Assessment Methods in Video Applications 13

3 MPEG Video Standards 15

3.1 MPEG Encoding Basics 16

3.2 MPEG-4 Part10 / AVC Standard 20

3.3 MPEG-4 Part 10 SVC Extension 22

4 Modeling HD Video Traces Using Time Series Analysis 25

4.1 Modeling Video Traces: Related Work 30

4.2 Modeling MPEG-4 Part 2 Video Traces 32

4.3 Modeling MPEG-4 Part 10 / AVC Video Traces 40

4.4 Modeling SVC-TS Video Traces 47

5 SAM-based Traffic Generator and other Developed Tools 52

5.1 SAM-based Traffic Generator 53

5.2 RTP Traffic Generator 57
5.3 SAM-Based Video Trace Analyzer 59
5.4 Resource Allocation in Mobile WiMAX Networks 61
5.4.1 Scheduling Algorithms 63
5.4.2 Earliest Deadline First (EDF) 63
5.4.3 Deficit Round Robin (DRR) 63
5.4.4 Earliest Deadline First with Deficit Round Robin (EDF-DRR) 64
5.4.5 Scheduling Algorithms with Enforced Deadline 64
5.4.6 Performance Evaluation 64
5.4.7 Simulation Configurations 66
5.4.8 Simulation Results 67
6 Statistical, Modeling, and Prediction Analysis of HD Video Traces 71
6.1 Encoding YouTube HD Videos 73
6.2 Factor and Cluster Analysis of Video Traces 74
6.2.1 Principal Component Analysis 76
6.2.2 Cluster Analysis Using K-means Clustering 79
6.3 HD Video Traces Collection Modeling Results 82
6.3.1 AR Modeling 82
6.3.2 ARIMA Modeling using Automatic Approach 83
6.3.3 SAM Model 83
6.4 Modeling Results 83
6.5 Forecasting HD Video Traffic 86
7 SAM-based Dynamic Resource Allocation Scheme 91
7.1 Using SAM in Online Traffic Forecasting 94
7.1.1 Model Parameters Estimation Methods 95
7.1.2 Forecasting Using SAM 96
7.2 Determining Trend Changes 96

7.3 SAM-based DRA Scheme 99
7.4 Experimental Results 101
7.4.1 Comparison Results Using 54 AVC Video Traces 105
7.4.2 Comparison Results Using Long AVC Video Traces 107
7.4.3 Comparison Results Using Long SVC-TS/SVC-SS Video Traces 108
8 Summary 113
References 116
Curriculum Vitae 123

List of Tables

Table 2-1. WiMAX Application Classes [9] 12

Table 4-1 Model Identification Based on ACF and PACF Plots 29

Table 4-2 All-Frame Model, Composite Model, and SAM Model AIC Comparison Results 35

Table 4-3 Statistical Analysis of Long Video Traces 36

Table 4-4 SAM Model Parameters Values for Various Movies 36

Table 4-5 Statistical Comparisons Between SAM, and the Calculated Optimal Models 37

Table 4-6 Comparison Between SAM_{AVC} and Calculated Models 42

Table 4-7 Encoding Parameters for YUV Reference Video Sequences 44

Table 4-8 Statistical Characteristics of Few of the Modeled YUV Video Sequences 45

Table 4-9 Comparison Between SAM_{SVC} and The Calculated Models 49

Table 5-1 Performance Evaluation Parameters [55] 65

Table 5-2 System Throughput, Delay and Delay Jitter with Enforced Deadline (5 Flows in overload Scenario) 68

Table 5-3 System Throughput, Delay and Delay Jitter with Enforced Deadline (2 well-behaved Floes and one Ill-behaved Flow in overload Scenario) 68

Table 6-1 Encoding Parameters for The Selected YouTube Video Collection 73

Table 6-2 Range of Statistical Values for YouTube Video Collection 75

Table 6-3 Correlation Between The Selected Variables 76

Table 6-4 Estimated and Rotated Factor Loadings 78

Table 6-5 Clustering Results Using K-means Clustering 81

Table 6-6 Comparison between AR, ARIMA, and SAM Using AIC, MAE, MARE and RMSE 84

Table 6-7 SNR^{-1} Comparison between AR, ARIMA and SAM 88

Table 7-1 Estimation Methods Comparison Results 95

Table 7-2 Percentage of Improvement Between Estimation Methods 96

Table 7-3 QDBA Parameters 102

Table 7-4 Percentage of Improvement for Using SAM over VSA.....105
Table 7-5 Percentage of Increment for Queue Occupancy and Allocated Bandwidth.....105
Table 7-6 Percentage of Improvement for Using SAM over VSA for Long Video Traces.....107
Table 7-7 Percentage of Improvement for Using SAM over VSA for SVC-TS Videos.....109
Table 7-8 Percentage of Improvement for Using SAM over VSA for SVC-SS Videos.....109

List of Figures

Figure 1-1 An Example of Video Traces Variability with Reference to The Mean Value 3

Figure 2-1 WiMAX Network 7

Figure 2-2 Component Layers of a Model 8

Figure 2-3 Downlink Packet Scheduler 9

Figure 2-4 Uplink Packet Scheduler 10

Figure 3-1 MPEG4 Video Standards 15

Figure 3-2 MPEG Video Hierarchy 16

Figure 3-3 Difference Between I, P, and B-frames in Frame Size and Variability 17

Figure 3-4 Seasonal Characteristics of MPEG Video 18

Figure 3-5 GoP Structure in MPEG 19

Figure 3-6 Main Steps of Video Transmission Process 20

Figure 3-7 AVC Layer Hierarchy 21

Figure 3-8 Types of Video Coding Scalability 22

Figure 3-9 B-Frame Prediction Structure 24

Figure 4-1 Time Series Characteristics 26

Figure 4-2 Interpreting ACF and PACF Plots 28

Figure 4-3 Videos Samples Traces 33

Figure 4-4 Examining "News" Video Model Residuals 34

Figure 4-5 SAM Model Results: LOTR II Movie Trace 39

Figure 4-6 Seasonality in AVC Encoded Movies 40

Figure 4-7 Quantization Level Effects on Video Frames 42

Figure 4-8 SAM_{AVC} Comparison Results 44

Figure 4-9 SAM_{AVC} Results for "News" YUV Reference Trace 46

Figure 4-10 Seasonality in SVC Encoded Video (Star Wars IV) 48

Figure 4-11 SAM_{SVC} Results 50

Figure 5-1 Trace-driven versus Model-based Simulations 52

Figure 5-2 Different Accuracy Levels Corresponding to Different Numbers of AR Coefficients 54

Figure 5-3 Random Shocks Implementation in SAM Video Trace Generator 55

Figure 5-4 CDF Comparisons for Short, Medium and Long Video Traces 56

Figure 5-5 Implementation of SAM Video Traffic Generator using C#.NET 57

Figure 5-6 RTP Packetizing 57

Figure 5-7 RTP Generated Packets Using RTP Packetizer 58

Figure 5-8 RTP Generator and SAM generator Framework 59

Figure 5-9 SAM-based Video Trace Analyzer GUI 59

Figure 5-10 An Example of SAM Trace Analyzer Generated Comparison Plots 61

Figure 5-11 Simulation Topology 67

Figure 5-12 System Throughput (Five Video Flows in Overload Scenario) 69

Figure 5-13 System Throughput (Two Well-behaved Flows and One ill-behaved Flow in Overload Scenario) 70

Figure 6-1 Modeling, Analyzing, and Generation of Video Traces Processes 72

Figure 6-2 Scree Plot for the HD Video Collection Data based on the Eight selected Variables which Indicates Two Principal Components 77

Figure 6-3 Scatter Plot of Varimax Rotated Factors F1* and F2* in the Space of the Two Principal Components 78

Figure 6-4 Determining Number of Clusters using Scree Test and Hieratical Analysis 80

Figure 6-5 Distribution of Movie Groups over The Two Clusters 81

Figure 6-6 Graphical Comparisons Between AR, ARIMA, and SAM 86

Figure 6-7 Comparisons between AR, ARIMA, and SAM in their SNR^{-1} Values 88

Figure 6-8 Prediction Comparison between AR, ARIMA, and SAM 89

Figure 7-1 Dynamic Resource Allocation Scheme 91

Figure 7-2 Renegotiating Bandwidth Resources Upon Traffic Changes 92

Figure 7-3 Detecting Traffic Trend Using GoP Aggregation 98

Figure 7-4 Comparing Different GoP Aggregation Levels 99

Figure 7-5 SAM-based Traffic Prediction 100

Figure 7-6 SAM-based DRA Model 101

Figure 7-7 QDBA Algorithm 102

Figure 7-8 Average Queue Size against Different Utilization and Delay Requirements 106

Figure 7-9 SAM versus VSA Prediction Rate Comparison 106

Figure 7-10 Frames Autocorrelation when Using AVC, SVC-TS and SVC-SS Encodings 108

Figure 7-11 SAM versus VSA Prediction Rate Comparison for the Silence of the Lambs Video Trace 110

Figure 7-12 Meeting Delay Requirements in SAM-based DRA 111

List of Abbreviations

3G	Third Generation Mobile Networks
ACF	Autocorrelation Function
AIC	Akaike's Information Criterion
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
AVC	Advanced Video Codec
B-Frame	Bi-directionally Predicted Frame
BE	Best Effort Service
BPSK	Binary Phase Shift Keying
BS	Base Station
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable Length Coding
CBR	Constant Bit Rate
CDF	Cumulative Distribution Function
CDMA 2000	Code-Division Multiple Access 2000
CFA	Confirmatory Factor Analysis
CIF	Common Intermediate Format (352 by 288)
CPU	Central Processing Unit
CSS	Conditional Sum of Squares
CSS-ML	Hybrid Approach between ML and CSS
DCD	Downlink Channel Descriptor
DL	Downlink
DRA	Dynamic Resource Allocation
DRR	Deficit Round Robin
DTS	Decode Time Stamp
ECDF	Empirical CDF
EDF	Earliest Deadline First
EDF-DRR	Earliest Deadline First with Deficit Round Robin
EDGE	Enhanced Data Rates for Global Evolution
EFA	Exploratory Factor Analysis
ertPS	Extended Real Time Polling Service
ES	Elementary Stream
FARIMA	Fractional ARIMA
FCH	Frame Control Header
FDD	Frequency Division Duplexing

FEC	Forward Error Correction
FSA	Fixed Step-Size Algorithm
FTP	File Transfer Protocol
GoP	Group of Pictures
GUI	Graphical User Interface
HD	High Definition
HDTV	High Definition TV
I-Frame	Intra Frame
IDR	Instantaneous Decoder Refresh
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPTV	IP Television
ISP	Internet Service Provider
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
LMS	Least Mean Square
LOTR	The Lord of The Rings
LQ	Listening Quality
LRD	Long Range Dependence
LTE	Long Term Evolution
MA	Moving Average
MAC	Medium Access Layer
MAE	Mean Absolute Error
MARE	Mean Absolute Relative Error
MCS	Modulation and Coding Schemes
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
MobileTV	Mobile Television
MPEG	Moving Picture Expert Group
MS	Mobile Station
MSE	Mean Square Error
MST	Maximum Sustained Rate
MTU	Maximum Transmission Unit
NAL	Network Adaptation Layer
NMSE	Normalized Mean Square Error
NN	Neural Network
NRM	Network Resource Controller
nrtPS	Non Real Time Polling Service
NS-2	Network Simulator 2
NTSC	National Television Standards Committee
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency-division Multiple Access

P-Frame	Predicted Frame
PACF	Partial Autocorrelation Function
PAL	Phase Alternating Line
PCA	Principal Component Analysis
PES	Packetized Elementary Stream
PSNR	Peak Signal to Noise Ratio
PTS	Presentation Time Stamp
PUSC	Partially Used Sub-Channelization
QAM	Quadrature Amplitude Modulation
QCIF	Quarter Common Intermediate Format (176 by 144)
QDBA	QoS-guaranteed Dynamic Bandwidth Allocation
QoE	Quality of Experience
QoS	Quality of Service
QP	Quantization Parameters
QQ	Quantile-quantile
RMSE	Root Mean Square Error
RTCP-XR	RTP Control Protocol Extended Reports
RTP	Real Time Protocol
rtPS	Real Time Polling Service
SAM	Simplified Seasonal ARIMA Model
SARIMA	Seasonal Autoregressive Integrated Moving Average
SI	Switching I
SNR	Signal To Noise Ratio
SNR^{-1}	Inverse of Signal to Noise Ratio
SP	Switching P
SRC	Stream Resource Controller
SRD	Short Range Dependence
SS	Size Scalable
STD	Scene Type Descriptor
SVC	Scalable Video Codec
TDD	Time Division Duplexing
TS	Time Scalable
UCD	Uplink Channel Descriptor
UDP	User Datagram Protocol
UGS	Unsolicited Grant Scheme
UL	Uplink
UPQ	User Perceived Quality
VBR	Variable Bit Rate
VCEG	Video Coding Expert Group
VCL	Video Coding Layer
VOD	Video Object Descriptor
VoIP	Voice over IP

VQEG	Video Quality Expert Group
VSA	Variable Step-Size Algorithm
WiMAX	Worldwide Interoperability for Microwave Access
YUV	One Luma (Y) and two Chrominance (UV) components

List of Symbols

AR(p)	Autoregressive component of order p
B	Backward operator
d	Differencing order
D	Seasonal differencing order
e_t	Error term or component at time or index t
F	Number of frames to be transmitted per second
$G_{0(i)}$	Actual size of the GoP at a time slot i
$G_{p(i)}$	Predicted size of the GoP at time slot i
I(t)	Irregularities or random component of time series
M	Number of frames per GoP
MA(q)	Moving Average component of order q
N	Number of Frames in Video Trace
p	The order of the autoregressive component
P	Seasonal autoregressive order
q	Moving Average order
Q	Seasonal Moving average order
$Q_{(i)}$	Queue size at the end of time slot i
R	Current transmission rate
RSS	Residual Sum of Squares
r_k	Autocorrelation coefficient for lag k
$R_{p(i)}$	Predicted bandwidth at time slot i
$R_{t(i)}$	Required bandwidth at time slot i for delay constraints
S(t)	Seasonal component of time series
T	Maximum allowed delay
Tr(t)	Trend component of time series
$\mu_{(i)}$	Allocated bandwidth at time slot i
w_i	Throughput for the i-th MS.
$\overline{x}$	Mean frame size
$\overline{\hat{x}}$	Estimated Mean frame size corresponding to the model
x_t	Time series value at time t
$\hat{x}_t$	Estimated frame size at time t
φ	Autoregressive coefficient
θ	Moving average coefficient
∇	Differencing operator
Φ	Seasonal Autoregressive coefficient
Θ	Seasonal Moving average coefficient

λ_i	Eigenvalue for the i-th component
ρ	Required bandwidth utilization

Chapter 1

1 Introduction

Video streaming traffic has been surging in the last few years, which has resulted in an increase of its Internet traffic share on a daily basis. The introduction of web based video applications such as video-based societies (represented mainly in video blogging), IP-based television transmissions (represented in IPTV and MobileTV), and the rising of video on-demand services that stream selected videos and TV shows over the Internet have been driving network researchers with a high motivation to seek better solutions to accommodate the continuing growth of user demands and to meet their expectations. The spread of broadband wireless networks, as represented in WiMAX and LTE technologies, have tremendous impact on the future of video streaming over the wireless medium. Such introduction of high bandwidth wireless networks allows better support for streaming video media on the go.

There is a crucial need to achieve a better understanding of the characteristics of video traffic and their effects over both wireless and wired broadband networks. The urgency to have such understanding has been emphasized with the advent of high definition (HD) videos, as they require more resources, and thus a better network support. Researchers aim to develop better resource allocation mechanisms to allow better utilization of the scarce network resources, to provide a superior control to manage different levels of quality of service (QoS), and to allow a better support for the latest demanding applications.

Web-based video streaming websites facilitate means to promising opportunities to distribute digital video contents to millions of people. Websites like YouTube [1] and Vimeo [2] are now considered among the top daily accessed websites for many Internet users. Such websites now account for 27 percent of Internet traffic, rising from 13 percent in just one year [3]. This surge in traffic share can be explained by considering the latest surveys, where the percentage of U.S. Internet users watching streaming videos have increased from 81% to 84.4%, and their average video watching time spent per month increased from 8.3 to 10.8 hours/month in just three months [4, 5].

Additionally, several websites, like Hulu [6] and Netflix [7], have started to offer access to TV shows and selected movies, which has increased the reliance of the daily Internet users on such websites. Such reliance is accompanied by an increase in the Internet users' expectations of the received quality of experience (QoE). All these reasons inspire network researchers to put more emphasis on handling such demanding traffic. Resource allocation and admission control depend on their ability to predict and manage the increasing volume of video stream demands. The need to analyze such high-demand traffic, and possibly to model them, is essential to develop better traffic engineering schemes to provide better quality of service (QoS) support.

Workload characterization is a key step in any performance analysis [8]. Incorrect workload can lead to meaningless results. In both wired and wireless networks, it is essential to have an accurate model of the possible traffic workloads. Additionally, to accurately evaluate the performance of new broadband access technologies, such as Mobile WiMAX, we need to be able to analyze the possible applications that will be used on these technologies. Streaming video is expected to be one of the key applications on future broadband access networks. Mobile video has already started appearing as a key service offered from many cellular companies.

Simulation provides an easy means to analyze different resource allocation strategies. While simulation environments like NS-2 provide the means to create the necessary network topology, there is still a need to provide an accurate workload for the test scenarios. The workload should represent the real world traffic accurately and should be easy to administer and adjust to different simulation conditions.

Internet video traffic is generally considered as variable bit rate (VBR) traffic, and exhibits a high degree of variability in the trace compared to constant bit rate (CBR). This means that a simple representation of the video traces using their mean values is deemed to be inaccurate. As Figure 1 shows, the mean value of the frame size, shown with a thick-red line, does not represent the movie trace statistical characteristics.

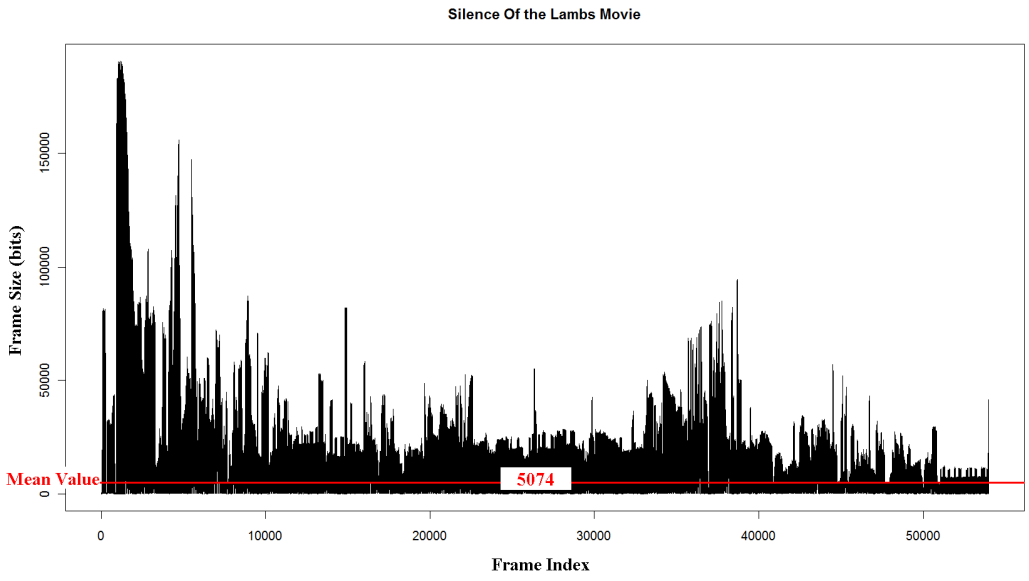


Figure 1-1 An Example of Video Traces Variability with Reference to The Mean Value

The exhibited variability of video traces results from the continuous changes in both motion and texture levels throughout the movie. The variance is usually harmonic within a single movie scene. As the movie progresses, the variance changes from one scene to another. This kind of behavior is what makes video traffic unique and challenging at the same time.

1.1 Dissertation Main Contributions

In this dissertation, we describe our methodology for researching the characteristics of video traffic over both wireless and wired networks. We also discuss the steps to model a variety of video traces encoded with the most common encoding standards. We target three of the latest and most used standards in video encoding: MPEG4-Part2, MPEG4-Part10/Advanced Video Codec (AVC) also known as H.264, and AVC extension to support scalability, viz., scalable video codec (SVC). As a result of our research, we provide an accurate and a practical source model of videos traces encoded for both wireless and wired networks.

One of the main design goals of this model is to achieve a general mathematical approach that is capable of representing different movie traces encoded with the most used video standards. Such model should provide a unified method to model movies traces with

different lengths, encoding settings, and standards. Furthermore, our model is simple for fellow researchers to use and configure. It does not require having a significant statistical background either to utilize it or configure it to their specific needs; it is also easy to implement, and it does not require a lot of operational resources.

In addition, we show the results of using our model-based prediction mechanism's ability to predict successive video traffic patterns, which provides better support for both admission control and resource allocation. This model can be easily adapted for real time scenarios because of its simplicity and accuracy. Its ability to predict subsequent video frames allows us to implement a dynamic bandwidth allocation scheme to provide better support for real time/live video traffic.

As a result of our prediction analysis, we provide a description of our model-based dynamic resource allocation (DRA) scheme. We show through our analysis results using our presented scheme that we can achieve significant improvement in utilizing the network resources.

As one of this dissertation's goals, we try to shed light on the inter-correlation between video frames and their unique statistical characteristics. We performed multivariate factor analysis using principal component analysis (PCA), exploratory factor analysis (EFA), and confirmatory factor analysis (CFA). In addition, we aim to better identify video traces by grouping them, depending on their statistical characteristics, into clusters. Therefore, we performed a cluster analysis on the our video collection using *k-means* clustering.

Due to the limited availability of video traces, we also describe our approach of collecting and encoding the large collection of HD video traces used in our verification and analysis steps. The limited availability of such traces is due to the resource-consuming nature of the process needed to produce representative and informative video traces. As a part of our contributions, we created a library of HD video traces available to fellow researchers. This video traces collection can be used not only to verify our results, but also to provide the means for the research community to develop and evaluate their own contributions. In fact, all our contributions are to be made available to the research community through our website, including our results and the developed code and tools.

Our contributions aim to provide the research community with better understanding of the key attributes in video streaming over both wired and wireless networks. Our general and unified approach to model videos encoded with the most common and latest video codecs allows it to be used in real time applications to deliver better quality of service control. The publicly available HD video traces library collection both allows the researchers to perform further statistical analysis, and to be used as a reference point for future video related studies.

1.2 Organization

We start this dissertation by giving background information and a brief introduction to the key issues of the promising broadband wireless network: WiMAX. Chapter 2 gives an introduction to WiMAX network and its unique characteristics. In addition, it explains the some of the motivations behind our research.

As a part of developing testing scenarios to evaluate the network capacity, we were confronted with the need of several video traces in order to provide an accurate representation of the possible traffic workloads. A limited number of traces are available publicly, and these traces do not represent all video genres. As a result, we started developing our mathematical video source model.

The moving picture expert group (MPEG) video codecs family is the current *de facto* standard for encoding videos. In Chapter 3 we provide a simple introduction to the encoding techniques used in the latest codec standards. We cover the following standards: MPEG4-Part2, MPEG4-Part 10, also known as advanced video codec (AVC) or H.264, and the latest extension to AVC to support video scalability: scalable video codec (SVC).

Chapter 4 provides an introduction to times series analysis. In this chapter, we describe the basis of our statistical model, and we illustrate the significance of our approach to overcome the implementation hurdles usually associated with time series modeling. This chapter also shows the results of modeling video traces encoded using the three video standards mentioned earlier. In addition, we show the importance of our approach by comparing it to other modeling approaches.

Chapter 5 describes the implementation details of several developed tools including our model-based traffic generator, and it demonstrates the results achieved by using the generated traffic traces to compare different scheduling methods for Mobile WiMAX networks.

In Chapter 6 we describe our methodology of choosing, collecting and encoding over 50 HD video traces that represent a wide variety of statistical characteristics. Based on our video collection, we show the results of performing a full statistical analysis. We also investigate the main video attributes used to represent a video trace by performing a principle component analysis. In addition, we show the results of performing a cluster analysis on our video collection. We also compare our model to two of the most used time series models both in terms of their modeling and their prediction accuracy. The two chosen models are selected to represent different approaches to model video traces using time series analysis.

In Chapter 7 we illustrate the details of our dynamic resource allocation (DRA) scheme. Comparing our scheme to one of the latest schemes, we show that ours provides both a better utilization of the network resources and a better support for the required quality of service levels associated with live video streams. Finally, we discuss and summarize our main contributions and their impact in the research field.

Chapter 2

2 Introduction to WiMAX Networks

IEEE 802.16 standard, or as commonly referred to as WiMAX, is a set of telecommunications technology standards aimed at providing wireless access over long distances in a variety of ways - from point-to-point links to full mobile cellular type access. It covers a metropolitan area of several kilometers and is also called WirelessMAN. WiMAX[2] is an acronym to Worldwide Interoperability for Microwave Access and its name was created by WiMAX Forum which was formed in 2001 to promote interpretability among 802.16 products. Theoretically, a WiMAX base station can provide broadband wireless access in range up to 30 miles (50 kms) for fixed stations and 3 to 10 miles (5 to 15 kms) for mobile stations with a maximum data rate of up to 70 Mbps [9, 10] compared to 802.11a with 54 Mbps up to several hundred meters, Enhanced Data Rates for Global Evolution (EDGE) with 384 kbps to a few kms, or Code-Division Multiple Access 2000 (CDMA2000) with 2 Mbps for a few kms.

As shown in Figure 2.1, the mobility and the wide coverage of WiMAX allow it to service different levels of client accesses. WiMAX base station acts as a central distribution and control point.

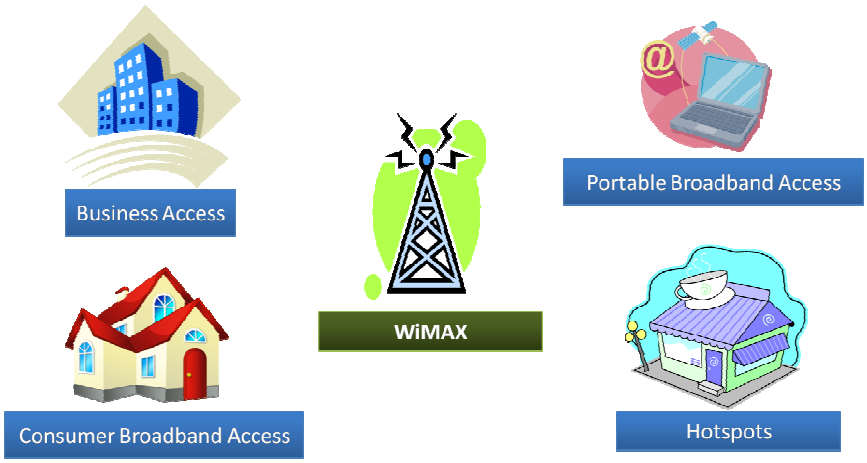


Figure 2-1 WiMAX Network

[2] WiMAX, Mobile WiMAX, Fixed WiMAX, WiMAX Forum, WiMAX Certified, WiMAX Forum Certified, the WiMAX Forum logo, and the WiMAX Forum Certified logo are trademarks of the WiMAX Forum.

It is important to mention that WiMAX interoperability specifications as implemented by WiMAX Forum members are a subset of the IEEE 802.16e standard. For instance, 802.16e standards allows different values for the duration of the orthogonal frequency-division multiple access (OFDMA) frame, while WiMAX Forum has selected only one value. These forced decisions are important to allow different equipments from different vendors to operate together and to allow a meaningful comparison between the different products.

2.1 WiMAX System Model

WiMAX Forum Working Group has been working on a system level simulation methodology to allow an accurate simulation of WiMAX networks [12]. The simulation methodology is not restricted to one simulation platform; it can be used with Network Simulator 2 (NS-2), or any other variant. A system level model should implement all the system levels, as shown in Figure 2-2. A link level model, for example, is only concerned about the transmission mechanism between the base station and the users. Such model will focus on the physical layer implementation and the surface distribution of the cell towers or base stations. On the other hand, a system level model is concerned about all the layers and their interactions.

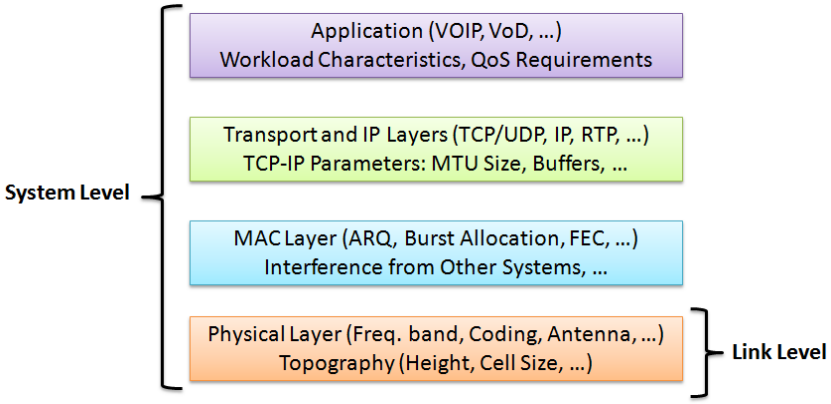


Figure 2-2 Component Layers of a Model

In our study we focus on the implementation of the application layer that includes the study of the network workload characteristics and the associated QoS requirements. Such study will take into consideration the inevitable interaction between the different layers.

IEEE 802.16e does not specify a scheduling algorithm for the medium access control (MAC) layer. It left the decision to the implementers to select the best scheduling mechanism. The chosen MAC scheduler id expected to be able to provide the best utilization of the available and scarce network resources. The scheduler should take into consideration both the network conditions and the data flows associated QoS parameters, if any. The scheduling service is a part of both uplink (from subscriber/mobile station to base station) and downlink (from base station to subscriber/mobile station) traffic processing.

Figures 2-3 and 2-4 shows the mechanism associated with downlink and uplink packet scheduling in WiMAX. The downlink scheduler needs to maintain a per-flow queue to meet the different QoS requirements for each of them. The uplink scheduler uses a similar mechanism, in addition it keeps track of the request/grant status of each uplink flow.

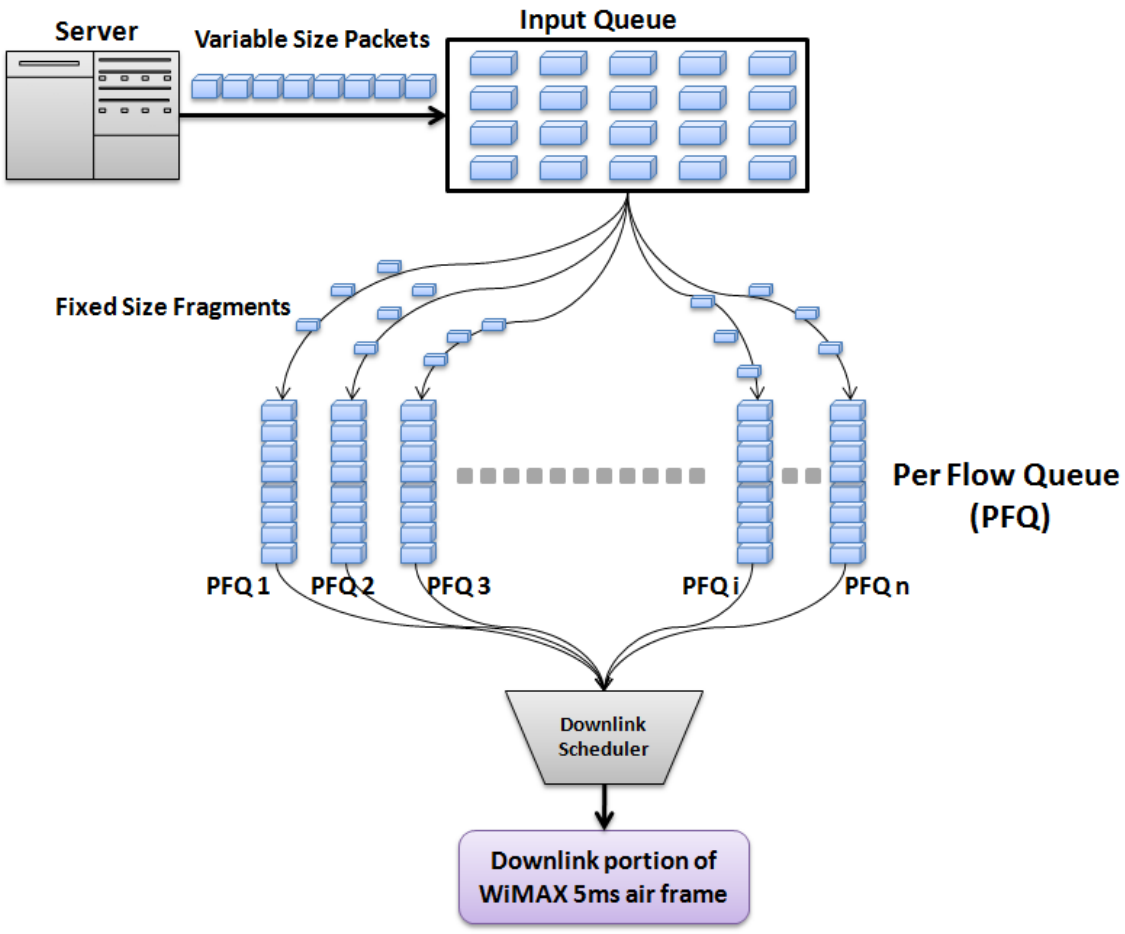


Figure 2-3 Downlink Packet Scheduler

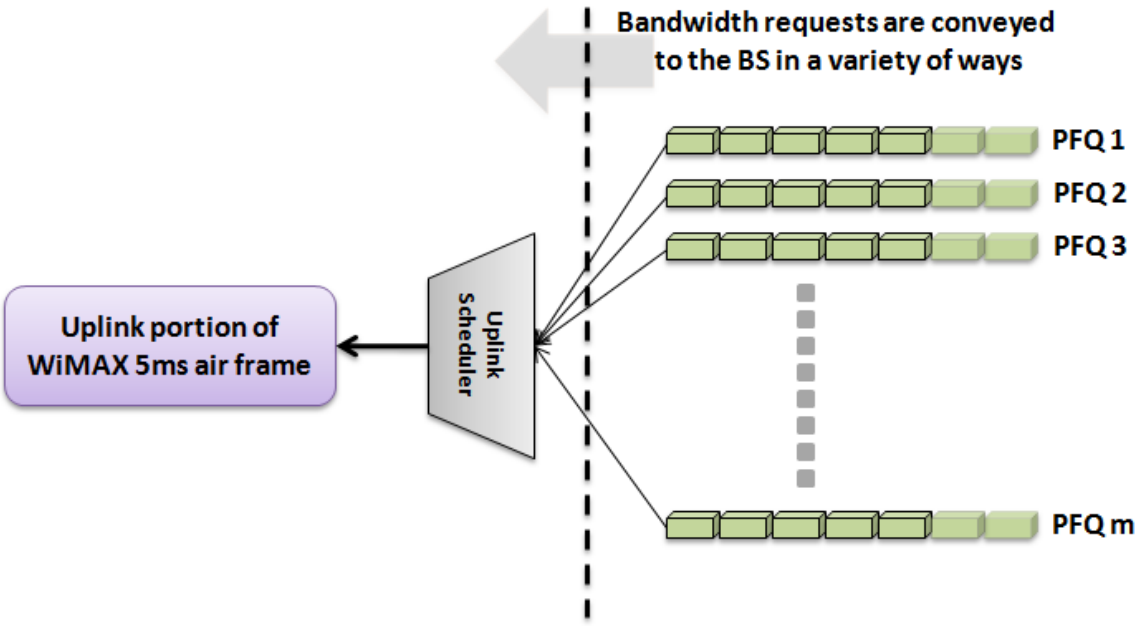


Figure 2-4 Uplink Packet Scheduler

As mentioned before, there is a strong correlation between the different system layers. For example, in the application layer the video stream is represented as a series of video frames, which is mapped on the network layer as a series of IP packets. An application layer model, or a user-level model, is more complicated than network level model, but it provides more information about user's quality of experience (QoE) and presents a more accurate simulation results [10, 11].

2.2 WiMAX Quality of Service (QoS) Classes

In terms of guaranteed services, WiMAX includes several Quality of Service (QoS) mechanisms at the MAC (Media Access Control) layer. Typically, the QoS support in wireless networks is much more challenging than that in wired networks because the characteristics of the wireless link are highly variable and unpredictable both on a time-dependent basis and a location dependent basis. With longer distances, multipath and fading effects are also put into consideration. Request/Grant mechanism is used for mobile stations (MSs) to access the media with a centralized control at base stations (BSs). Therefore, MSs are not allowed to access the wireless media unless they register and request the bandwidth allocations from the BS first, except for certain time slots reserved specifically for contention-based access.

IEEE 802.16, or WiMAX, defines five QoS service classes: Unsolicited Grant Scheme (UGS), Extended Real Time Polling Service (ertPS), Real Time Polling Service (rtPS), Non Real Time Polling Service (nrtPS) and Best Effort Service (BE). Each of these has its own QoS parameters such as minimum throughput requirement and delay/jitter constraints. We explain in the following paragraphs the main differences between these QoS classes.

UGS: This service class provides a fixed periodic bandwidth allocation. Once the connection is setup, there is no need to send any other requests. This service is designed for constant bit rate (CBR) real-time traffic such as E1/T1 circuit emulation. The main QoS parameters are maximum sustained rate (MST), maximum latency and tolerated jitter (the maximum delay variation).

ertPS: This service is designed to support VoIP with silence suppression. No traffic is sent during silent periods. ertPS service is similar to UGS in that the base station allocates the maximum sustained rate in active mode, but no bandwidth is allocated during the silent period. There is a need to have the BS poll the mobile station during the silent period to determine if the silent period has ended. The QoS parameters are the same as those in UGS.

rtPS: This service class is for variable bit rate (VBR) real-time traffic such as MPEG compressed video. Unlike UGS, rtPS bandwidth requirements vary and so the BS needs to regularly poll each MS to determine what allocations need to be made. The QoS parameters are similar to the UGS but minimum reserved traffic rate and maximum sustained traffic rate need to be specified separately. For UGS and ertPS services, these two parameters are the same, if present.

nrtPS: This service class is for non-real-time VBR traffic with no delay guarantee. Only minimum rate is guaranteed. File Transfer Protocol (FTP) traffic is an example of applications using this service class.

BE: Most of data traffic falls into this category. This service class guarantees neither delay nor throughput. The bandwidth will be granted to the MS if and only if there is a left-over bandwidth from other classes. In practice most implementations allow specifying minimum reserved traffic rate and maximum sustained traffic rate even for this class.

Note that for non-real-time traffic, traffic priority is also one of the QoS parameters that can differentiate among different connections or subscribers within the same service class. Consider bandwidth request mechanisms for uplink. UGS, ertPS and rtPS are real-time traffic. UGS has a static allocation. ertPS is a combination of UGS and rtPS. Both UGS and ertPS can reserve the bandwidth during setup. Unlike UGS, ertPS allows all kinds of bandwidth request including contention resolution. rtPS cannot participate in contention resolution. For other traffic classes (non real-time traffic), nrtPS and BE, several types of bandwidth requests are allowed such as piggybacking, bandwidth stealing, unicast polling and contention resolution.

2.3 WiMAX Application Classes

WiMAX standard classifies applications into five categories as shown in Table 1. Each application class has its own characteristics such as the bandwidth, latency and jitter constraints in order to assure a good quality of user experience [9].

Table 2-1. WiMAX Application Classes [9]

Classes	Applications	Bandwidth Guidelines		Latency Guidelines		Jitter Guidelines		QoS Classes
1	Multiplayer Interactive Gaming.	Low	50 kbps	Low	< 25ms	N/A		rtPS , UGS
2	VoIP and video Conferencing	Low	32-64 kbps	Low	< 150ms	Low	< 50ms	UGS, ertPS
3	Streaming Media	Low to High	5kbps to 2Mbps	N/A		Low	< 100ms	rtPS
4	Web Browsing and Instant Messaging	Moderate	10kbps to 2 Mbps	N/A		N/A		nrtPS, BE
5	Media Content Downloads	High	> 2Mbps	N/A		N/A		nrtPS, BE

As can be noticed streaming media, including streaming video, is classified under rtPS QoS class. This class requires communication between the base station and the mobile stations to determine the minimum and maximum transfer rates to be allocated. While this arrangement can work for pre-encoded and analyzed video traces like videos and TV shows, if we ignore the obvious overhead, it does not work for live stream media like MobileTV.

2.4 Quality of Experience Assessment Methods in Video Applications

Quality of service is often represented by the level of degradation of the received signal at the application level when compared to the original transmitted signal. Quality of service does not necessarily indicate the user satisfaction level of the provided service. What might be acceptable at the application level may cause inconveniences at the user level. Thus, quality of experience (QoE) is used to indicate the service degree of gratification at the user's level.

The main objective of the quality of experience (QoE) assessment methods through evaluating the network performance of multimedia systems over IP networks is to correlate the user perceived quality (UPQ) to the measured degradation of the received signal, or to the current network condition.

There are two main approaches for evaluating multimedia systems: objective and subjective methods. Subjective methods aim to use the help of human subjects to measure the quality of the received media samples. Objective methods measure the communication link quality and map it to a representative index value of users' experience.

Video quality of service (QoS) assurance is more demanding than voice and audio because it incorporates more factors that need to be monitored and analyzed. In addition to that, video media users tend to be less forgiving about distortions in the received video.

One of the most common video quality problems that are caused by different factors in the video transmission system is video jerkiness. Video jerkiness is when video frames are shown in a discontinuous manner, which can be caused by encoding, network, or synchronization problems. Video blur and video noise are usually originated from defected camera, or encoder/decoder related problems. Partial and complete video blackouts are symptoms of network loss and lack of sufficient bandwidth.

Another common problem is video blockiness, where visible blocks are scattered across the video screen. An extreme case of blockiness is represented by video distortion. Video distortion results in segmented video frames. The causes of these behaviors can be either

encoder/transcoder problems, packets loss, or simply because there is no enough bandwidth to support the transmitted video [108].

Because of the structure of the encoding mechanism of video frames, some video frames have a higher impact on the final delivered frame than others. Reference frames are considered more important to assure the quality of the transmitted video than sub and/or predicted frames.

These are some of the changes that are to be considered to have an accurate assessment models for multimedia transmission over WiMAX networks. Additional factors should be considered to reflect the network design and its impact on the received signal strength. Though subjective methods are not suitable for testing real time multimedia traffic, they provide a good reference point for quality performance evaluation. Subjective methods should be adjusted as well to adapt to the medium changes. Since WiMAX supports mobility, additional testing environment factors should also be considered. The speed of the moving vehicle, the density of the broadcasting cells in the area, and the hands-off smoothness between cells are some examples of such factors.

In order to have better allocation mechanisms for live video streaming, a dynamic allocation method is preferable to avoid large queues, large delays and excessive loses. Additionally, such models will help in network simulations, where new technologies and research ideas need to be evaluated and tested for broadband wireless networks. These evaluations takes into consideration the quality of service that can be translated to represent the quality of experience perceived at the receiver side.

The developed source model should be able to capture the statistical characteristics of videos accurately. In addition, it should provide a simple and a general approach to handle the different encoding standards and settings. Such model will be the base for generating video traffic to be used in different simulation scenarios. It will also be the first step to develop a dynamic resource allocation scheme to guarantee the quality of service associated with the highly-demanding video traffic. As a first step we investigate the main characteristics of the most used video encoding standards. In the next chapter we will discuss the key features in MPEG video encoding, and the different standards that belong to the same codec family.

Chapter 3

3 MPEG Video Standards

In this chapter, we explain the fundamentals of MPEG (Moving Picture Expert Group) encoding. This basic introduction aims to give the reader the necessary background to help justify our modeling approaches.

MPEG is a collection of standards used to code both audio and video information. There are four main MPEG families: MPEG-1 was the first video compression standard. Many of the concepts used in later MPEG standards were introduced in MPEG-1. This includes MP3 encoding format for audio. MPEG-2 is used in TV broadcast and most DVD encoding. MPEG-3 was originally targeting HDTV (High Definition TV) broadcast but it was abandoned. MPEG-4 is a collection of standards for compressing audio and visual digital data.

We shall focus in this dissertation on the still-evolving and most common family of standards: MPEG-4. The three main standards we consider, as shown in Figure 3-1, are: MPEG-4 Part2, MPEG-4 Part10, and its scalable extension. MPEG-4 Part2, also the basis for Xvid and Divx codecs, which is designed for low bit rate videos like web streaming media, and conventional videophone [13]. MPEG-4 Part10, also known as advanced video codec (AVC), also known as H.264 standard, is the current standard for high definition videos, and is used by various high definition applications and media like: YouTube high definition video streams, iTunes store, and Blue-ray discs.

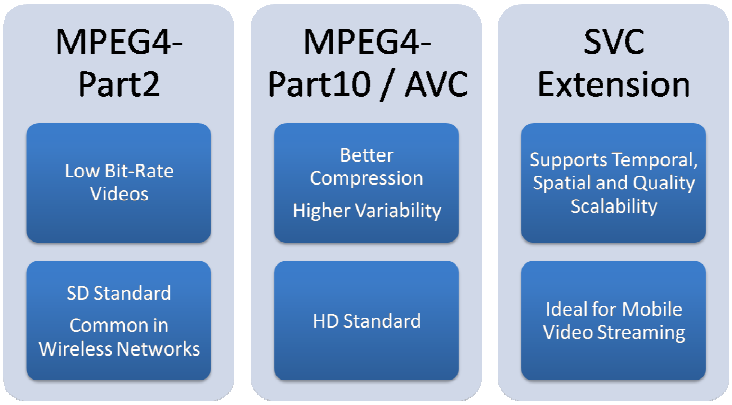


Figure 3-1 MPEG4 Video Standards

The third standard is the scalable video codec (SVC) extension which supports different types of scalability in the transmitted video stream. A subset of the transmitted video stream represents a lower temporal or spatial resolution, or even a lower video quality signal. Such arrangement is hierarchal in nature that consists of a base layer, with the lowest video resolution or quality level, and a series of enhancement layers. This allows different video inhomogeneous video clients with different decoding and network resources to receive the same video stream and decode the level of details accordingly to their capabilities. This feature is important for multicasting with the need to transmit different levels of the video stream, which is especially essential for limited resources networks like mobile networks.

3.1 MPEG Encoding Basics

MPEG uses a hierarchy of layers to represent video scenes. The hierarchy, as shown in Figure 3-2, consists of 6 different layers. Block layer is a group of pixels (8x8 pixels) that holds the visual information. To allow more efficient computations, four blocks (16x16 pixels) are grouped together to form a *macroblock*. A single row of macroblocks in a video frame is called a slice. These slices are then grouped to form a video frame or a picture.

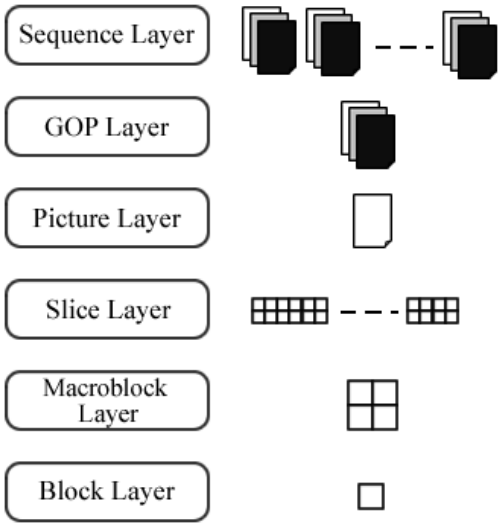


Figure 3-2 MPEG Video Hierarchy

For compression, successive video frames are considered together as a group of pictures (GoP) that represents an independent unit in the video scene. GoP sizes vary depending on the encoding options, increasing GoP size results in a lower bit rate and a better

compression rate, but it also results in a less robust compression. Sequence layer is comprised of a sequence of GoPs. A sequence layer can be thought of as a video scene.

There are three types of compressed video frames: Intra-coded Frames or I-frames, Predicted frames or P-frames, and Bi-directional predicted frames or B-frames. These frames are divided into two main groups: intra-frames and inter-frames, depending on the encoding process used to compress them.

An Intra-frame or I frame represents a reference frame and it is compressed independently. No information from other frames is used in the compression. Therefore, this frame can be decompressed even if other frames in GoP are lost. Predicted or P-frames result from encoding a video frame by its difference from the prediction, based on previous I frame or P-frame. Bi-directionally predicted or B-frames result from encoding a video frame using the difference from its prediction using both the previous I or P frame, and the next I or P frame. P and B-frames belong to Inter-frames group and are considerably smaller in size than I frames. But they do require larger buffers to accommodate the backward and/or forward predictions. Figure 3-3 below shows the difference between I, P, and B-frames in both variability and frames sizes. As we can notice, I-frames are larger in size, and they are less variable than P and B-frames.

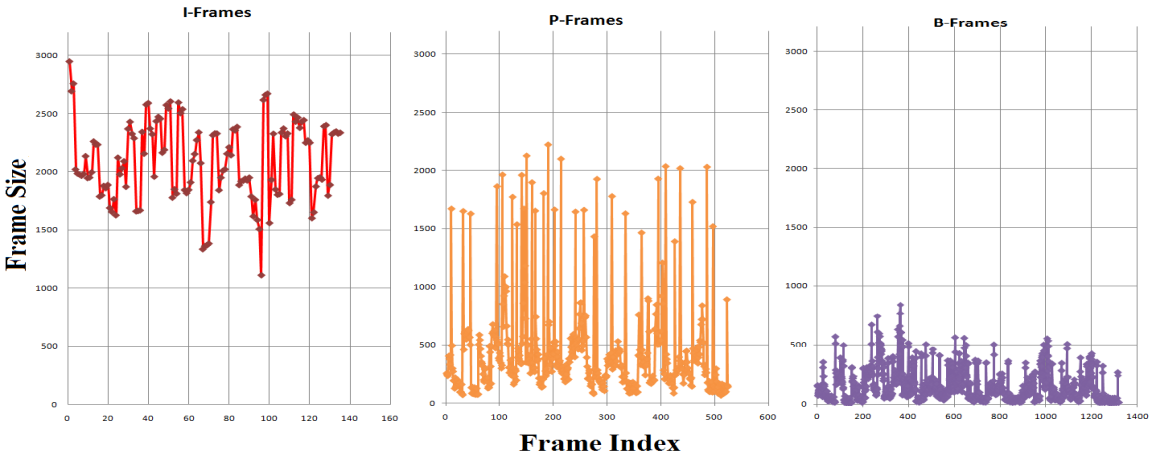


Figure 3-3 Difference Between I, P, and B-frames in Frame Size and Variability

As shown in the Figure 3-4, the pattern of video frames is repeated every "s" rames where "s" is the Group of Pictures (GoP) size. This observation has led us to consider modeling the video traces using time series analysis methods. Time series analysis, as we will discuss

more in details in Chapter 4, is concerned about analyzing and modeling data that exhibit temporal characteristics.

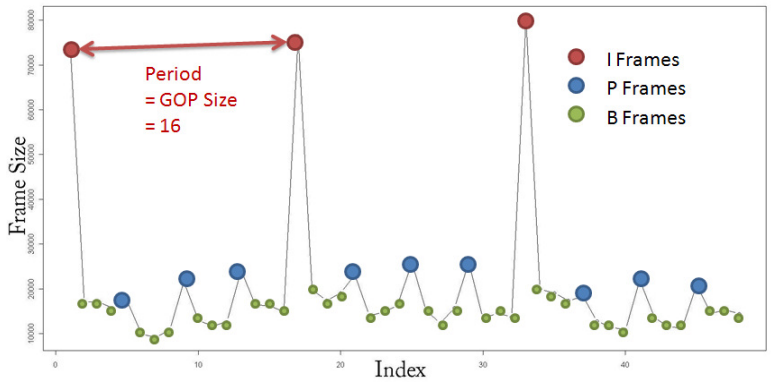


Figure 3-4 Seasonal Characteristics of MPEG Video

Because of the complexity of the obvious MPEG video patterns, other researchers have pointed out that a better approach to the problem is to use a multiplexed model or what is also known as a composite IPB model [14, 15]. In this approach, the video frame sequence is divided in to three sequences of I, P, and B type frames and each sequence is modeled separately. During generation, three streams are generated and then combined to form the final the MPEG video trace. The individual models representing each frame stream are simpler than a single "all-frames" model. Though this approach produces slightly better results than the one or all-frames model, each movie trace needs to be analyzed three times in order to produce the needed traces. In addition to that, having three models is hard to manage in real time application as in dynamic allocation processes. Our approach is to find one simple model to represent the entire video frames pattern regardless of the encoding settings, or movie characteristics.

GoP is usually presented as a sequence of frames starting with an I-frame then followed by a number of P and B-frames. The most commonly used GoP sizes are 15 for National Television Standards Committee (NTSC) systems used in North America, and 12 for Phase Alternating Line (PAL) systems used in Europe and many other countries. The most common maximum size allowed for GoP is: 18 for NTSC and 15 for PAL. As shown in Figure 3-5, video scenes are composed of a series or a sequence of GoPs, thus called a Sequence Layer. One of the common patterns of GoP is G12B2, which means a GoP of size

12 and 2 B-frames between each successive I or P frames. The encoding order for this sequence is: IBBPBBPBBPBB.

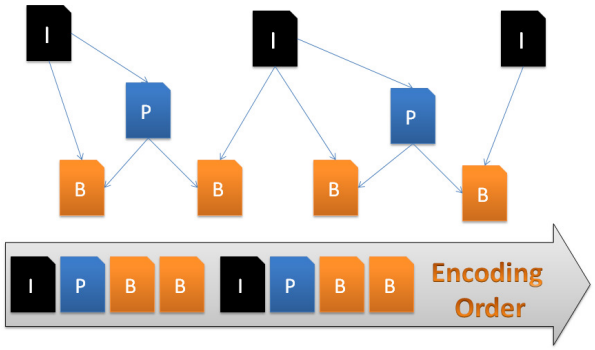


Figure 3-5 GoP Structure in MPEG

It is worth mentioning that to ensure that the video frames are presented or displayed correctly to the client, their presentation order information is sent alongside with the decoding order. MPEG, as variable bit rate traffic, exhibits the presence of two frame-dependences: short range dependence (SRD) because of the inter-coding characteristic presented by B and P frames, and long range dependence (LRD) because of the intra-coding characteristics. A good model has to address and present both dependencies. It is important for these relationships to be captured in any valid video traffic model. Please refer to [16-18] for more background information on MPEG compression.

Video transmission process consists of five main stages: pre-processing, encoding, transmitting/storing, decoding, and finally post processing and error recovery stage as shown in Figure 3-6. Elementary streams (ES) are the raw output of the video encoder. Each video and audio stream is considered an elementary stream. At least two elementary streams are necessary to have a video with both image and sound. By synchronizing the audio and video streams the final desired stream is achieved.

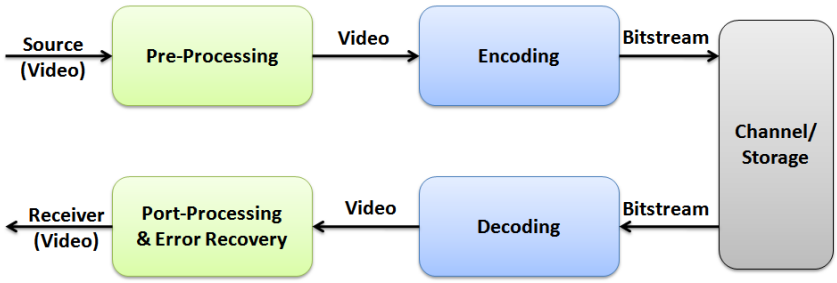


Figure 3-6 Main Steps of Video Transmission Process

Elementary streams are divided further into elementary stream packets to form what is known as packetized elementary stream (PES). PES packets have variable lengths that can reach hundreds of kilobytes. Inside the PES there are two timing stamps used for video streams: presentation time stamp (PTS), and decode time stamp (DTS). As their names indicate, PTS defines the time the frame should be displayed to the client, and DTS defines the time the frame should be decoded by the video decoder. These time stamps are used to synchronize the video stream with the audio stream. In order to represent video stream before being encapsulated for different storage or transmission mediums, modeling should be performed on the video's elementary stream or bitstream.

3.2 MPEG-4 Part10 / AVC Standard

MPEG-4 Part 10 standard or advanced video codec (AVC) is a joint result of ISO/IEC MPEG and the ITU-T video coding expert group (VCEG). AVC standard was completed in March 2003, and then approved by ITU-T in May 2003. AVC provides enhanced tools to improve the compression efficiency up to 50% over MPEG-2, and up to 30% compared with MPEG-4 Part2 codec [18], and better support of interactive and non-interactive video applications. Such improvements come with increased complexity in the design of the decoder up to two times than its predecessor MPEG4 simple profile decoder.

AVC standard introduced several improvements in the design over its predecessors, including the fact that it allows storing multiple frames in the decoder memory as opposed to a single frame in previous standards. This feature allows the decoder to reference multiple frames in the decoding process. This enables the prediction of the video frame using previous frames outside structure of one GoP. Such coding structure is also referred as open GoP, because there is no need for I-frames to be recurring on fixed space as in the previous

standards. Although this feature increases compressing efficiency, misusing it may results in worsening the user's experience. As we described before, the video stream can be accessed randomly only at I reference frames. To enhance the user's experience and allow better seeking abilities, AVC uses instantaneous decoder refresh (IDR) frames [19]. The presence of these frames indicates to the decoder that no subsequent frames will reference video frames previous to that point.

In the extended profile setting, AVC provides the tools to use new types of video frames: switching I (SI), and switching P (SP) frames or slices. SI and SP slices allow better switching between video streams with different data rates. These frames also enable better support for trick modes that enable abilities such as fast-forward, and fast-reverse.

To provide a better support for both coding and network communication systems, AVC is divided conceptually into two layers: video coding layer (VCL), and network adaptation layer (NAL). The video coding layer describes the video content efficiently. The network adaptation layer provides the header information to encapsulate the video content to allow it to be stored or sent across various transport layers or storage media, as shown in Figure 3-7. For more information the reader can refer to [19].

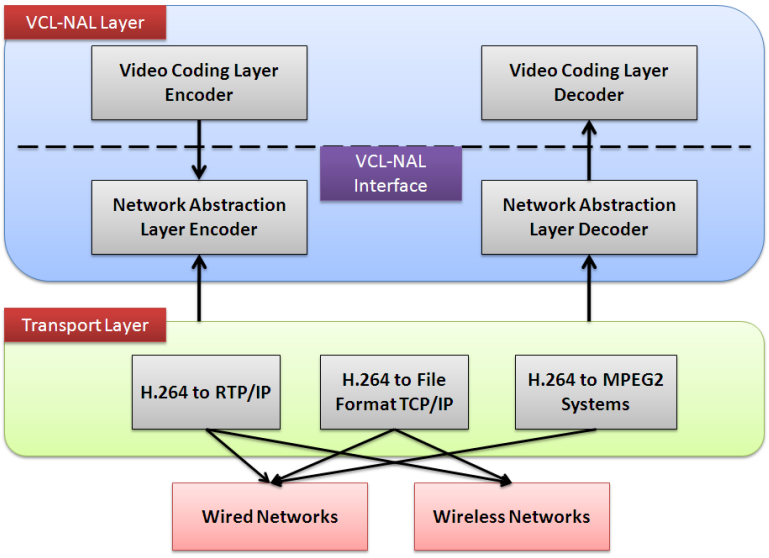


Figure 3-7 AVC Layer Hierarchy

NAL units are divided into VCL and non-VCL units. VCL units are data units that carry the video frames information. Non-VCL units carry information about the data like parameters set, and other optional data that can enhance the usability of the decoded frames.

NAL units can be formed as packets or byte stream formats depending on the associated transport layer. Systems like MPEG-2 expect a continuous byte or bit stream, the identification of the boundaries of the NAL units is achieved by a specific pattern in the byte stream. On the other hand, IP based systems like real time protocol (RTP) expect packetized video streams. Packet based division nullifies the need of pattern based boundary recognition. Most real time video transmissions are based on IP/RTP.

3.3 MPEG-4 Part 10 SVC Extension

Scalable video codec (SVC) extension was added to AVC standard to support video scalability. There are three different types of scalability supported by the standard: temporal scalability (TS), spatial scalability (SS), quality scalability, and a hybrid spatio-temporal-quality scalability. These types of scalability are depicted in Figure 3-8.

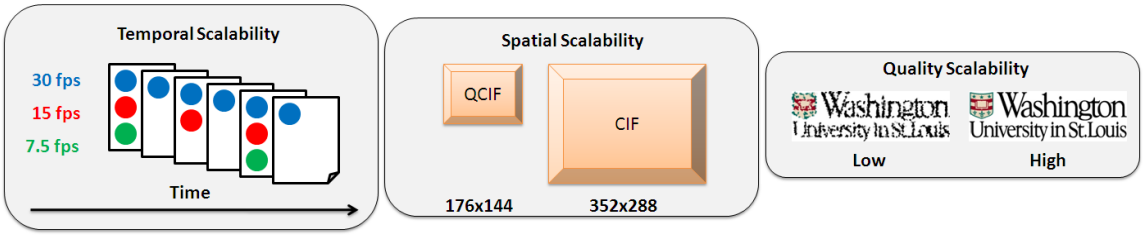


Figure 3-8 Types of Video Coding Scalability

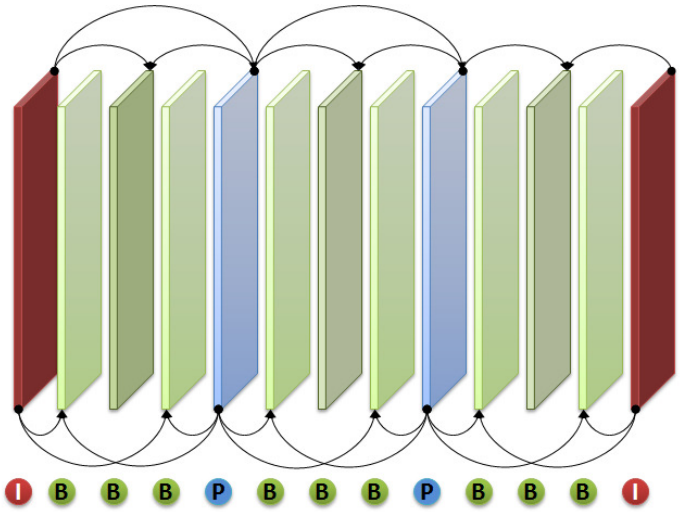
Temporal scalability is performed by splitting the video bit stream into a base layer and a hierarchy of enhancement layers. Processing more portions of the bit stream (i.e. enhancement layers) increases the frame rate of the received video bit stream [20].

Spatial scalability is performed by having a hierarchy of enhancement layers, and each layer corresponds to a spatial resolution. In each spatial layer, predictions are performed as in single layer coding. Inter-layer prediction mechanism is applied to improve the coding efficiency and avoid the redundancy provided in the previous simple multicasting approach. Up-sampling lower layers can be applied to achieve the desired resolution when higher layers are not present.

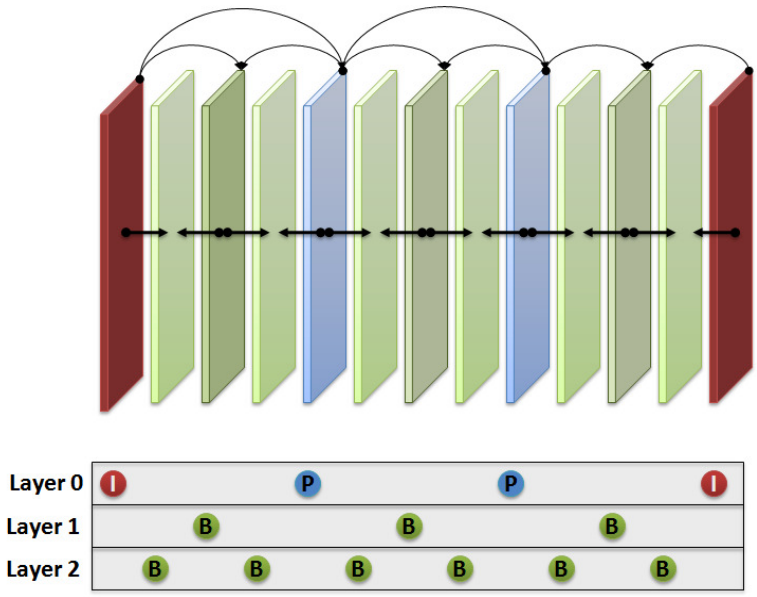
Quality scalability is also known as signal-to-noise ratio (SNR) or fidelity scalability. In this type of scalability, the base layer and the enhancement layers have the same spatial resolution. The higher enhancement layers are usually encoded with smaller quantization levels than the lower layer to provide better picture quality. So by decoding a subset of the received bit-stream, the same temporal and spatial characteristics of the complete bit stream is achieved but with lower fidelity level.

Temporal scalability, or SVC-TS, is better suited for mobile video devices, since it can meet different bandwidth constraints. It is also better for low power CPU devices [21]. AVC video standards encoders support SVC-TS to a certain degree.

As shown in Figure 3-9 (a), in the previous standards to predict a P-frame only one reference frame (either I or P-frame) from the past was allowed. In B-frame prediction, one past reference frame, and one future reference frame were allowed. AVC introduced a new concept in B-frame prediction that allows accessing multiple reference frames, and referencing these frames with unequal weights.



(a) Classical B-frame Prediction Structure



(b) Hierarchical B-frame Prediction Structure

Figure 3-9 B-Frame Prediction Structure

AVC introduced a new concept in predicting B-frames. This new technique views B-frames as a hierarchy of frames. SVC adopted the new approach as it naturally represents the hierarchy of the enhancement layers. Therefore, SVC single-layer stream is decodable by the existing AVC decoders. In this new approach, the number of referenced frames restriction in classical B-frames prediction is lifted. In addition, other B-frames can be referenced to obtain better compression ratio. Figure 3-9 (b) illustrates how B-frames can be referenced by other B-frames in the shown *dyadic* hierarchy of B-frames. In this type of hierarchy distribution the number of B-frames between I and P frames are equal to $n = 2^k-1$, where $k=1,2,3,\ldots$.

In this chapter we provided the reader the necessary background information about MPEG video encoding standard. In the next chapter we will provide an introduction to time series analysis.

Chapter 4

4 Modeling HD Video Traces Using Time Series Analysis

A time series is a series of values $\{x_1, x_2, x_3, \ldots, x_t, \ldots\}$ that can be observed through time at discrete sampling points. A time series has a close connection to time as it is represented corresponding to its sampling time. For example, x_1 is observed at time 1, and x_t is observed at time t. For the sake of simplicity it is easier to assume that these values are sampled at integer times. Time series are used to model the current behavior of a time series in order to forecast its future values. The models are usually considered with the near-present points as they are more representative of the current values of the time series. Once a valid model is developed, the time series future values can be forecasted using an appropriate forecasting technique.

As shown in Figure 4-1, time series are classically decomposed into three components:

$$x_t = Tr(t) + S(t) + I(t)$$

where:

- Tr(t) is a trend component.
- S(t) is a seasonal component.
- I(t) is a irregular or random component.

The trend component indicates the underlying direction of the time series. The seasonal component describes a pattern in the time series that occurs periodically or *seasonally*. The irregular or random component is an expression to indicate the parts of the time series that do not conform to the developed model due to random or transient variations between one period and another.

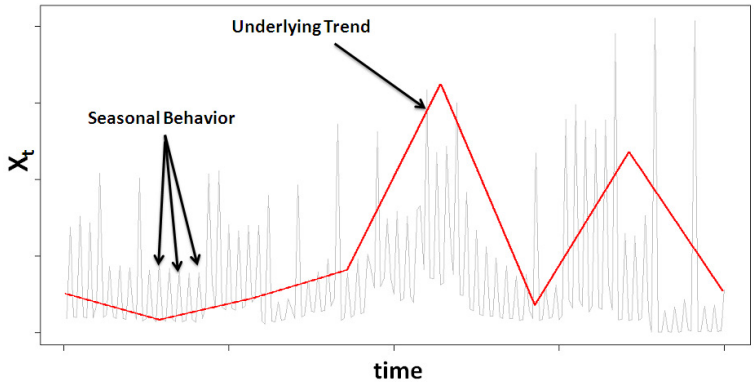


Figure 4-1 Time Series Characteristics

Time series in its simplest form can be representing a single constant value that does not change over time. This can be easily represented as:

$$x_t = b + e_t$$

where b is the constant term, and e_t is the random or error component that corresponds to each sample value at time t. An autoregressive model (AR) represents an equation that expresses the regression of X_t on its previous values. An autoregressive process of order p, or AR(p), can be expressed as :

$$x_t = \varphi_1 x_{t-1} + \cdots + \varphi_p x_{t-p} + e_t$$

where $e^t \sim N(0,\sigma^2)$ is the error term. To allow a better representation of AR model, we use the backward operator (B), where:

$$B\ x_t = x_{t-1}$$

and $B^j x_t = x_{t-j}$

Using the backward notation, AR model can be represented as:

$$x_t = \left(\varphi_1 B + \cdots + \varphi_p B^p\right) x_t + e_t$$

X_t is said to be a moving average (MA) process if it is dependent on its pervious error terms. MA of order q or MA(q) can be expressed as:

$$x_t = \left(1 - \theta_1 B - \cdots - \theta_q B^q\right) e_t$$

Therefore, an autoregressive moving average (ARMA) model that combines the previous two models can be expressed as:

$$x_t = \left(\varphi_1 B + \cdots + \varphi_p B^p\right) x_t + \left(1 - \theta_1 B - \cdots - \theta_q B^q\right) e_t$$

or

$$\varphi_p(B)\, x_t = \theta_q(B)\, e_t$$

where

$$\varphi_p(B) = 1 - \varphi_1 B - \cdots - \varphi_p B^p \text{ and } \theta_q(B) = 1 - \theta_1 B + \cdots - \theta_q B^q$$

ARMA assumes a stationary time series, or in other words, there is no systematic change in the mean or the variance influenced by a trend, but in practice most time series are non-stationary. To remove the non-stationarity of a series, a differencing operator is introduced. After integrating the differencing process into the time series, the integrated series is then analyzed to produce a valid model. This process results in an autoregressive integrated moving average (ARIMA) model. The order of differencing is denoted by *d*, and the differencing operator for *d* degree of differencing using backward operator notation is:

$$\nabla^d x_t = (1 - B)^d x_t$$

So an ARIMA process can be described as:

$$\varphi_p(B) \nabla^d x_t = \theta_q(B)\, e_t$$

A seasonal time series is a series that exhibits a seasonal periodic behavior every *s* observations. This behavior can be expressed by extending the definition of ARIMA explained earlier. Seasonal ARIMA or SARIMA represents the seasonal autoregressive part of order P as Φ_P , and the seasonal moving average of Q order as Θ_Q, and the seasonal differencing of order D as ∇_s^D. Multiplicative SARIMA thus can be denoted as:

$$\varphi_p(B) \Phi_P(B^s) \nabla^d \nabla_s^D x_t = \theta_q(B) \Theta_Q(B^s)\, e_t$$

A simpler notation to represent the order of each of the SARIMA model components is:

$$SARIMA = (p, d, q) \times (P, D, Q)^s$$

In order to represent a time series model using SARIMA, the model's components order and seasonality need to be identified first. The identification process requires human intervention to determine the best model to represent the analyzed data [22].

Searching for a good ARIMA model is based on the following steps:

- Model Identification: at this stage we try to decide the order of the model parameters (i.e. p, d, q, P, D, and Q)
- Parameter Estimation: the next step is to estimate the model coefficients using one of the different available estimation methods.
- Diagnostic checking and model verification which include modification to the model if necessary.

The model identification process is based on the interpretation of the video trace autocorrelation function (ACF) and partial autocorrelation function (PACF) plots. ACF plots are commonly used tools to check for randomness in a data series by plotting the data set values over several time lags. Given a data series x_t, PACF for a lag k is the autocorrelation between x_t and x_{t-k} that is not accounted for by lag 1 to k-1 inclusive.

The interpretation is dependent on the researcher expertise. As shown in Figure 4-2, the ACF plot can be interpreted in two different ways: either it trails at lag=1, or it cuts off at lag=2. The presence of these two interpretations leads to different models to consider in the next steps of analysis.

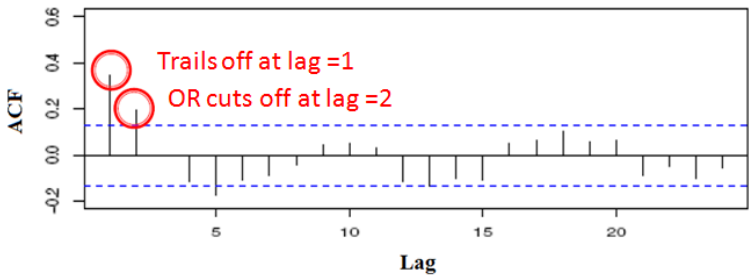


Figure 4-2 Interpreting ACF and PACF Plots

Table 4-1 shows a summary of the associated model parameter with the corresponding interpretation of the ACF and PACF graphs.

Table 4-1 Model Identification Based on ACF and PACF Plots

	AR(*p*)	**MA(*q*)**	**ARIMA(*p*, *q*)**
ACF	Trails off	Cuts off at lag *q*	Trails off at lag *q*
PACF	Cuts off at lag *p*	Trails off	Trails off at lag *p*

Because of the different possible interpretations, there might be more than one model to consider. To determine the seasonal part of ARIMA, we repeat the same process but we instead compare the values at the lags equals to the model predicted seasonal period.

The most common estimation methods are: maximum likelihood (ML) estimation, conditional sum-of-squares (CSS) estimation, and a hybrid approach where the starting values are estimated using CSS then ML is used to complete the estimation process (CSS-ML) [1]. This two-step process, (i.e. determining the model and estimating its parameters), is a time consuming process and requires a substantial statistical background to identify the best possible model to represent a time series [23].

The model's verification process is essential to know that the chosen model is accurate. If the verification process fails, the researcher needs to either try a new model from the set of the candidate models obtained earlier, or try different series transformation processes like taking the log of the video trace values and model it instead.

The most common tests include plotting the histogram of the modeling residuals to see if it resembles a random process. The same result can be achieved using either a quantile-quantile (QQ) plot, or a density plot.

We can also examine the correlation of the modeling residuals to check if there is a certain pattern in the residuals. If there is a pattern, then the model needs to be reconsidered. The examination is done by plotting the ACF graph for residuals.

Another test that can be considered is the portmanteau test, where the model residuals correlations are tested to be different from zero. The most common portmanteau is the Ljung-Box test, which can be defined as follows:

$$Q = N(N+2)\sum_{k=1}^{s}\frac{r_k^2}{(N-k)}$$

where N is he number of observations (or video frames), s is the number of coefficients to test the autocorrelation, r_k is autocorrelation coefficient for lag k, and Q is the portmanteau

test statistics. The null hypothesis in this case is that there is no autocorrelation coefficient up to lag s that is different from zero.

In this simple introduction we described the key characteristics of time series analysis and the used models to represent the characteristics of the time series under observation. We chose seasonal ARIMA over other seasonal time series approaches for its simplicity and accuracy to represent seasonal time series. Our model, namely: simplified seasonal ARIMA model or SAM, is based on seasonal ARIMA model representation. The simplification of ARIMA, as will be discussed later in more details in the next chapter, allows a unified approach to model video traces as a time series without the need of human intervention. Such simplification allows the model to be considered not only for modeling video traces with ease, but also for using it to be the basis of real time video traffic predictor.

In the next sections we will discuss our approach to model the different types of video traces encoded with the three previously discussed standards: MPEG-4 Part2, AVC, and SVC. We also discuss how our research results have confirmed the accuracy and applicability of our presented SAM model.

4.1 Modeling Video Traces: Related Work

Numerous methods to model MPEG traffic have been considered in previous research works. These techniques aim to resolve the inhomogeneous behavior of VBR video traffic. One of the first approaches was the use of M/G/∞ input processes [24]. In [25-29] a Markov chain has been used to model the video traffic. Markov chain is known to be easy to implement but it does not model the video traffic accurately. This is due to the fact that simple Markov chain models fail to capture the long range dependence (LRD) nature of video traffic. In addition, it is also known that Markov chain models need a considerable number of states and parameters to achieve an acceptable accuracy.

One of the main approaches to model MPEG traffic and overcome the continuous changing of video frame size patterns, is to divide the video trace into I, P, and B frame series. Each of these series is then modeled independently. This model is usually referred to as IPB model, or composite model. In [14,15,26,28,30] the authors argue that a single accurate model for MPEG video traffic is hard to achieve if not infeasible. The IPB modeling approach is not only difficult to implement, it requires 3 separate processes of modeling and

requires a series of multiplexing and de-multiplexing of video frames in order to provide the necessary composite model. Such excessive computation is not favorable for online video processing.

More sophisticated methods have been proposed in [31, 32] where wavelet models for video traffic were used. This allowed a unified approach for both SRD and LRD. However, these models do not consider the compression structure of MPEG, which influences the video traffic especially on small time scales.

In both [33] and [25], video motion and texture have been taken into consideration while modeling video traffic. A larger time period than a scene called "Epoch" was introduced in [33]. The authors have also identified different levels of changes inside the video file: GoP size variation, scene variation, and finally epoch (group of scenes) variation.

Seasonal time series models like autoregressive integrated moving average (ARIMA) have been used to model GSM traffic [30]. Fractional ARIMA (FARIMA)has been used to capture the long range dependence (LRD) between video frames, but these efforts were later questioned for real implementation because of their implementation complexity and the marginal improvements that they provide [30, 33].

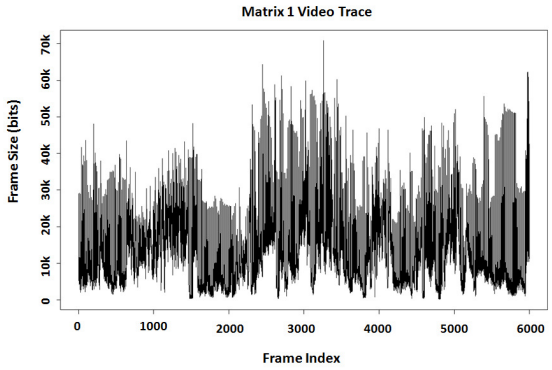
In [34] a seasonal FARIMA has been considered to model the traffic. FARIMA is different from ARIMA in the value of differencing *d*, as FARIMA uses fractional value for *d* instead of an integer. Typical values of fractional-d are between (0, 0.5) to express the long range dependence of the series. The proposed model has 11 parameters. In [35] an FARIMA model has been used, where they showed that it is better than wavelet and simple autocorrelation models. Using FARIMA has the problem of long synthesizing time. It has been shown that FARIMA has only marginal improvements over SARIMA models [23].

Most of the models mentioned above used short movie scenes; which raises questions about their applicability to long traces. In addition, some of these models either have complex procedures to generate video frames, or require up to thousand of coefficients in order achieve the desired level of accuracy [34, 35]. The main concern in these approaches that the developed models are either: scene, movie, encoding standard, or encoding setting specific. This may hinder any possible deployment of these models for real time applications.

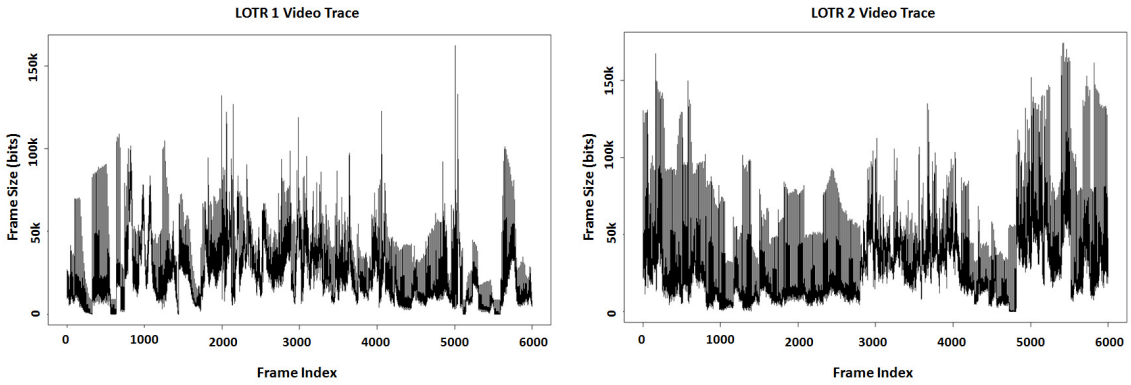
As we mentioned before, we aim to have a simple, general and accurate model that is capable of capturing the statistical characteristics of video traces. This model shall be used in both video frame generation and traffic forecasting.

4.2 Modeling MPEG-4 Part 2 Video Traces

We started our HD video analysis with video traces encoded with MPEG4 Part2 standard. For our first video samples, we used short TV commercials of 15-30 seconds length. Since our main objective is to model video traffic targeting mobile devices in a WiMAX simulation, all the videos that we tested were encoded specifically for mobile devices. We chose the following encoding settings: QVGA size (320x240), and a GoP size of 15. The GoP pattern is IBBPBBPBBPBBPBB or G15B2 [36]. Then we extended our video samples to include videos available from the video research group at University of Arizona [16]. The chosen movie scenes have the following specifications: CIF size (352×288), 25 fps, GoP G12B2, and the selected encoding profile is Advanced Simple Profile (ASP) to support B-frames. The three chosen movies are: Lord of the Rings I (LOTR I), Lord of the Rings II (LOTR II), and Matrix I. Video scenes have been chosen to be 6,000 frames in length (240 seconds or 4 minutes). These scenes have been chosen randomly from the first 20,000 frames of the total video trace. Figure 4-3 shows the frame size traces for the selected video samples.



(a) Matrix I Video Trace



(b) LOTR I Video Trace (c) LOTR II Video Trace

Figure 4-3 Videos Samples Traces

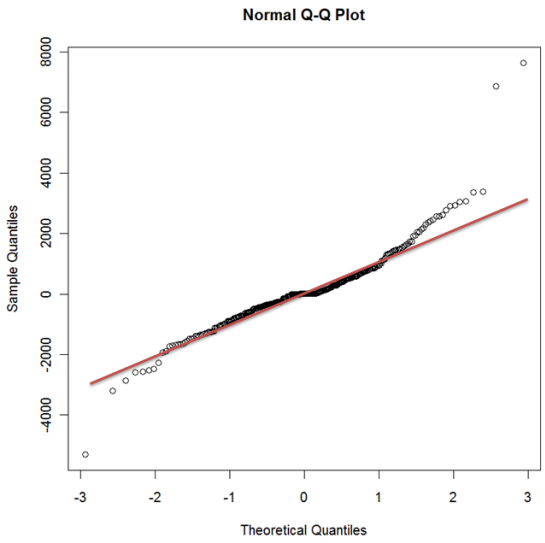
Our first concern was to verify the claim that a separate model for each frame type, or IPB model is better than an all-frames model [22, 23, 37, 38]. We also wanted to verify if a single model is can accurately model video frames.

Through our ARIMA modeling analysis, we noticed that although the optimal model for each trace was different, a particular simple model was very close to optimal in all cases. We call this model: Simplified Seasonal ARIMA Model or SAM [36]. SAM can be represented as follows:

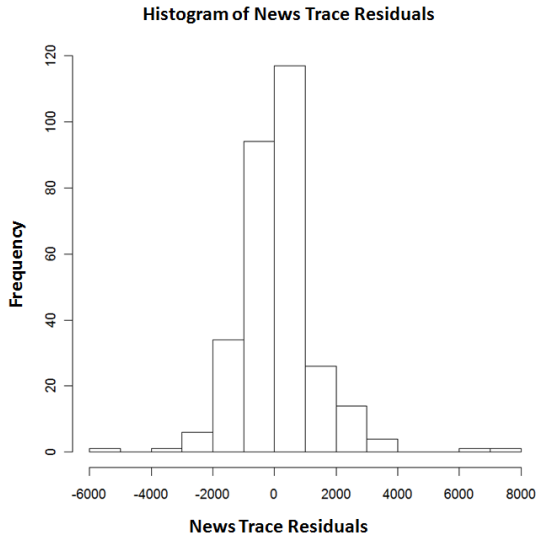
$$SAM = (1,0,1) \times (1,1,1)^{s}$$

The model has an auto-regression component of order 1, an integration of order 0, a moving average component of order 1. There is a seasonal period of *s*, where s is the seasonality of the video series, and is represented in MPEG-4 Part2 video traces by their GoP size. The seasonal part itself has an auto-regressive part of order 1, integrated part of order 1, and a moving average part of order 1. A similar approach to simplify the modeling process was considered in modeling airline data [39].

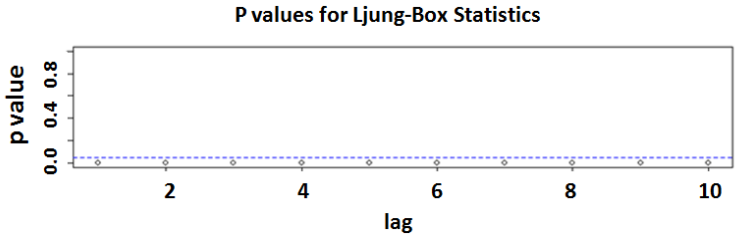
We first tested if SAM can be considered a good model. We conducted several tests to check if the modeling residuals can be considered as a Gaussian random process. An example of these examinations are shown in Figure 4-4, where the residuals QQ plot, the residuals density plot, and the Ljung-Box test for residuals confirm that SAM can be considered as a valid model.



(a) Residuals Normal QQ Plot



(b) Residuals Histogram



(c) Ljung-Box Test Results

Figure 4-4 Examining "News" Video Model Residuals

To test if SAM can be considered as an accurate simplification of the modeling process, we compared the following three methods in obtaining a model for the selected video traces:

1. A single (or all-frame) model for all video frames using the previously discussed ARIMA analysis approach.
2. A composite model (or an IPB model) obtained by modeling each of the I, P, and B frame series separately. The three models are then combined into a single composite model.
3. We used our simplified seasonal ARIMA$(1, 0, 1)\times(1, 1, 1)^s$ model.

We measured the goodness of each model using the commonly used *Akaike Information Criterion* or AIC index [40]. AIC takes into consideration both the complexity of the model

and its accuracy. Lower AIC values indicate a better goodness-of-fit for the model. AIC is generally defined as

$$AIC = 2k + N[\ln(2 \cdot \pi \cdot RSS / N) + 1]$$

here k represents the number of model's parameters, N represents the number of video trace frames, and RSS is the residuals sum of squares.

Our results, as presented in Table 4-2, confirmed that IPB model is slightly more accurate, but we argue that this slight improvement is not justifiable given the extra efforts needed to analyze and implement the three different models. The difference between the optimal composite model and all-frames model is less than 1%. The SAM model is similarly close to the optimal all-frames model. The difference is less than 0.1%. The obvious advantage of using SAM is that we can it for all movie traces. SAM is a very simple model that requires only 4 parameters to represent a movie trace, in addition to the standard deviation of the modeling errors.

Table 4-2 All-Frame Model, Composite Model, and SAM Model AIC Comparison Results

Movie		All-Frames Model	Composite model (I-Frames), (P-Frames), (B-Frames)	SAM model
Matrix I	**Model**	$(3, 0, 1)\times(1, 1, 1)^{12}$	(0, 1, 3), (1, 1, 1), (3, 1, 6)	$(1, 0, 1)\times(1, 1, 1)^{12}$
	AIC	**120369.3**	**119775.3**	**120378.1**
LOTR I	**Model**	$(1, 0, 1)\times(1, 1, 1)^{12}$	(0, 1, 5), (0, 1, 1), (2, 1, 2)	$(1, 0, 1)\times(1, 1, 1)^{12}$
	AIC	**125689.7**	**125270.9**	**125689.7**
LOTR II	**Model**	$(3, 0, 3)\times(1, 1, 1)^{12}$	(0, 1, 3), (0, 1, 1), (1, 1, 2)	$(1, 0, 1)\times(1, 1, 1)^{12}$
	AIC	**127488.4**	**125278.9**	**127597**

We then repeated the analysis on entire movie sequences and confirmed that SAM model can be used to model entire movies. For this analysis, we used the following six movies: the Matrix trilogy, and The Lord of the Ring trilogy. Table 4-3 shows the statistical characteristics of these movies. The Hurst index value indicates the video sequence's ability to regress to its mean value, with higher values indicating a smoother trend, less volatility, and less roughness. Its value varies between 0 and 1. This is also an indication of the long range dependence (LRD) between the frames. For finite number of frames *N*, the Hurst index can be computed by first calculating the mean adjusted series *Y*:

$$Y_i = x_i - \bar{x}\,, \quad i = 1,2,\ldots,N$$

where x_i is the frame size at index i, $\bar{x}$ is the mean frame size over the trace length *N*, then we calculate the cumulative deviate vector *S:*

$$S_i = \sum_{j=1}^{i} Y_j \ , \quad i = 1,2,\ldots,N$$

the next step is to calculate the range value *R ,* and we divide it over the standard deviation value denoted by σ:

$$R = \frac{\max(S) - \min(S)}{\sigma}$$

$$Hurst\ Index = \frac{\log(R)}{\log(N) - \log(2)}$$

Table 4-3 Statistical Analysis of Long Video Traces

Movie	Standard Deviation	Mean	Variance	Hurst Index
LOTR 1	9594.778	9342.26	92059757	0.9158
LOTR 2	11178.38	11481.00	124956269	0.9158
LOTR 3	10794.25	11145.63	116515800	0.9233
Matrix 1	7946.338	7348.922	63144295	0.9011
Matrix 2	10687.00	9508.467	114212020	0.9147
Matrix 3	12701.56	10522.08	161329728	0.9253

The mean and the variances of the frame sizes listed in Table 4-3 indicate that the six movies are quite different. However, these movies can all be well represented by the SAM model. Our results, as shown in Table 4-4, show that the six movies have similar model parameters (autoregressive or AR, moving-average or MA, seasonal autoregressive or SAR, and seasonal moving-average or SMA). The difference between these values is small and it is less than 1%.

Table 4-4 SAM Model Parameters Values for Various Movies

Movie	AR	MA	SAR	SMA
LOTR 1	0.9262	-0.6911	0.2411	-0.8638
LOTR 2	0.9306	-0.6770	0.2715	-0.8610
LOTR 3	0.9322	-0.6818	0.2683	-0.8440
Matrix 1	0.9241	-0.6561	0.1602	-0.8050
Matrix 2	0.9382	-0.6809	0.2336	-0.8760
Matrix 3	0.9327	-0.6372	0.1002	-0.8951

Mean	**0.93**	**-0.67**	**0.21**	**-0.86**
[Min, Max]	[0.924,-0.938]	[-0.691, -0.637]	[0.1,0.271]	[-0.895,-0.805]
Abs ([Max-Min]/Mean)	0.015054	0.080597	0.814286	0.104651

We further compared the optimal all-frame models with SAM for these six movie traces as shown in Table 4-5. We used the following error comparison quantities: mean absolute error (MAE), mean absolute relative error (MARE), inverse of signal to noise ratio (SNR^{-1}), and normalized mean square error (NMSE). Notice that on all these statistical measures, SAM is close to optimal with less than 1% difference. MAE, MARE, NMSE and SNR^{-1} are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|e_i|$$

$$MARE = \frac{1}{N}\sum_{i=1}^{N}\frac{|e_i|}{x_i}$$

$$NMSE = \frac{\frac{1}{N}\sum_{i=1}^{N}|e_i|^2}{\bar{x}\cdot\bar{\hat{x}}}$$

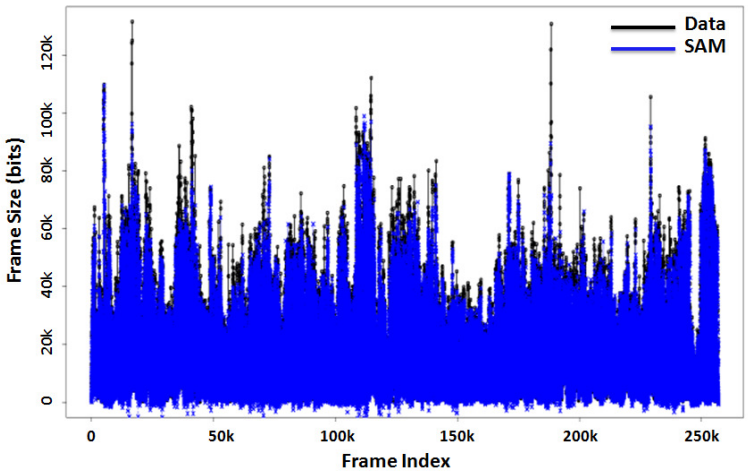
$$SNR^{-1} = \frac{\sum_{i=1}^{N}|e_i|^2}{\sum_{i=1}^{N}|x_i|^2}$$

where e_i is the modeling error at index i, x_i is the frame size at the i-th index, N is the number of frames, $\bar{x}$ is the mean frame size, and $\bar{\hat{x}}$ is the mean frame size corresponding to the model.

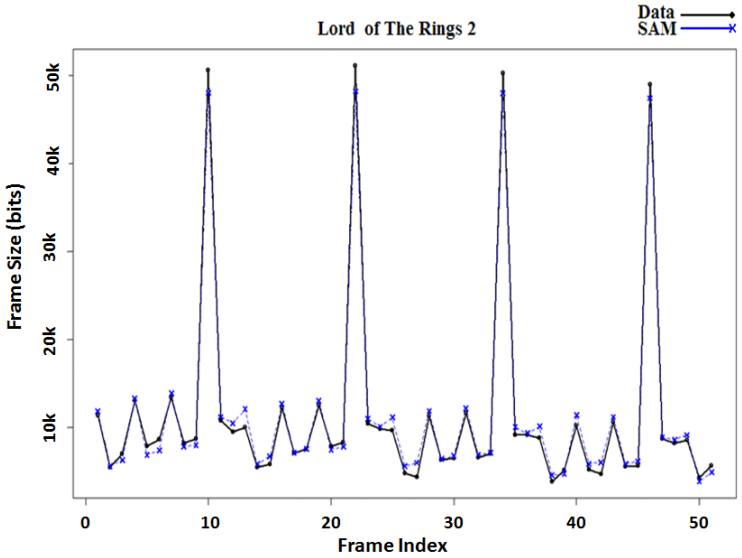
Table 4-5 Statistical Comparisons Between SAM, and the Calculated Optimal Models

Optimal Model				
Movie	**MAE**	**MARE**	**SNR^{-1}**	**NMSE**
LOTR 1	1850.149	0.3256206	0.0848033	0.1652013
LOTR 2	2038.680	0.2806260	0.0708604	0.1456091
LOTR 3	1940.064	0.2889833	0.0685161	0.1415653
Matrix 1	1553.833	0.3700388	0.0957917	0.177721
Matrix 2	2126.052	0.3839772	0.0993043	0.1779137
Matrix 3	2830.622	0.3941804	0.1267721	0.2137702
SAM				
Movie	**MAE**	**MARE**	**SNR^{-1}**	**NMSE**
LOTR 1	1851.281	0.3240269	0.0848344	0.1652620
LOTR 2	2043.132	0.2799332	0.0709581	0.1458099
LOTR 3	1944.378	0.2888479	0.0686010	0.1417407
Matrix 1	1553.584	0.3694246	0.095829	0.1777901
Matrix 2	2132.762	0.3864979	0.0995010	0.1782661
Matrix 3	2845.982	0.3957961	0.1277827	0.2154743
MAX Diff %	**0.5426%**	**0.6565%**	**0.7972%**	**0.7972%**

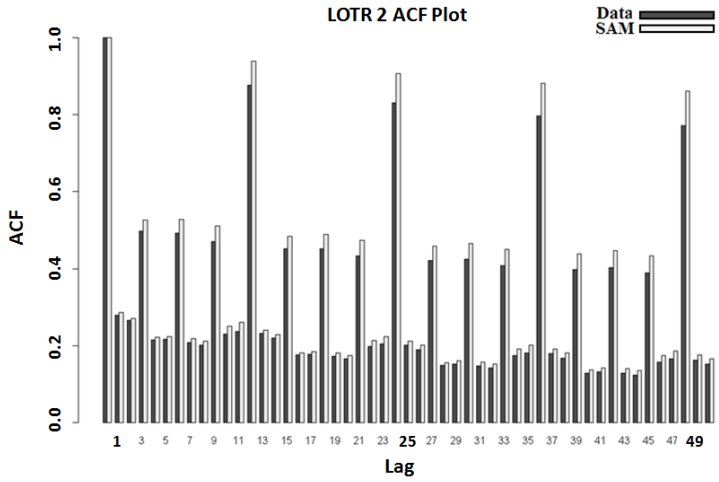
We compared SAM to the original video traces using several graphical comparisons. The comparisons include the following graphs: auto-correlation function (ACF), cumulative distribution function (CDF), and video trace comparison. Figure 4-5 shows the results for LOTR II trace, which demonstrates the accuracy of SAM.



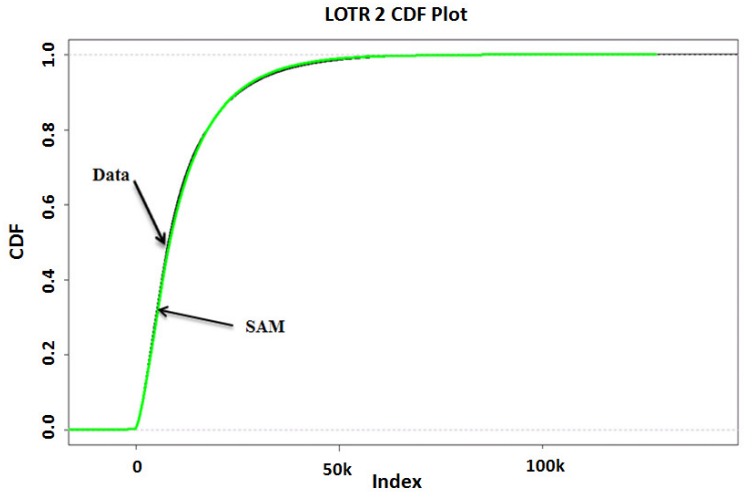
(a) Full Trace Length Comparison



(b) A Close-up Comparison



(c) ACF Comparison



(d) CDF Comparison

Figure 4-5 SAM Model Results: LOTR II Movie Trace

We have shown in this section that SAM model is capable of capturing the statistic features of MPEG-4 Part2 video traces. Similar movies with similar statistical characteristics have been shown to have similar parameter values. These results have encouraged us to pursue our analysis to analyzing movies encoded with other commonly used codec for mobile video.

4.3 Modeling MPEG-4 Part 10 / AVC Video Traces

MPEG-4 Part 10 or AVC is also known as H.264 standard. The standard has shown significant improvements over older codecs. AVC encoded movies have lower mean values compared to MPEG-4 Part 2 videos. This is due to the fact that AVC compression is more complex and thus on the down side it requires more processing power. Long range dependence (LRD) level between video frames has been recorded to be similar to MPEG videos. Because of the new techniques in AVC compression, the encoded videos have higher variability in their frame sizes. Therefore, an accurate model that can represent highly variable sizes of video frames is a nontrivial task. The reader can refer to [17] for more information about AVC codec and the characteristics comparison between AVC and MPEG videos.

One of the main differences between MPEG-4 Part2 and AVC encoded videos is the multiple-frame-reference feature in AVC. As we mentioned earlier, this feature allows the picture frames to refer to multiple reference frames to improve the frame's compression. We noticed that this features results in changing the seasonality period from s to $2s$ for AVC videos, where s is the GoP size. Figure 4-7 shows the autocorrelation function (ACF) for AVC coded video. Notice that the repetition period is $2s$. This observation led us to change our SAM model from its previous formula to the following formula.

$$SAM_{AVC} = (1,0,1) \times (1,1,1)^{2s}$$

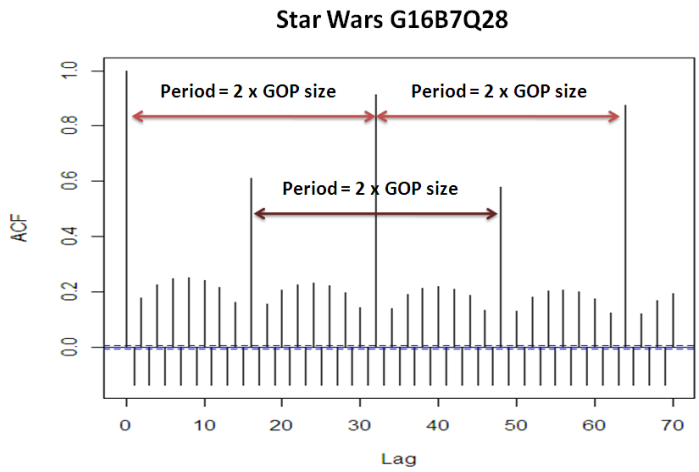
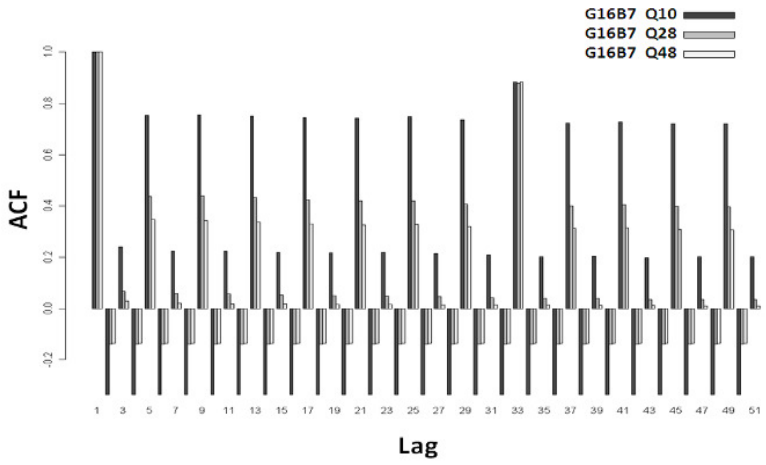


Figure 4-6 Seasonality in AVC Encoded Movies

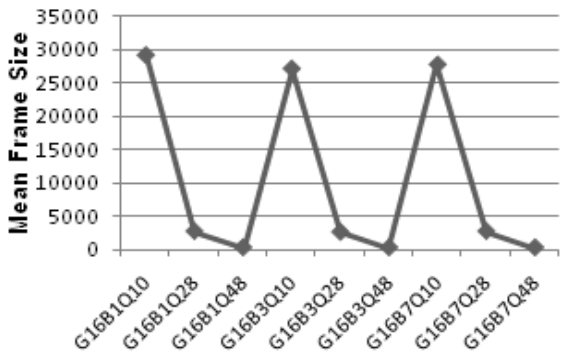
Another way to describe SAM is to represent the seasonality of the model independently of the GoP size. The seasonality can be described by referring to the interval between two consecutive maximum or peak ACF values instead. This value can be obtained easily either visually from the ACF plot, or using simple mathematical approaches like comparing the maximum ACF values over a reasonable number of lags.

To test the applicability of the model, we analyzed several full AVC-encoded movie traces with different video encoding settings. Through our analysis, we noticed that different quantization levels will result in scaling the frames sizes up and down without interfering with their autocorrelation, as shown in Figure 4-7(a). We have chosen GoP structure of G16B7 with a quantization level of 28 (I=28, P=28, B=30) as a good compromise of the quality and the size of video frames.

Increasing the quantization parameter (QP or Q) results in lower frame sizes, lower quality, and requires more computational power to decode on the receiver side. Q28 is a good choice compared to Q10, and Q48 as shown in Figure 4-7(b). Our decision is based on the fact that Q28 does not result in large frame sizes, compared to Q10, and does not require extensive computational power, compared to Q48. A quantization level of 28 (Q28) is close to the AVC JM reference software [41] default values (I=24,P=24,B=24) as well.



(a) Quantization Effect on ACF



(b) Quantization Effect on Mean Frame Size

Figure 4-7 Quantization Level Effects on Video Frames

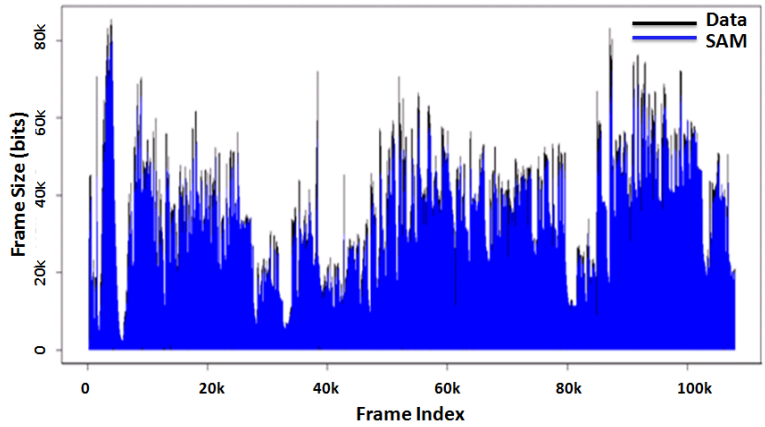
We have tested our model using the following GoP structures: G16B1, G16B3, and G16B7. We used the following movies : Silence of the Lambs (~30 min), Star Wars IV(~30m), the Tokyo Olympics (~74 min), a clip of an NBC news broadcast (~30 min), and a Sony demo (~10 min). We tested several optimized models for AVC encoded traces and compared them against the SAM_{AVC} model. These optimized models are the result of extensive analysis of the video traces to determine the best possible model following the steps in [37].

Because SAM_{AVC} is simpler than any of the other calculated models, its AIC results were better than the calculated models. Table 4-6 shows some of the obtained results.

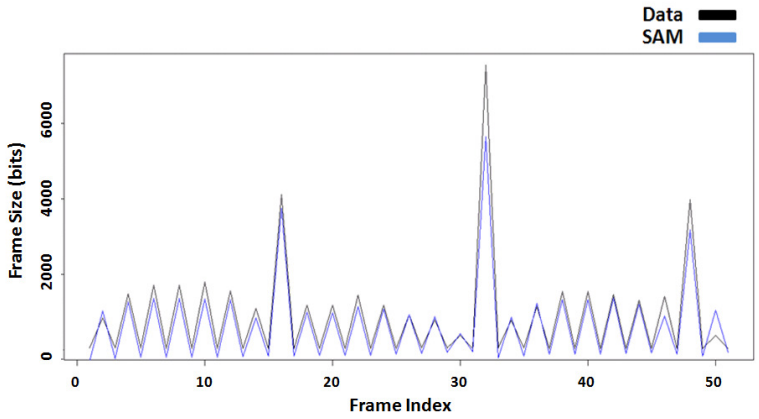
Table 4-6 Comparison Between SAM_{AVC} and Calculated Models

Movie	Calculated Model	SAM_{AVC} Model
Star Wars IV	**$(1,1,1)x(1,1,1)^{32}$**	**$(1,0,1)x(1,1,1)^{32}$**
	AIC = 1045796	AIC = 1040561
Silence of the Lambs	**$(2,2,2)x(1,1,1)^{32}$**	**$(1,0,1)x(1,1,1)^{32}$**
	AIC = 1051632	AIC = 1049195
Tokyo Olympics	**$(2,0,2)x(1,1,1)^{32}$**	**$(1,0,1)x(1,1,1)^{32}$**
	AIC= 2702438	AIC = 2695848

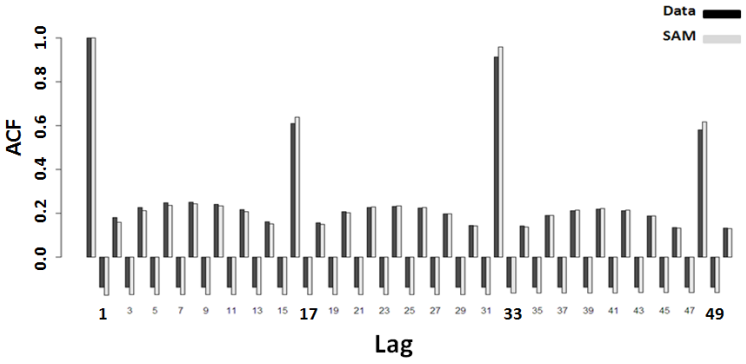
Figure 4-8 shows an example of our results. The figure shows how well the generated trace using SAM_{AVC} for Star Wars IV movie trace compares to the original trace. Notice that the model is capable of representing the modeled traces accurately, which is shown in the full video trace, ACF, and CDF graphs comparisons. Moreover, the results show that SAM_{AVC} is capable of modeling even the sudden transitions of the video frame sizes in the modeled video traces. In our analysis we used both R [42] and SAS applications [43].



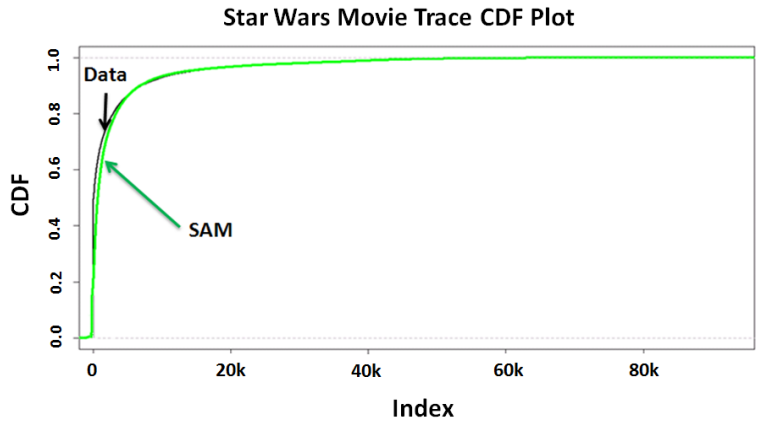
(a) Full Trace Comparison



(b) A Close-up Trace Comparison



(c) ACF Comparison



(d) CDF Comparison

Figure 4-8 SAM_{AVC} Comparison Results

We have conducted a similar test on several of the commonly used YUV 4:2:0 reference video sequences available through [44]. To evaluate SAM_{AVC} with different encoding settings, we encoded the video sequences with the following encoding settings, as shown in Table 4-7.

Table 4-7 Encoding Parameters for YUV Reference Video Sequences

Parameter	Value
FPS	30
Resolution	CIF (352x288)
Profile	Main & Extended
Decoder Min. Support	CIF and below with 3041280 samples/sec
Number of Reference Frames	3
IDR Period	30
Symbol Mode	CAVLC

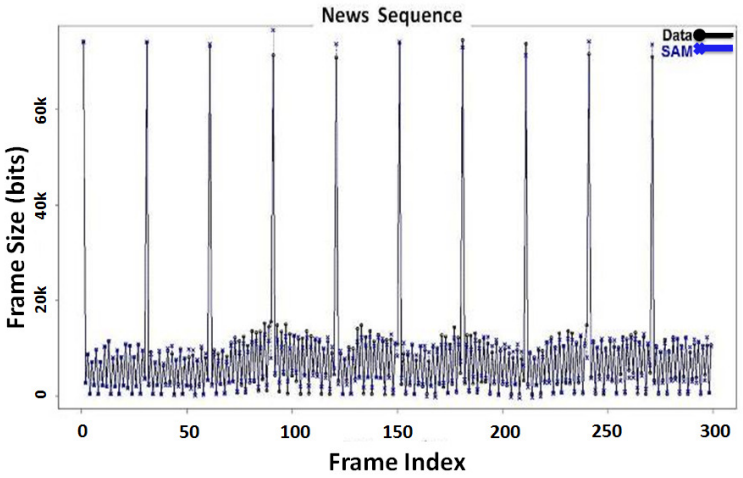
We have encoded the video sequences using Main and Extended profiles with a frame rate of 30fps. We specified the minimum requirement for the decoder to decode CIF resolution videos at 3041280 samples per second. We used instantaneous decoding refresh (IDR) frames with a period of 30, which matches the fps rate. An IDR frame is a special type of I frame that allows better seeking precision and thus enhances the user's experience. We used also Context-Adaptive Variable Length Coding (CAVLC) mode since it is supported by all H.264 profiles, unlike Context-Adaptive Binary Arithmetic Coding (CABAC) mode. We chose these encoding parameters to be close to the suggested settings for HD video in [45].

Table 4-8 below shows few of the statistical characteristics of some of the analyzed video sequences. The shown values show the statistical diversity found in these video traces. Hurst index is an indication of the video trace ability to regress to the mean, with higher values indicating a smoother trend, less volatility, and less roughness.

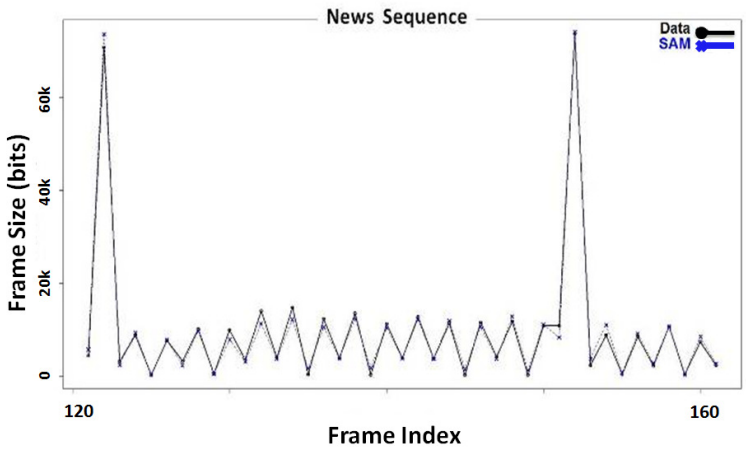
Table 4-8 Statistical Characteristics of Few of the Modeled YUV Video Sequences

Movie	Mean	Max, Min	Hurst Index
Bridge(Close) [1998 frames]	15460	99060,96	0.5491166
News [300 frames]	8602	74490,304	0.4757949
Foreman [300 frames]	15090	137000,264	0.6385751

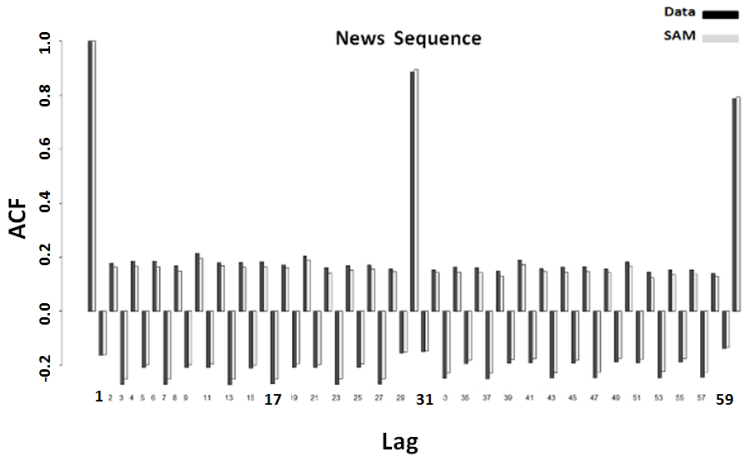
Figure 4-9 shows a comparison of measured traces and SAM model generated traces for one sample YUV video sequence. Notice that SAM model accurately represents the video sequences.



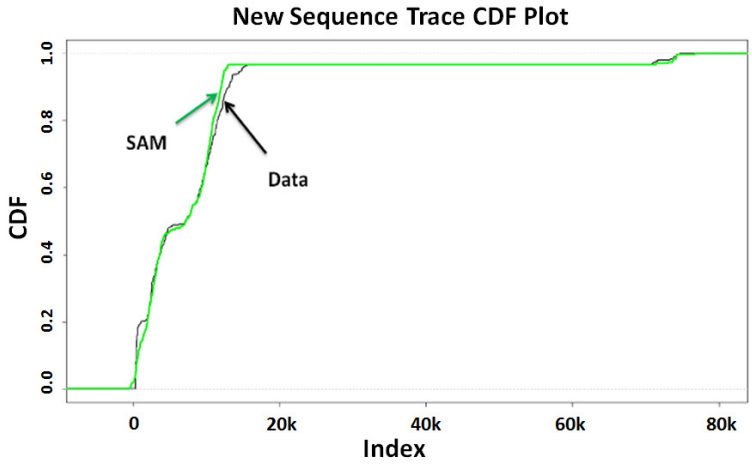
(a)Full Trace Comparison [300 frames]



(b) A Close-up Comparison [frames 120-160]



(c) ACF Comparison



(d) CDF comparison

Figure 4-9 SAM_{AVC} Results for "News" YUV Reference Trace

In this section, we presented our analysis and results of modeling various video traces encoded with different encoding settings using SAM_{AVC} model. In the next section, we will show our analysis and results of modeling SVC-TS encoded video traces.

4.4 Modeling SVC-TS Video Traces

In this section, we discuss our approach to model video traces encoded with the Scalable Video Coding (SVC) extension with emphasis on temporal scalability. SVC provides a better solution to support the wide variety of video quality levels required due to the heterogeneity of hardware and software capabilities of mobile units [17, 46, 47]. The two main SVC scalability modes support scalability in two dimensions: spatial and temporal.

Temporal scalability is performed by splitting the frames into a base layer and a hierarchy of enhancement layers. The enhancement layers increase the frame rate of the transmitted video and reference the base layer frames. In spatial scalability, a higher resolution is achieved by assigning a down-sampled resolution to the base layer, then it is combined with one or more enhancement layers.

Temporal scalability, or SVC-TS, is better suited for mobile video devices, since it can meet different bandwidth constraints. It is also better for low power CPU devices [21]. Older video standards encoders support SVC-TS to a certain degree. For instance, AVC encoders did not require any change of the design to support a reasonable number of temporal or enhancement layers [48].

Our analysis of SVC encoded video focused on temporal scalable video. We have tested several movies provided by the same source for MPEG4-Part2 and AVC movie traces [17]. Our analysis has led us to adapt the SAM model to the two different types of layers: base layer, and enhancement layers. We considered in our analysis SVC video traces with base layer (Layer 0) and three enhancement layers. For our analysis, we chose similar encoding parameters to the ones used in AVC encoded video traces. The chosen encoding parameters are: a CIF size, and a GoP structure of G16B7 with a quantization level of 28 (I=28, P=28, B=30).

SVC encoded video traces are more complex than the traces we previously analyzed. Figure 4-10 shows the seasonality in SVC coded video traces that led to our formula for SAM_{SVC}.

Notice how the enhancement layers correspond to the GoP size s. For example, the seasonality for layer 0, or the base layer, is equal to 2×s, or 32. For enhancement layer 2, the seasonality is equal to s/2, or 8.

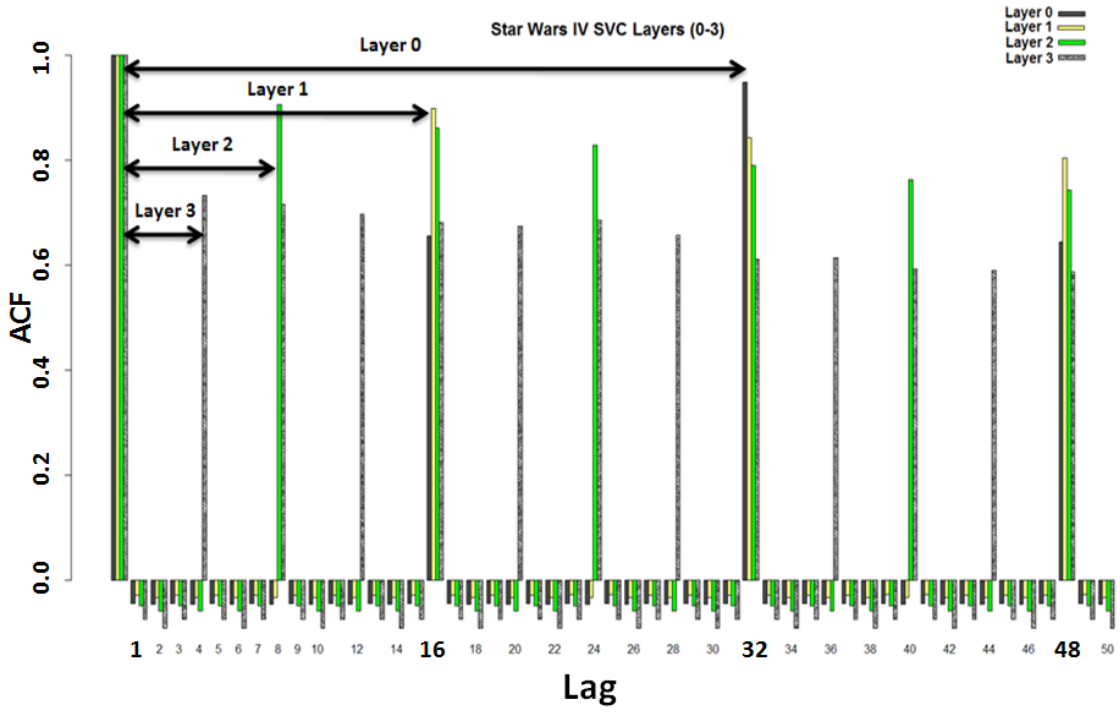


Figure 4-10 Seasonality in SVC Encoded Video (Star Wars IV)

We have concluded that the following adjustment to the SAM model, as shown below, is successful in modeling the traces correctly. Here s represents the number of frames between two consecutive I-frames, and L represents the layer level. For the base layer, L is zero and SAM_{SVC} will be identical to SAM_{AVC}.

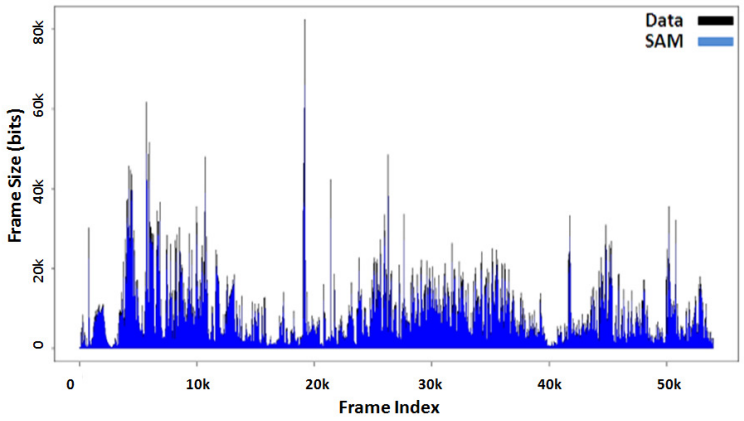
$$SAM_{SVC} = (1,0,1) \times (1,1,1)^{\frac{2s}{(2^L)}}$$

Similar to our approach in analyzing AVC video traces, Table 4-9 shows the difference in AIC values between SAM_{SVC} and calculated models. The AIC values are very close in almost all cases. Again, SAM_{SVC} proves to be a preferred model because of its simplicity and generality.

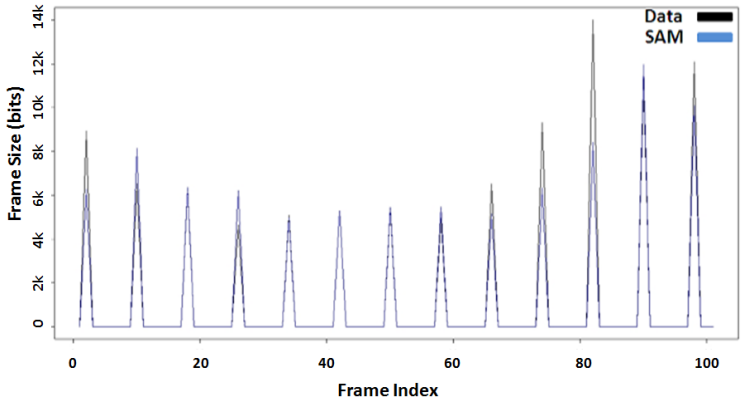
Table 4-9 Comparison Between SAM_{SVC} and The Calculated Models

	Layer 0	Layer 1	Layer 2	Layer 3
Movie	**Tokyo Olympics**			
SAM	$(1,0,1)x(1,1,1)^{32}$	$(1,0,1)x(1,1,1)^{16}$	$(1,0,1)x(1,1,1)^{8}$	$(1,0,1)x(1,1,1)^{4}$
AIC	2702089	2434080	2360311	2484516
Cal. Model	$(1,1,1)^{32}$	$(1,1,0)^{16}$	$(1,1,1)^{8}$	$(1,1,1)^{20}$
AIC	2702085	2436772	2361808	2350824
Difference%	**~0%**	**-0.11%**	**-0.063%**	**5.68%**
Movie	**Star Wars IV**			
SAM	$(1,0,1)x(1,1,1)^{32}$	$(1,0,1)x(1,1,1)^{16}$	$(1,0,1)x(1,1,1)^{8}$	$(1,0,1)x(1,1,1)^{4}$
AIC	1071597	947948.2	915773.8	924247.4
Cal. Model	$(1,1,1)^{32}$	$(1,0,1)(1,1,0)^{16}$	$(1,0,1)^{8}$	$(1,1,1)^{4}$
AIC	1071593	952005.8	917630.6	925132.6
Difference%	**~0%**	**-0.43%**	**-0.2%**	**-0.1%**

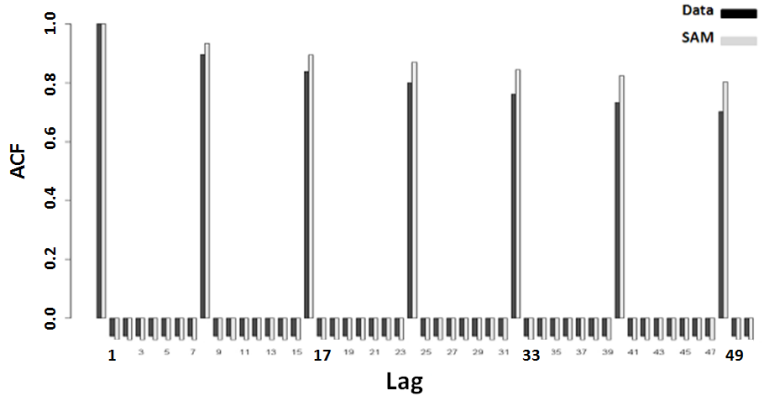
Figure 4-11 shows some of the results obtained from modeling the Star Wars SVC-TS enhancement layer-1 video trace. Note the ability of SAM_{SVC} to model the layer statistical characteristics correctly as demonstrated by the comparison of the actual trace, ACF, and CDF graphs.



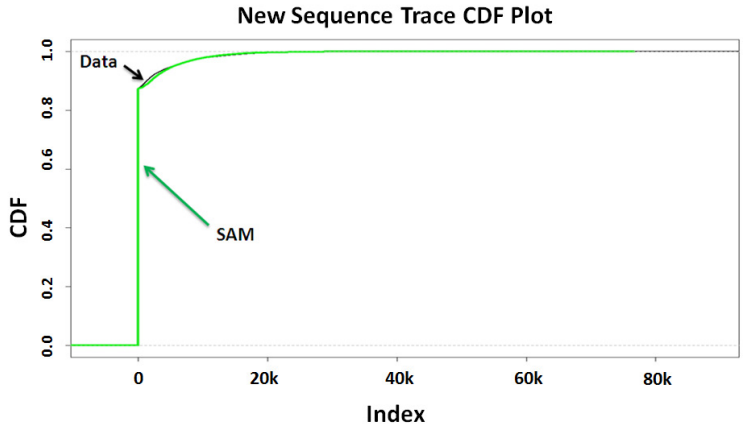
(a) Full Trace Comparison



(b) A Close-up Comparison



(c) ACF Comparison



(d) CDF Comparison

Figure 4-11 SAM_{SVC} Results

In this chapter, we demonstrated the validity of our presented SAM model to provide a convenient and accurate approach to model video traces encoded with MPEG-4 Part2, AVC, and SVC encoding standards with various encoding settings. We showed through both statistical and graphical comparisons the capability of SAM in representing the statistical characteristics of tested videos. In order to facilitate the usage of SAM, we developed different tools including a trace generator that is based on our model. The following section will discuss in detail its implementation and our design decisions.

Chapter 5

5 SAM-based Traffic Generator and other Developed Tools

Simulation provides an easy means to analyze different resource allocation strategies. While simulation environments like NS-2 provide the means to create the necessary network topology, there is still a need to provide an accurate workload for the test scenarios. The workload should represent the real world traffic accurately and should be easy to administer and adjust to different simulation conditions.

There are two ways to provide traffic workloads for mobile video simulations: actual video traces used by trace-driven simulations, and statistical models that can be used to generate the required video sequences for the simulations. Figure 5-1 illustrates the two approaches. Statistical modeling requires additional step of analyzing the video traces and modeling them in order to generate the sequences that represent the statistical characteristics of real videos.

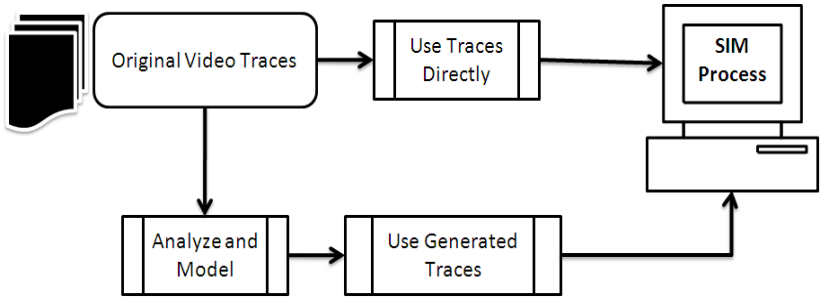


Figure 5-1 Trace-driven versus Model-based Simulations

Trace-driven simulations are known for their credibility. It is easy to convince others that the workload is representative and accurate since a real frame trace is used in the analysis. On the other hand, their usefulness and flexibility are questionable. It is hard to adjust any parameter or to extend the trace if there is a need for continuing simulations beyond the frames available in the trace file. Even if a shorter trace is required, it is not easy to determine the starting point in the trace [8, 17].

Statistical traffic models are considered a better workload choice since they provide a better understanding of the tradeoffs of the various traffic characteristics. Once a representative

model is obtained, it is easy to change and adapt it to different workload parameters. Most of the statistical models are complex and require a significant amount of time to verify and implement. A simple statistical model is, therefore, preferred as long as it closely represents the real network traffic.

The lack of good and simple video models has deterred researchers from considering statistical models as an option for their simulations. Although, there have been many attempts to provide such models, these attempts have been marked as complex, and hard to implement. In addition to that, many of these approaches were developed by working only on short movie scenes. In order to obtain a reliable and meaningful statistical model, long movie traces with thousands of frames need to be examined [25, 28, 32, 34, 49-51].

5.1 SAM-based Traffic Generator

In this section, we describe our approach to develop SAM-based traffic generator. Our analysis on all the movies was done using the open analysis package R [42]. R provides several tools to model and display the obtained results. To simulate our results we started using two tools provided by R: *arima.sim* function and *gsarima* package's function *garsim* [52]. Both functions can simulate ARIMA models but not seasonal ARIMA models. In order to proceed, we had to convert our ARIMA model to either abstracted AR, or MA models. This approach is well known to statisticians to simplify model simulations. For more information the readers can refer to [22, 23].

Our choice to select AR model over MA model was based on the fact that it is easier to keep track of the old values generated by the simulator. In addition to that, after testing both AR and MA models, we noticed that MA coefficients values do not converge over time. This affects the simulation accuracy and its implementation applicability. By converting the SAM model to a series of AR coefficients, we were able to determine the desired level of accuracy and complexity of the model. For instance, the total number of AR coefficients for MPEG-4 Part2 SAM model is 1650. Figure 5-2 below shows the different levels of accuracy that corresponds to different numbers of used AR coefficients. From our analytical analysis and simulation results, we found that 250 AR coefficients are sufficient to provide an accurate simulation.

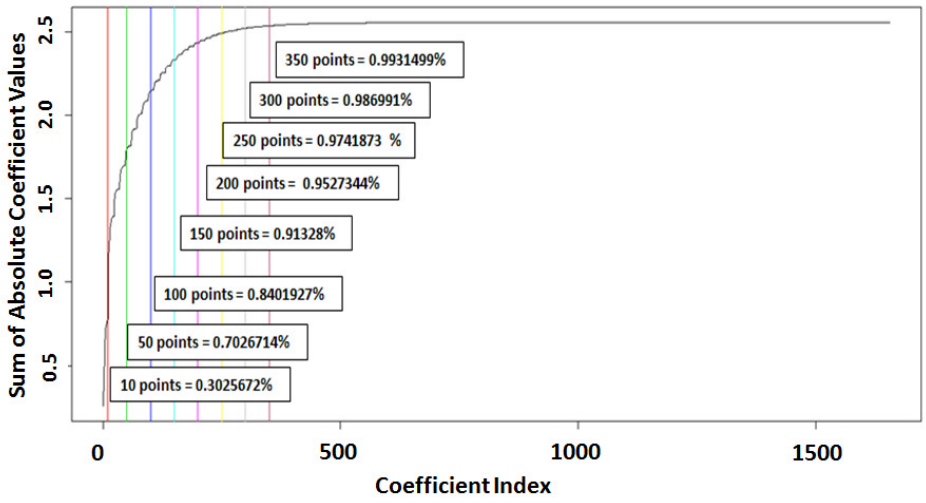


Figure 5-2 Different Accuracy Levels Corresponding to Different Numbers of AR Coefficients

SAM generator incorporates *arrep* function as a component of its implementation, which is a part of *gsarima* package. Function *arrep* is capable of converting ARIMA models to their representations as a series of AR coefficients. That allows the generator users to supply only the five parameters mentioned before, (i.e., four ARIMA parameters plus the standards deviation of error terms). The traffic generator is capable of generating any specific number of frames, and allows the user either to store the results into a file or to use the continuous stream as an inner layer for other applications.

One of the challenges in writing the traffic generator is to imitate the randomness of the transition of the movie scenes. A straight forward implementation of the seasonal ARIMA model is capable of capturing the statistical relationships between the frames, but it cannot predict or simulate the random shocks in video frame sizes [49]. This is because the model represents a smoothed version of the modeled traces due to the differencing method used. These random shocks represent a sudden transition of video frame sizes, and show up as spikes in video traces.

To overcome this problem, SAM traffic generator includes a simple mechanism that inserts random shocks into the video stream while guaranteeing the frame sizes to be within reasonable values (i.e. non-negative frame sizes). The mechanism starts by picking random points in the generated trace to be the center of the random shock representing 1% of the trace points. Then these points are multiplied by ten. The near surrounding points values are

increased to allow a smoother transition to the center of the random shock. The farther points' values are decreased in order to emphasis the sudden transition of the random shock, and to maintain the same mean frame size value over the entire movie trace.

Figure 5-3 shows a comparison between Matrix 1, Matrix 3 and a generated trace using the SAM traffic generator with random shocks.

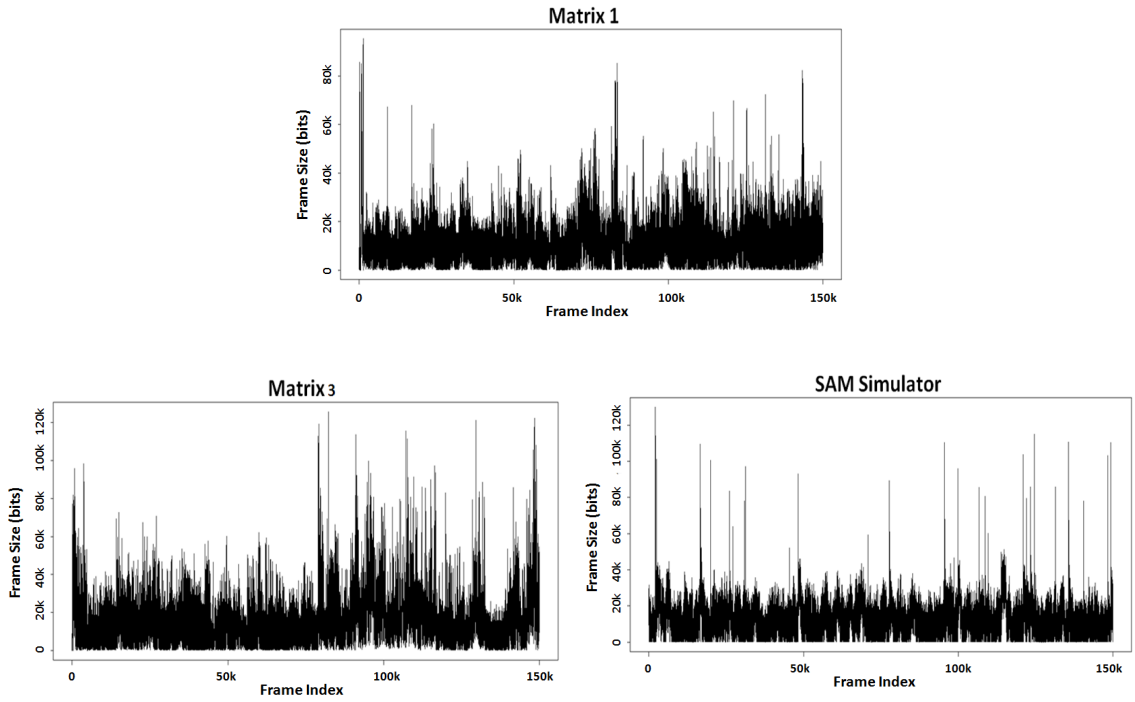


Figure 5-3 Random Shocks Implementation in SAM Video Trace Generator

Another major concern in designing the SAM traffic generator was to ensure that in addition to the mean and range, the generated trace cumulative distribution function (CDF) is within the acceptable range of the other related movies. This should hold true for different trace period lengths. Figure 5-4 shows our results of comparing the frames size CDFs of SAM traffic generator traces against original movie traces for short length traces (5k frames), medium length traces (30k frames), and long traces (150k frames). The introduction of random shocks has the side effect of slightly influencing the distribution of frame sizes while the mean frame size is maintained. As a result, the short length trace generation gives better results than medium and long trace generation processes.

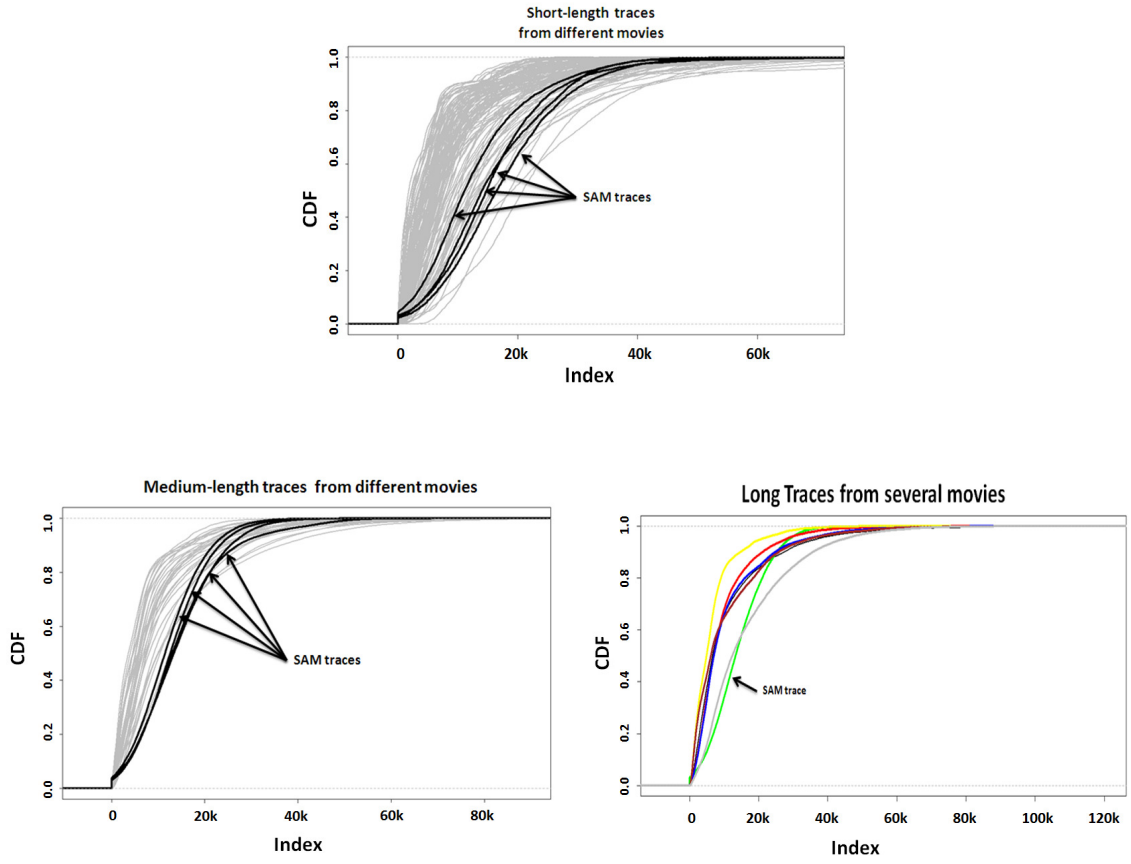


Figure 5-4 CDF Comparisons for Short, Medium and Long Video Traces

The SAM traffic generator can generate any required number of frames and has a stable and reliable performance. We have conducted several trials. With non-optimized code under debug mode we were able to generate 500k frames in less than 6.7 seconds.

The SAM traffic generator described so far produces frame sizes of video frames. In the next section we present the implementation details of SAM RTP traffic generator.

Figure 5-5 shows the graphical user interface (GUI) interface of the SAM frame generator with RTP add-on implemented using C#.NET. Users can easily specify the SAM model coefficients, encoding method, and the length of the video trace to be generated.

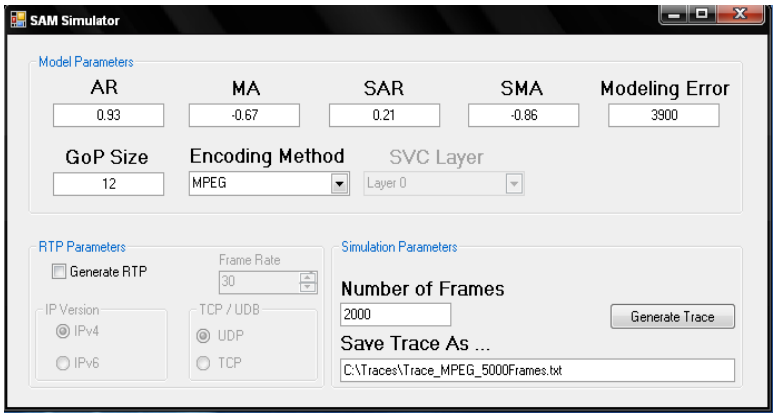


Figure 5-5 Implementation of SAM Video Traffic Generator using C#.NET

5.2 RTP Traffic Generator

Our implementation of the SAM traffic generator allows users to integrate the generated frames with any protocol layer with ease. On most systems, these video frames are transmitted using real time transport protocol (RTP). RTP protocol is defined in RFC 3550 [54]. In this section we present the details of our RTP packet generator based on the SAM model.

RTP packetizing is a very simple mechanism and follows two simple rules: packets can carry data from one video frame only. If the frames are small in size, one can fit as many full frames as the packet size allows. These model rules are illustrated in Figure 5-6.

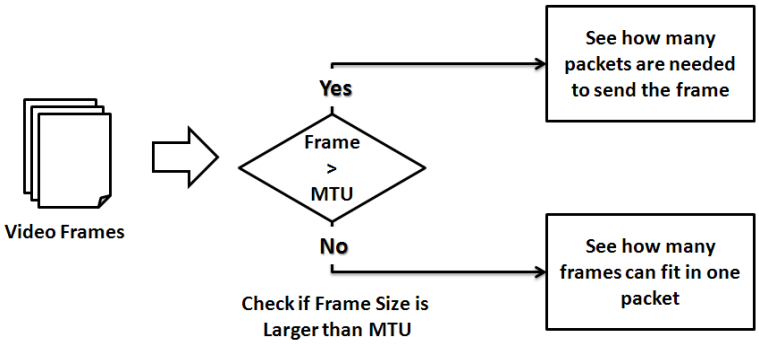
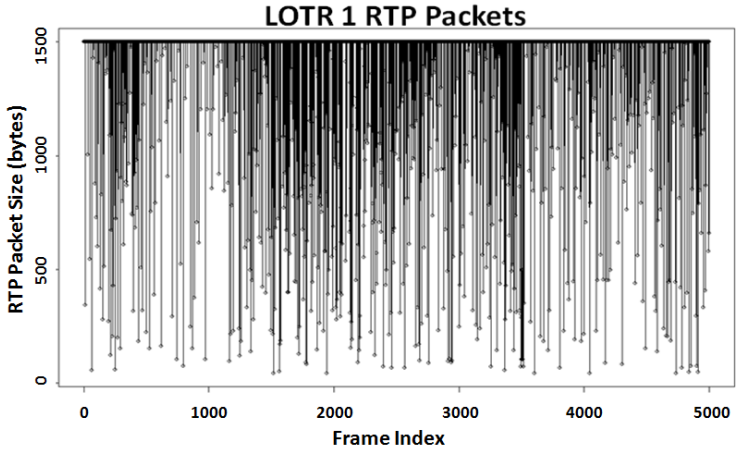


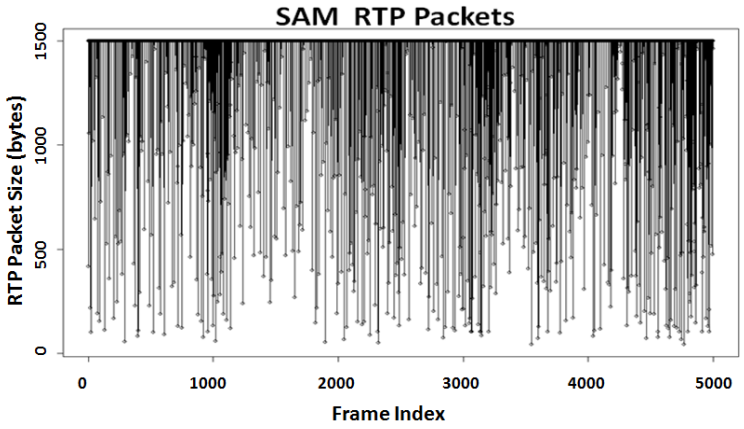
Figure 5-6 RTP Packetizing

We have tested the SAM RTP packet generator against RTP packets generated using original movie traces. The results have confirmed that the generated RTP packets share the same statistical characteristics. Figure 5-7 shows the generated RTP packets from LOTR I movie

trace and the SAM RTP packet generator with an maximum transmission unit (MTU) size of 1500 bytes. Since the same RTP packetizing method had been applied to both original and generated traces, that have been compared before, we omitted the statistical comparison between the two traces to avoid redundancy.



(a) Lord of the Ring I RTP packet



(b) SAM RTP packet

Figure 5-7 RTP Generated Packets Using RTP Packetizer

The SAM RTP packet generator is just an example of what can be integrated to the SAM traffic generator to meet any desired simulation conditions. Other protocols can be implemented as easily, which gives great opportunities to test different standards or custom protocols and to optimize network performance. Figure 5-8 shows the abstract framework of RTP packet generator and its interaction with SAM traffic generator. RTP packet

generator can be replaced with any other protocol layer that can use the generated frames from the underlying SAM traffic generator modules.

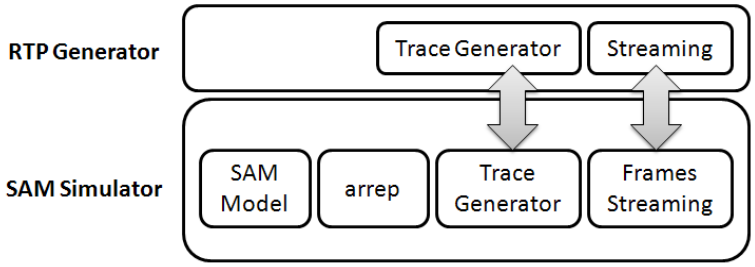


Figure 5-8 RTP Generator and SAM generator Framework

In the next section we demonstrate the design of another developed tool to facilitate using SAM in video traces analysis.

5.3 SAM-Based Video Trace Analyzer

To ease the analysis of video traces and the comparison of SAM model against the original trace, we developed a simple GUI, shown in Figure 5-9, that allows the users to load the video trace frame size information from a text file. SAM analyzer then processes the information and calculates the seasonality of the trace, its SAM parameters, and its AIC value.

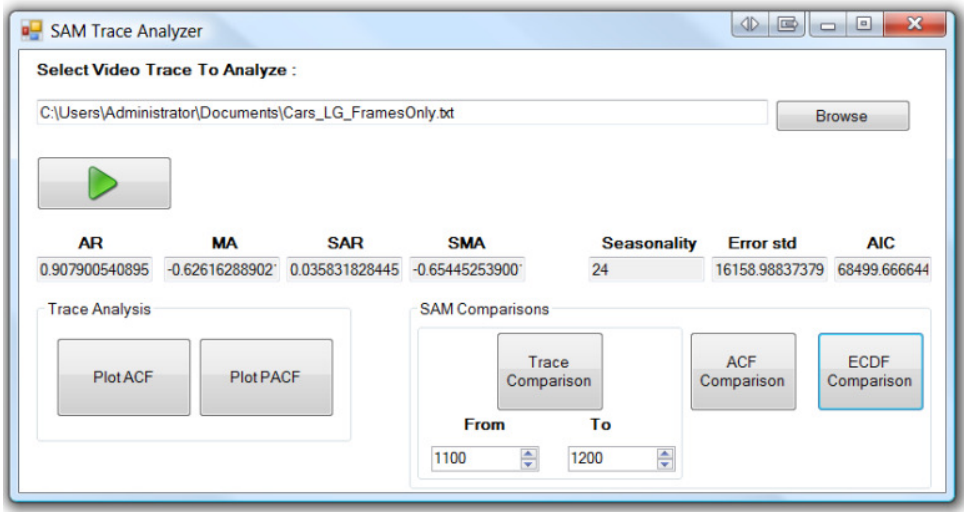
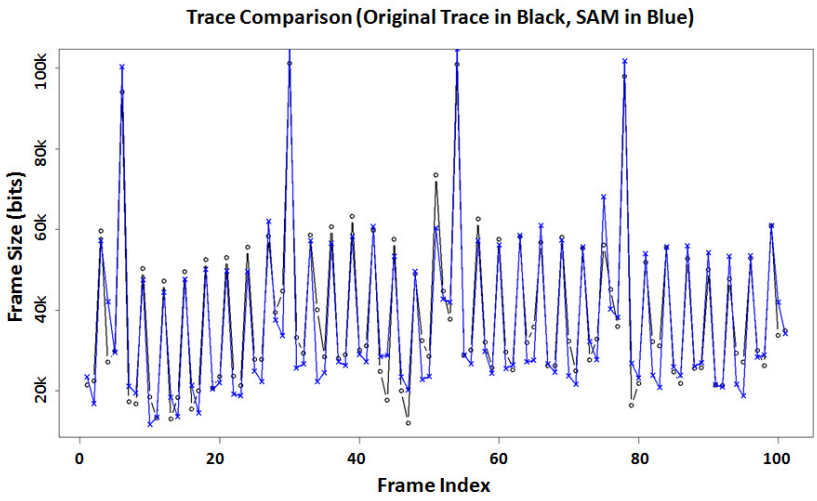


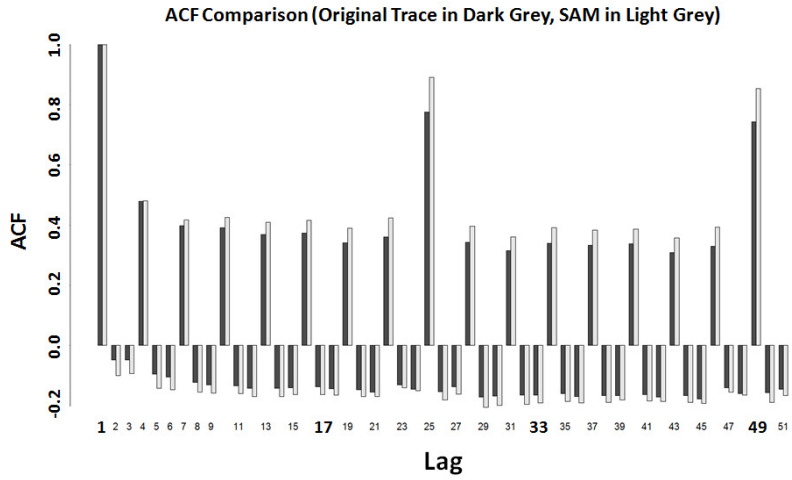
Figure 5-9 SAM-based Video Trace Analyzer GUI

The user can plot the ACF, and PACF graphs of the video trace. In addition, the user can plot video trace, ACF and empirical CDF (ECDF) versus SAM comparison graphs. Figure 5-10 shows an example of the comparison graphs generated by SAM trace analyzer.

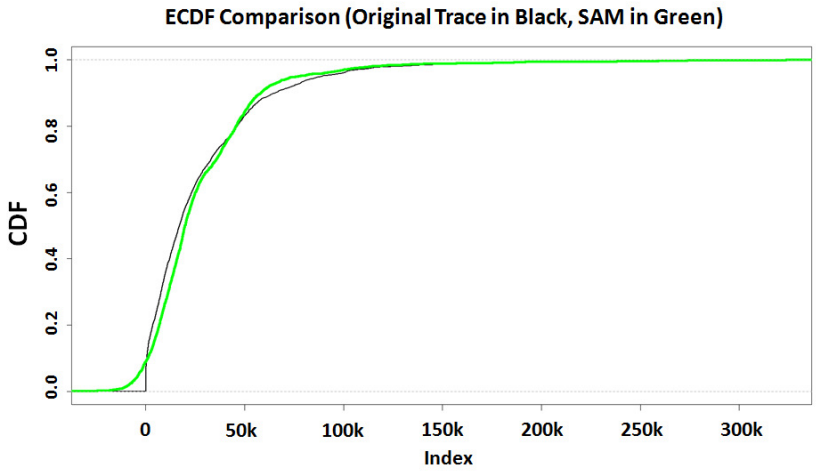
Additional comparisons can be added upon user needs. SAM trace analyzer is implemented using C#. Our implementation provides an interface to R compiled code to allow full utilization of its capabilities.



(a) Trace Comparison (frames between 1100-1200)



(b) ACF Comparison (first 50 lags)



(c) ECDF Comparison

Figure 5-10 An Example of SAM Trace Analyzer Generated Comparison Plots

In the section we discusses the use of SAM video frame generator in WiMAX simulations. We compare in this analysis different mobile WiMAX scheduling algorithms.

5.4 Resource Allocation in Mobile WiMAX Networks

One of the main reasons for our mobile video traffic modeling is to understand and optimize the performance of mobile video over WiMAX networks. In this section, we present the results of our analysis of various scheduling methods for Mobile WiMAX networks using the SAM traffic generator. This analysis illustrates that SAM generator can be used to test and develop new and improved resource allocation schemes.

Mobile WiMAX networks use an orthogonal frequency division multiple access (OFDMA) technique to increase high data rate, cover longer distance and support mobility. In general, the entire channel is divided into multiple subcarriers. The number of subcarriers is proportional to the channel spectral width. These subcarriers are grouped into a number of subchannels. Then, each mobile station (MS) is assigned a group of subchannels for certain amount of time. The time is measured in units of Orthogonal Frequency Division Multiplexing (OFDM) symbol times.

Mobile WiMAX uses a fixed frame-based allocation. Basically, each frame is of 5 ms duration [55]. It starts with a downlink preamble and frame control header (FCH) followed

by a downlink (DL) map and an uplink (UL) map. These maps contain the information elements that specify the burst profile for each burst. The burst profile consists of burst-start time, burst-end time, modulation type and forward error control (FEC) used or to be used in the burst.

Bi-directional communication can be achieved by frequency division duplexing (FDD) in which uplink and downlink use different frequency bands or time division duplexing (TDD) in which the uplink traffic follows the downlink traffic in time domain. All scheduling schemes discussed in this analysis can be used for both FDD and TDD systems. However, to keep the discussion focused, we use TDD throughput this section.

Although the standard allows several configurations such as mesh networks and relay networks, our focus is only on point-to-multipoint network configuration. Thus, the resource allocation problem is basically that the BS is the single resource controller for both uplink and downlink directions for each MS. Each MS has an agreed quality of service (QoS) requirement that is negotiated between the BS and MS at the time of connection setup. The BS grants transmit opportunities to various MSs based on their bandwidth requests and QoS.

In this experiment, we basically focus on how to allocate the number of slots for each MS in each Mobile WiMAX frame. Each slot consists of one subchannel allocated for the duration of some number of OFDM symbols. The number of subcarriers in the subchannel and the number of OFDM symbols in the slot depend upon the link direction (uplink or downlink) and the permutation scheme used. For example, in Partially Used Sub-Channelization (PUSC) permutation scheme, which is commonly used in Mobile WiMAX, one slot consists of one subchannel over two OFDM symbol periods for DL and one subchannel over three OFDM symbol periods for UL [56].

Mobile WiMAX supports several Modulation and Coding Schemes (MCSs), such as Binary Phase Shift Keying (BPSK) and several Quadrature Amplitude Modulation (QAM) schemes. BPSK results in 1 bit per symbol and is used for poor channel conditions. QAM schemes result in more bits per symbol and are used for reliable channel conditions. Since the MCS used for a mobile station depends upon the location of the mobile station and varies with time, the slot capacity (number of bits in the slot) is not constant. Given equal number of

slots, mobile stations at different locations may be allowed to use different MCSs, resulting in different resource allocations. In the study presented here, our focus is on fairness and delay among multiple users and so we assume all users to be in similar channel conditions.

Although Mobile WiMAX supports several classes of services for different kinds of traffic such as voice, video and data. Our focus in this section is the video traffic. The main QoS parameters are the throughput and delay constraints.

In Mobile WiMAX networks, a simple Earliest Deadline First (EDF) scheduling algorithm is generally used to schedule real-time traffic especially video traffic, and Deficit Round Robin (DRR) scheduling algorithm is commonly used to schedule non-real-time traffic [9, 10, 57] in downlink direction. We compared these two algorithms and a combination of the two.

5.4.1 Scheduling Algorithms

In this section, we briefly describe the three scheduling algorithms. These are Earliest Deadline First (EDF), Deficit Round Robin (DRR) and Earliest Deadline First with Deficit Round Robin (EDF-DRR) in the context of mobile WiMAX networks.

5.4.2 Earliest Deadline First (EDF)

Given a set of flows, the first algorithm, EDF [60] simply compares the packets at the head of the flow queues and schedules the packet the earliest deadline constraint. One additional complication in the case of WiMAX is that the entire packet may not fit in the current WiMAX frame and a fragment may be left over. These fragments are transmitted first in the next WiMAX frame. We need to ensure that the fragment will meet the deadline otherwise the entire packet is discarded.

5.4.3 Deficit Round Robin (DRR)

The second algorithm, DRR [58], avoids packet fragmentation by scheduling only a full packet. If a packet will result in exceeding the fair share, the packet is not scheduled and the deficit (amount that would have been allocated) is remembered. However, to fully utilize a WiMAX frame, we use a modified version of DRR, DRR with fragmentation described in [59]. In general, if a packet meets the fair share limit, we schedule it in the current WiMAX frame and if necessary, allow the part that will not fit in the current WiMAX frame to be scheduled in the next frame. This ensures that WiMAX frame capacity is not wasted.

Note that in order to overcome the issue of varying link capacity in Mobile WiMAX networks, the fair share is derived from the queue length and MCS level. Moreover, we use Max-Min fair algorithm to derive the fair share so that the left over space within Mobile WiMAX frame can be used [59].

5.4.4 Earliest Deadline First with Deficit Round Robin (EDF-DRR)

With EDF-DRR, we basically apply EDF first then regulate the packet stream with DRR. In other words, the packets are sorted according to the deadline then DRR is used to decide whether the packet with the earliest deadline is eligible for transmission without exhausting the flow's credits (deficits). Again, we allow fragmented packets to be transmitted for full frame utilization.

5.4.5 Scheduling Algorithms with Enforced Deadline

For real-time traffic, video traffic in particular, received packets with huge delay or over the deadline are not useful. Since the deadline or average delay is negotiated during the connection setup, we can use the deadline information at the scheduler by dropping the packets that are over the deadline and save the bandwidth. Therefore, for all three algorithms described above packets are dropped if it will not meet the deadline after transmission. We analyzed cases without this option, however, the results showed worse performance with a large fraction of packets being discarded at the destination due to exceeding the deadline. We concluded that given the resource constrained nature of wireless medium, any reasonable implementation should minimize waste by discarding late packets before transmission.

5.4.6 Performance Evaluation

In this section, we present simulation results of system throughput, delay, jitter, and fairness for EDF, EDF-DRR and DRR algorithms. We also show the results of those algorithms with deadline enforced. We consider only the downlink resource allocation.

Table 5-1 Performance Evaluation Parameters [55]

Parameters	Values
PHY	OFDMA
Duplexing Mode	TDD
Frame Length	5 ms
System Bandwidth	10 MHz
FFT size	1024
Cyclic prefix length	1/8
DL permutation zone	PUSC
RTG + TTG	1.6 symbol
DL:UL ratio	2:1 (29: 18 OFDM symbols)
DL Preamble	1 symbol-column
MAC PDU size	Variable length
ARQ and packing	Disable
Fragmentation	Enable
DL-UL MAPs	Variable

The simulation configuration and parameters follow the performance evaluation parameters specified in Mobile WiMAX System Evaluation document and WiMAX profiles [55, 61]. These parameters are briefly summarized in Table 5-1. With 10 MHz system bandwidth, 5 ms frame, 1/8 cyclic prefix and a DL:UL ratio of 2:1, the number of downlink symbol-columns per frame is 29 [55]. In PUSC mode, there are 30 subchannels and each slot consists of one subchannel over 2 symbol duration. As a result, there are $30 \times (28/2) = 420$ downlink slots per frame. Of these, 51 slots are required for Frame Control Header (FCH), DL MAP and UL MAP (repetition of 4 and QPSK-1/2) and Downlink Channel Descriptor (DCD) and Uplink Channel Descriptor (UCD) overheads assuming a case of five mobile stations.

In our analysis, interference is represented as a change of MCS. To keep it simple, the MCS level is constant over the simulation period. A lower bit rate MCS indicates more interference than a higher bit rate MCS. In this study, we use only one MCS for the entire movie. It is possible to extend this to a time varying interference, but it would require agreeing to a model for time variation and would create more questions than it would answer. Thus, we have left that for future study. We also selected QPSK-3/4 (9 bytes per slot) for all mobile stations. Therefore, the system throughput for five MSs is around 5.4

Mbps excluding UCD/DCD. Notice that the actual overheads depend on the number of actual burst allocations in both uplink and downlink and other management messages.

5.4.7 Simulation Configurations

We used a modified version of WiMAX Forum's NS-2 simulator [61] in which a Mobile WiMAX module has been added [62]. There are three main simulation configurations in order to show the fairness among MSs and the delay constraint for all three algorithms.

First configuration is an underload scenario with three video flows with 1.35 Mbps average rate each and one Constant Bit Rate (CBR) flow with 3 Mbps. The purpose of CBR flow is to measure the unused space in the frame. We treated the CBR flow as a lower priority so that the CBR flow acquires transmission opportunity only if there is any unused space in the WiMAX frame.

Second configuration is an overload scenario with five video flows with 1.35 Mbps average rate each.

In the third configuration, we used three video flows and one CBR flow; however, one of the video flows is not well-behaved, sending more traffic. For this overloading flow, we used the SAM traffic generator to generate a video stream with an average rate of 3.3 Mbps. Because of the overload, CBR flow does not really get any transmission opportunities in this case.

Although we use three to five flows to show the effect of fairness, the results are expected to be similar with more MSs and higher MCS levels. Note that the video frames were packetized and RTP, user datagram protocol (UDP), and IP headers overheads were added. All video flows' deadlines were set to 20 ms.

All simulations were run from 0 to 10 seconds with 5 seconds of traffic duration. Flows start at 5 seconds end after 10 seconds. There are ranging, registration and connection setup processes during the first 5 seconds.

To quantify fairness, we used Jain Fairness Index [62], which is computed as follows:

$$f(w_1, w_2, \ldots, w_n) = \left(\sum_{i=1}^{n} w_i\right)^2 / n \sum_{i=1}^{n} w_i^2$$

here w_i is the throughput for the i-th MS, for n MSs or n flows. In our simulations, n is 3 or 5. Figure 5-11 shows the simulation topology. The link between BS and MSs is the bottleneck. At the BS, there is one queue for each MS and each queue is 100 packets long.

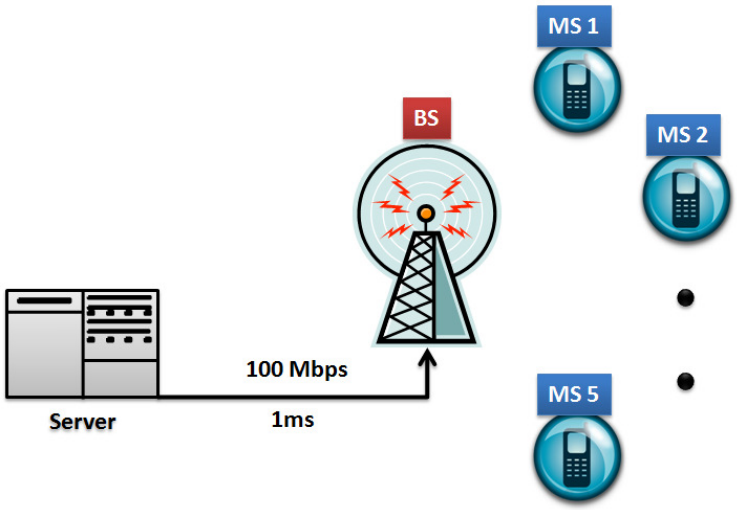


Figure 5-11 Simulation Topology

5.4.8 Simulation Results

In this section, we show system throughput, delay, delay jitter of EDF, DRR and EDF-DRR with and without enforced deadline. For the first *underloaded* scenario, all the algorithms (with and without enforced deadline) result in the same throughput: 1.5, 1.24 and 1.49 Mbps with 1.19 Mbps for CBR. There are no dropped packets for video flows. The average delay ranges from 6 to 7 ms.

Table 5-2 shows results for the second overload scenario with deadline enforcement. Because of the deadline enforcement, average delays for all three algorithms are within the specified deadline of 20 ms plus 5 ms additional transmission delays (duration of the WiMAX frame). EDF is unfair while DRR and EDF-DRR are fair. The degree of fairness of DRR is a bit higher than EDF-DRR, 0.9998 versus 0.9986, respectively. Figure 5-12 also shows the system throughput for all three algorithms over time.

Table 5-3 and Figure 5-13 show our results for the case with one ill-behaved flow. Deadline is enforced for all flows. Again, EDF cannot maintain the share for well-behaved flows. On the other hands, DRR and EDF-DRR can achieve Max-Min fairness for all flows.

Table 5-2 System Throughput, Delay and Delay Jitter with Enforced Deadline (5 Flows in overload Scenario)

(A) EDF

MS	Send (Mbps)	Receive (Mbps)	Delay (ms)	Jitter (ms)
1	1.49	0.90	21.38	1.80
2	1.18	0.76	20.83	2.07
3	1.53	1.14	21.31	1.79
4	1.24	0.74	21.22	1.83
5	1.47	1.11	21.20	1.75

(B) DRR

MS	Send (Mbps)	Receive (Mbps)	Delay (ms)	Jitter (ms)
1	1.49	0.95	18.65	3.84
2	1.18	0.98	16.27	4.30
3	1.53	0.96	19.03	3.46
4	1.24	0.97	16.91	4.31
5	1.47	0.98	18.28	3.86

(C) EDF-DRR

MS	Send (Mbps)	Receive (Mbps)	Delay (ms)	Jitter (ms)
1	1.49	0.89	19.63	3.08
2	1.18	0.95	17.07	4.26
3	1.53	0.85	19.90	3.24
4	1.24	0.89	18.04	4.04
5	1.47	0.88	19.63	3.06

Table 5-3 System Throughput, Delay and Delay Jitter with Enforced Deadline (2 well-behaved Floes and one Ill-behaved Flow in overload Scenario)

EDF

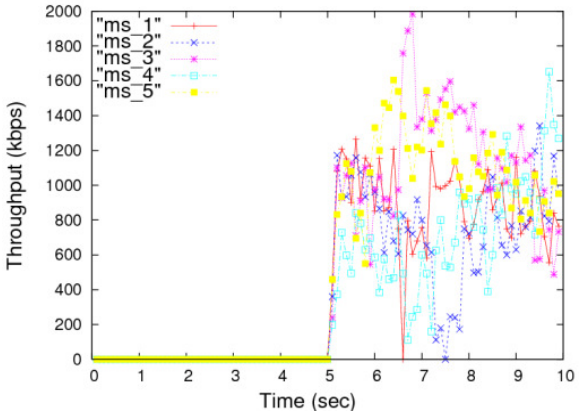
MS	Send (Mbps)	Receive (Mbps)	Delay (ms)	Jitter (ms)
1	1.49	1.19	20.41	1.93
2	1.24	1.03	20.16	1.99
3	3.33	2.66	21.28	1.83

(A) DRR

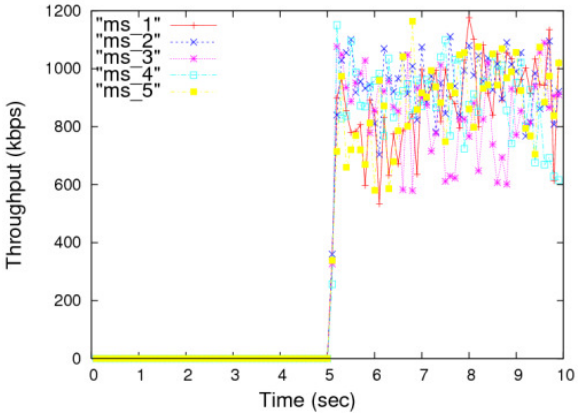
MS	Send (Mbps)	Receive (Mbps)	Delay (ms)	Jitter (ms)
1	1.49	1.49	9.69	3.33
2	1.24	1.24	8.70	3.05
3	3.33	2.34	20.58	2.87

(B) EDF-DRR

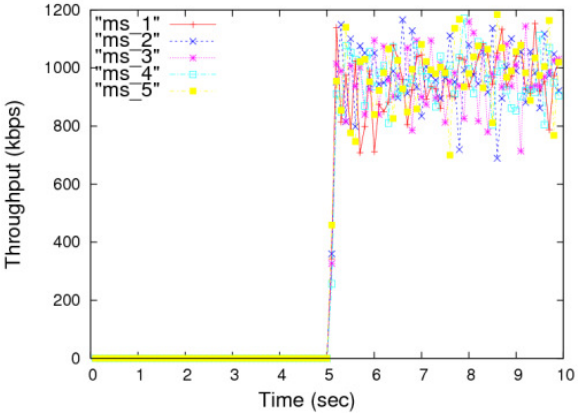
MS	Send (Mbps)	Receive (Mbps)	Delay (ms)	Jitter (ms)
1	1.49	1.49	9.80	3.72
2	1.24	1.24	9.07	3.41
3	3.33	2.34	20.62	2.77



(a) EDF

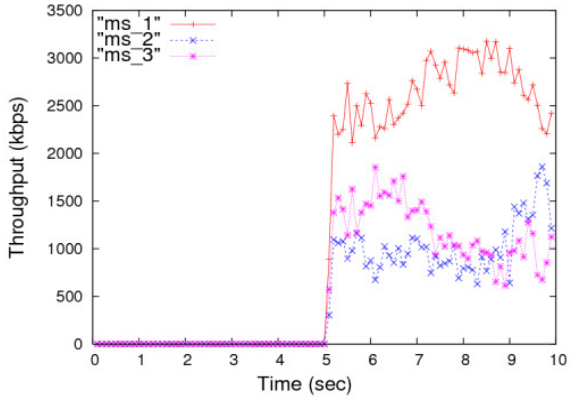


(b) DRR

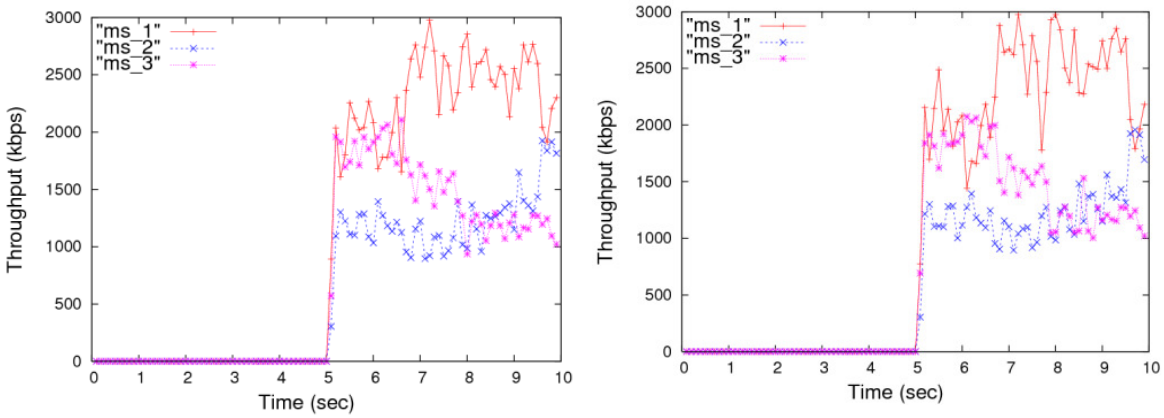


(c) EDF-DRR

Figure 5-12 System Throughput (Five Video Flows in Overload Scenario)



(a) EDF



(b) DRR (c) EDF-DRR

Figure 5-13 System Throughput (Two Well-behaved Flows and One ill-behaved Flow in Overload Scenario)

In this section we discussed the implementation of a video frame generator based on our SAM model. One of the key issues in the development of the generator was the implementation of random shocks that simulate scene changes observed in actual movies. The generator allows easy adjustment of traffic parameters such as average frame rate, average frame size, standard deviation of errors etc.

We also discussed how our RTP packet generator addition to the SAM traffic generator allows producing packet traffic suitable for use in performance studies of mobile video. The SAM frame generator was used to study the resource allocation in a mobile WiMAX network using WiMAX Forum's NS-2 model and WiMAX Forum's system evaluation guidelines.

Given the resource constrained nature of the wireless medium, for mobile video and other real-time traffic, it is important to discard packets that exceed the deadline before transmission on the wireless medium. The simulation results for EDF, DRR, and EDF-DRR show that EDF is most unfair, EDF-DRR is less unfair. DRR is fair and provides the best performance for real-time mobile video traffic.

In the next chapter we discuss our results of encoding and analyzing over 50 AVC encoded HD video traces.

Chapter 6

6 Statistical, Modeling, and Prediction Analysis of HD Video Traces

In this chapter, we present our work to analyze, model, and predict high-definition (HD) video traces encoded with the H.264/AVC codec. These video traces are encoded with higher resolution and quality than previously studied. We compare three modeling methods: autoregressive (AR), autoregressive integrated moving average (ARIMA) using the automated approach proposed in [63], and our simplified seasonal ARIMA (SAM) model that was developed for the less resource demanding mobile video traces [64, 65]. In addition we compare these models in their prediction accuracy.

There have been several contributions that aimed to achieve a better understanding of the relationship between the statistical characteristics of video traces and their impact on data networks. In [66], the authors presented a statistical and factor analysis study of 20 MPEG1 encoded video traces and their impact on ATM networks. Similar approaches were presented in [67] with emphasis on video trace frame size distribution. The author in [68] performed a statistical analysis on four MPEG-4 AVC encoded video traces demonstrating the quantization effects over several statistical measurements, including the inter-correlation between video frames. In [18], the authors compared the statistical characteristics of AVC standard versus its predecessor, viz., MPEG-4 Part2 in terms of bit rate distortion performance, bit rate variability and long range dependence (LRD).

In this chapter, we present our work of analyzing and modeling over 50 HD video traces from YouTube HD videos section. We aim to investigate the main statistical characteristics that define a HD video trace. This identification is important for two main reasons: it helps in clustering video traces depending on certain statistical criteria to help choose the correct traffic workload, or in other possible data mining processes. Additionally, it helps define the main statistical attributes of HD video traces that should be considered to achieve a valid statistical model.

One of the main challenges in developing a valid video workload model is the availability of an adequate number of traces to test the proposed model. The available traces on the web are scarce and do not represent all the different types of videos. Thus, one of the aims of this contribution is to provide researchers with a sufficient number of traces to support their future studies. All our tools, results and video traces are available through our website [69].

Figure 6-1 summaries the main steps taken in analyzing and modeling the selected videos and shows each step's corresponding outputs.

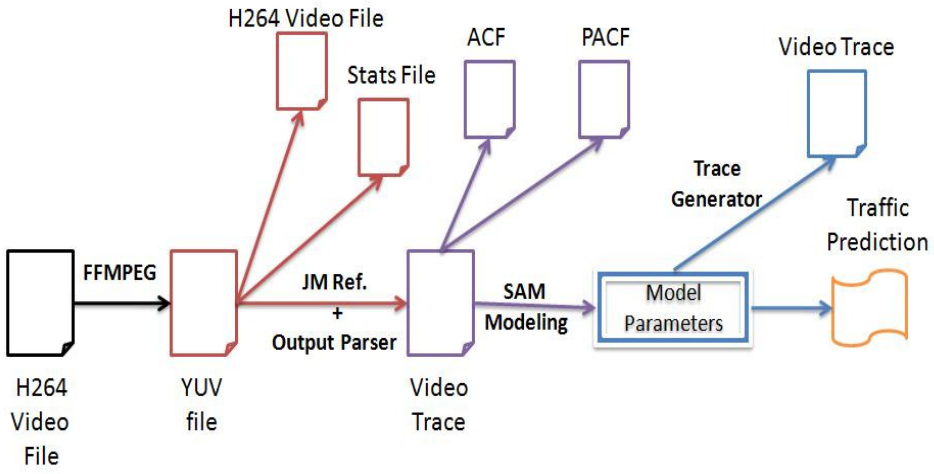


Figure 6-1 Modeling, Analyzing, and Generation of Video Traces Processes

Our encoding process starts with an HD YouTube video in *mp4* format, which is then converted to a YUV (4:2:0) raw video. The raw video is consequently encoded with AVC, and the process produces the following: an encoded movie file, its encoding statistics file, and a full verbose description of the encoding process. The verbose output is then parsed using our analysis tool to get the video trace information, which is then modeled using AR, ARIMA or SAM. The video trace is used also to produce the video frames' autocorrelation function (ACF) and the partial autocorrelation function (PACF) graphs.

The SAM parameters for each video can be used in either video traffic prediction analysis, or in generating video traces. SAM frame generator uses these parameters to generate a movie trace that is statistically close to the original movie trace.

This chapter is organized as follows: Section 6.1 discusses the methodology of obtaining and encoding our collection of HD videos. Section 6.2 shows the results of our statistical

analysis, including both factor and cluster analysis. In Section 6.3, we compare the results of modeling the video traces, and provide a simple introduction to SAM. Section 6.4 discusses the approach to evaluate the prediction accuracy of the compared models and the comparisons results. Finally, we conclude this chapter and give some insight to the impact of our results.

6.1 Encoding YouTube HD Videos

To represent real life video traffic load, we chose YouTube website as our source. YouTube is currently the most popular video streaming website on the Internet [71]. Our first step in selecting the candidate videos from YouTube was to make sure that we have a good variety of both texture/details and motion levels. To select a representative group of the available videos, we started our selection process with some of the most visited videos in YouTube HD section [72]. Then, we increased our collection by selecting three random videos from each of the 15 subcategories available for YouTube website's users. In total we have collected 54 video files in mp4 format.

Then, we analyzed the collected videos using *MediaInfo* [73] to determine the encoding parameters for the various videos and to select the most commonly used parameter values. We made sure that the parameter values we selected were consistent with those recommended in [45, 74] for YouTube video encoding. Our next step was to convert all these videos to raw or YUV (4:2:0) format. This step is important to ensure unified encoding parameters for all the collected videos to allow objective comparisons. We performed the conversion process using the open source coding library FFMPEG [75].

To convert YUV files to the H.264/AVC format, we tested two publically available encoding libraries: x264 [76] and JM reference software [77]. Though x264 is significantly faster than JM reference software, it provided us with less information about the encoding process. Table 6-1 lists the main encoding parameters used with JM reference software.

Table 6-1 Encoding Parameters for The Selected YouTube Video Collection

Encoding Parameter	Value
FrameRate	24
OutputWidth	1280
OutputHeight	720

ProfileIDC	100 (High)
LevelIDC	40 (62914560 samples/sec)
NumberBFrames	2
IDRPeriod	24
NumberReferenceFrames	3
QP (Quantization Parameter)	I=28, P=28,B=30

These parameters were chosen to represent the majority of the videos we have collected. We used in our encoding process Instantaneous Decoding Refresh (IDR) frames. IDR frames are special type of I frames that allow better seeking precision and thus enhance the user's experience. We used closed-GOP setting [74] to ensure that all I-frames are IDR frames, hence improving the user's online experience. The majority of the collected videos have a frame rate of 24 fps.

The *ProfileIDC* parameter defines the video profile, which, in this case, is set to high. This parameter, along with the *LevelIDC* parameter specifies the capabilities that the client decoder must have in order to decode the video stream. Parameter *NumberBFrames* specifies the number of B slices or frames between I, IDR and P frames. The quantization parameters used are the default values for the encoder. The parameter *NumberReferenceFrames* sets the maximum number of reference frames stored in the decoder buffer, and it is set to three frames. All other encoding parameters are set to the default values of JM reference software. In the course of our analysis and encoding processes, we used two versions of JM reference software: v15.1 and v16.0.

The encoding procedure is both time and resource consuming process. The encoding of a single video file took on average 37 hours, with an average encoding rate of 0.02 fps. The average size of a raw YUV (4:2:0) video file is around 4 GB. These figures support our conviction of the necessity to have a valid trace model and generator. The output of the encoding process is then run through our parser to extract the video trace frame size information needed for the next steps of our analysis and modeling.

6.2 Factor and Cluster Analysis of Video Traces

In this section we discuss the steps taken to perform a full statistical analysis of the collected video traces in order to achieve a better understanding of the main factors that can be used to represent a video trace in order to develop a representative statistical model.

Multivariate analysis is used to reveal the full structure of the collected data, and any hidden patterns and key features [78]. Multivariate analysis is used especially when the variables are closely related to each other, and there is a need to understand the underlying relationship between them. We have computed the following statistical quantitative values for traces frame sizes: mean, minimum, maximum, range, variance, standard deviation, the coefficient of variance, and the median value. In addition, we computed the Hurst index value ,as we discussed earlier, which indicates the video sequence's ability to regress to its mean value, with higher values indicating a smoother trend, less volatility, and less roughness. Its value varies between 0 and 1. This is also an indication of the strength of the long range dependence (LRD) between video frames. The Hurst index in computed as we shown in Chapter 4.

We also computed the *skewness* value that represents the symmetry of the observed distribution around its center point:

$$Skewness = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^3}{(N-1)\times\sigma^3}$$

here σ is the standard deviation of the frames sizes. Additionally, we computed the kurtosis value, which is an indication whether the observed video trace distribution is peaked or flat relative to a normal distribution. The kurtosis equation as follow:

$$Kurtosis = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{(N-1)\times\sigma^4}$$

Table 6-2 Range of Statistical Values for YouTube Video Collection

	Mean	Range	Variance	Hurst	Coefficient of Variance	Median	Skewness	Kurtosis
Max	83340.43	1198416	13767760363	0.902836	3.9860815	62748	6.58066	61.34631
Min	9782.01	65576	154362485	0.498937	0.6875022	448	0.2287191	1.64370

As Table 6-2 shows, the collected videos represent a statistically diverse data samples. And as we mentioned before, the video frame size of HD videos has a high variance. The table shows the most important statistical variables that have been collected. We noticed through our preparation analysis that the *min* variable does not contribute to the total variance significantly, and thus it was disregarded. Both *max* and *range*, and *variance* and *standard*

deviation pairs are almost identical. We picked range and variance to represents the two pairs respectively. In the next sections, we will discuss the methodology and results of performing both factor and cluster analysis.

6.2.1 Principal Component Analysis

One of the most common factor analysis methods is principal component analysis (PCA), where a group of possibly related statistical variables are analyzed and then reduced to a smaller number of uncorrelated factors. These factors accounts for most of the variance in the observed variables. By performing this process, we aim to minimize the number of variables to represent a video trace without much loss of information [78].

Our first step is to determine the smallest number of statistical variables to represent each video trace. Table 6-3 shows the correlation between the selected variables. These statistical variables collectively represent the majority of the samples variation.

TABLE 6-3 CORRELATION BETWEEN THE SELECTED VARIABLES

	Mean	**Range**	**Variance**	**Hurst**	**Coefficient of Variance**	**Median**	**Skewness**	**Kurtosis**
Mean	1	0.48	0.73	0.48	-0.40	-0.9	-0.36	-0.23
Range	0.48	1	0.74	0.34	0.19	0.25	0.51	0.6
Variance	0.73	0.74	1	0.36	0.13	0.41	0.13	0.14
Hurst	0.48	0.34	0.36	1	-0.44	0.41	0.25	0.17
Coefficient of Variance	-0.40	0.19	0.13	-0.44	1	-0.56	0.71	0.51
Median	-0.9	0.25	0.41	0.41	-0.56	1	-0.49	-0.33
Skewness	-0.36	0.51	0.13	0.25	0.71	-0.49	1	0.93
Kurtosis	-0.23	0.6	0.14	0.17	0.51	-0.33	0.93	1

The importance of each factor is represented by its *eigenvalue*. To determine the number of factors to extract we used Kaiser-Guttman rule [79]. By following this rule, we excluded the factors with eigenvalue less than 1. We supported our selection by performing the *Scree* test [80] as shown in Figure 6-2, where we plotted the relationship between the number of factors and their cumulative contribution to the total variance of the data set, and we looked for either large spaces between the plotted variables or a knee in the graph to determine the number of factors to be considered.

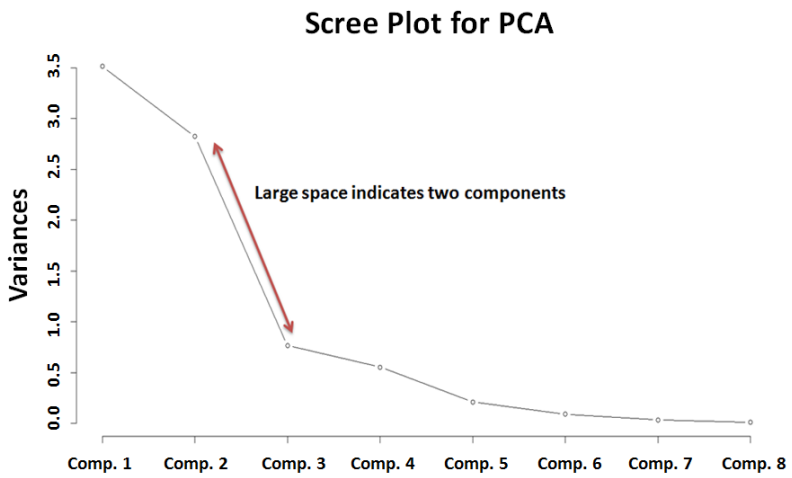


Figure 6-2 Scree Plot for the HD Video Collection Data based on the Eight selected Variables which Indicates Two Principal Components

Our analysis resulted in choosing two factors with the following eigenvalues: $\lambda 1 = 3.51$, and $\lambda 2 = 2.82$. These factors account for 79% $[(\lambda 1 + \lambda 2) / 8]$ of the total standardized variance. We confirmed that these two factors are sufficient to explain the inter-correlations among variables by performing several non-graphical tests [81].

To simplify the factor structure and spread out the correlations between the variables and the factors (their loadings values) as much as possible, we performed both orthogonal and oblique rotations on the factors [82]. We chose *varimax* orthogonal rotation as it gave the best results. As shown in Figure 6-3, the two significant groups are the mean and skewness groups. Table 6-4 shows the loadings values for both varimax rotated and un-rotated factors.

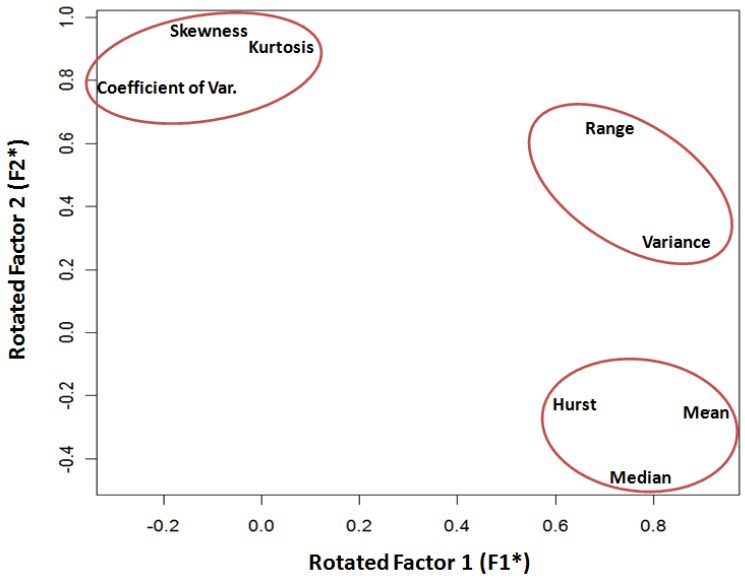


Figure 6-3 Scatter Plot of Varimax Rotated Factors F1* and F2* in the Space of the Two Principal Components

Table 6-4 Estimated and Rotated Factor Loadings

	Estimated		*Rotated (varimax)*	
	F_1	**F_2**	**F_1***	**F_2***
Mean	0.84	0.46	**0.93**	-
Range	-	0.95	0.73	0.62
Variance	0.39	0.80	**0.84**	-
Hurst	0.62	-	0.64	-
C. Var	-0.75	0.35	-	0.77
Median	0.87	-	0.77	-0.46
Skewness	-0.75	0.62	-	**0.97**
Kurtosis	-0.62	0.67	-	**0.91**

As can be noticed, the rotated factors are better spread out and simpler to interpret. From Table 6-4 we can note that the first factor F1* defines mainly *mean* and *variance* values. The second factor defines mainly *skewness* and *kurtosis* values. We chose the *mean* to represent the first factor since it has the highest load. We chose *kurtosis* as a representative of F2* since it has the lowest correlation between it and the mean (-0.23). This analysis shows the importance of *skewness* and *kurtosis* in HD videos traces. These two variables were considered irrelevant in a previous video analysis [66]. This realization can be explained by the dependence of these variables on the standard deviation that accounts for a significant proportion of the total variance of HD videos traces.

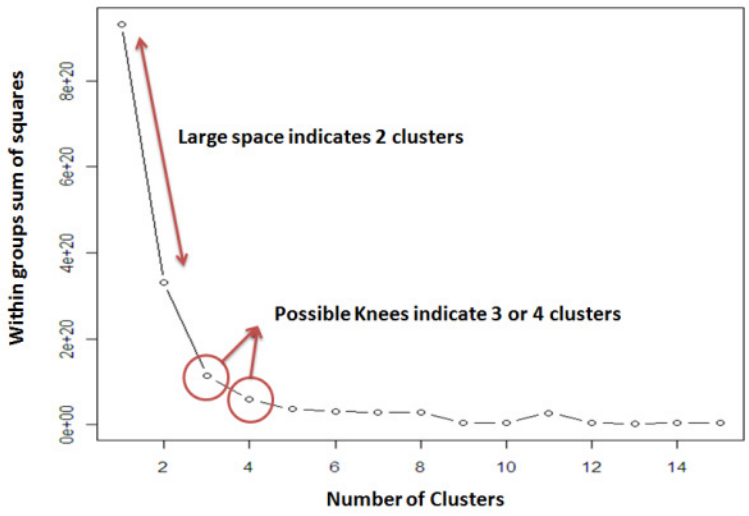
6.2.2 Cluster Analysis Using K-means Clustering

We have demonstrated that the selected two factors, or principal components, are sufficient to characterize a HD video trace. The second step of our analysis is to group the collected video traces into clusters. We used one of the most popular clustering methods: *k-means* clustering algorithm [83]. K-means algorithm achieves clustering by minimizing the within-cluster sum of squares as shown below.

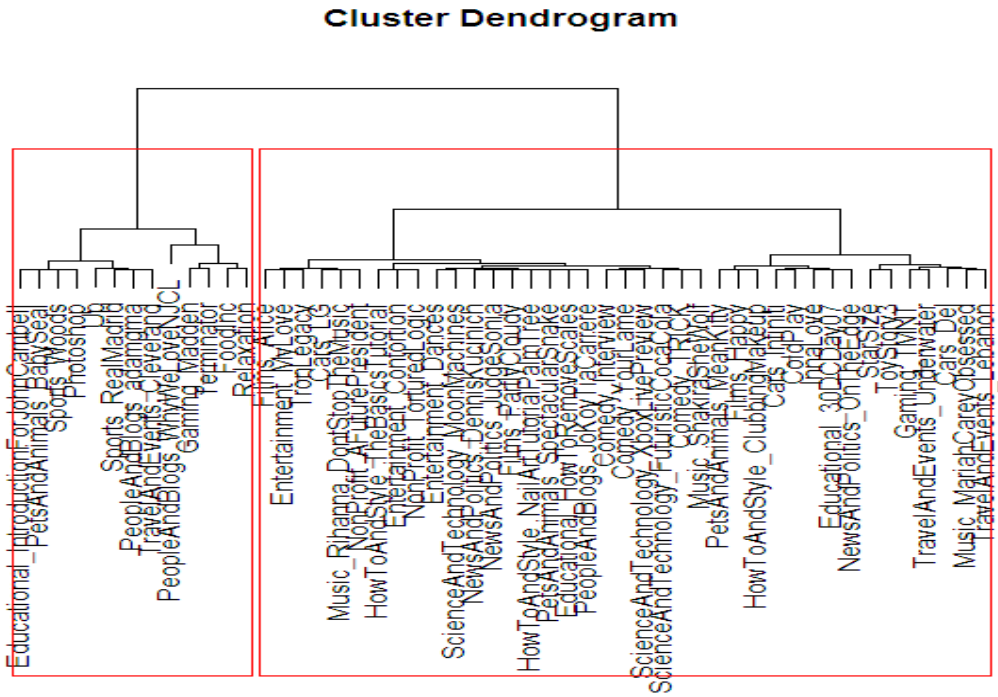
$$\arg\min_{s} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2$$

where x_i is the video trace size at index i, k is the number of sets ($k < n$, n: number of video traces), S_i is the i-th set, and μ_i is the mean of S_i.

Our next step is to estimate the number of clusters or groups to consider for k-means clustering. PCA helps give an insight of how many clusters the video traces can be grouped into [84]. In our case, PCA suggests that we need two clusters. In order to verify the analysis results from PCA, we proceeded with computing the within-cluster sum of squares for different number of clusters. Our aim is to select the minimum number of clusters that allow the minimal possible value for the within-cluster sum of squares. By plotting these values to represent a graph similar to the previously shown *scree* test in Figure 6-2, the large spaces between the plotted variables and the graph knee values indicate the possible values are: two, three, and four clusters as shown in the Figure 6-4 (a). To further investigate the best possible number of clusters to use, we performed a hierarchical clustering to identify the number of clusters using Ward's method [83]. As shown in Figure 6-4 (b), the video traces are divided into two main clusters. Our choice of grouping the video traces into two clusters was further verified by performing silhouette validation method [85].



(a) Within Groups Sum of Squares vs. Number of Clusters



(b) Hierarchal Clustering Result

Figure 6-4 Determining Number of Clusters using Scree Test and Hieratical Analysis

By performing *k-means* clustering we grouped the video traces into 2 clusters. Table 6-5 shows the two chosen principal components corresponding to the centriods of the two

clusters, and the two clusters main members. Figure 6-5 shows the distribution of video groups over the two clusters.

Table 6-5 Clustering Results Using K-means Clustering

Variables	Cluster 1	Cluster 2
Mean	59,251	32,582
Kurtosis	12.028829	9.099512
No .of Elements	13	39
Main Video Groups	Films, People and Blogs, Sports, Educational	Films, Music, News, Comedy, Cars

In summary, video traces that belong to cluster 2 have significantly lower mean values, and have considerably low peaks compared to normal distribution, and lighter tails as indicated by their low Kurtosis values.

We also notice that films category video traces are spread across both clusters. Most blogs and sport category videos are characterized as peaky video traces because of their content. News and comedy videos are less peaky and have lower means than other movies.

To summarize, in this section, we demonstrated our results of performing both factor and cluster analysis on our collection of video traces. Both methods of analysis give us a better understanding of the distribution of the movie traces and their key statistical attributes.

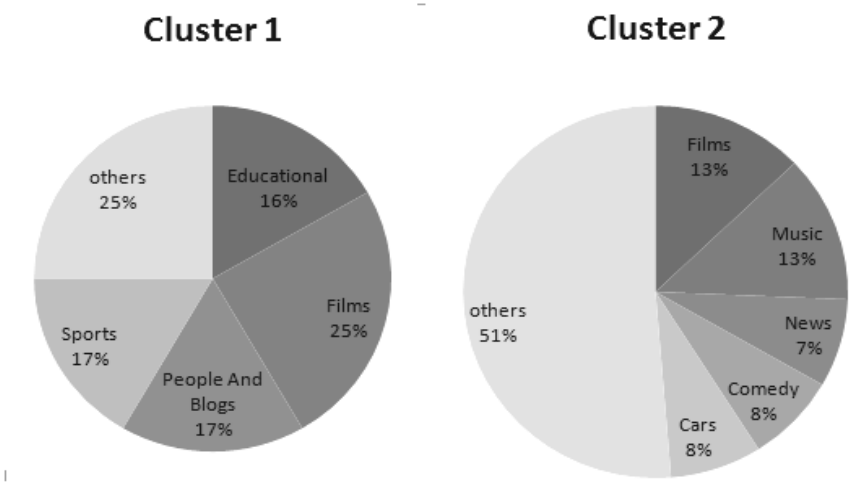


Figure 6-5 Distribution of Movie Groups over The Two Clusters

6.3 HD Video Traces Collection Modeling Results

In this section discuss and compare three statistical models to represent HD video traces. Several models to represent Variable Bit Rate (VBR) MPEG traffic have been proposed in the recent years. Some of the models proposed are based on Markov chain models, which are known for their inefficiency in representing the long range dependence (LRD) characteristics of MPEG traffic [25, 46]. Due to the high influence of LRD, multiplicative processes have been considered like Fractional ARIMA (FARIMA), which have been shown to be accurate, but also computationally demanding and provide marginal improvements over ARIMA [22]. Wavelet-based prediction has been shown to require more computation resources, and providing less accurate results than ARIMA [86]. Our aim is to select a simple to implement, accurate and applicable model for all video traces without the need of significant statistical background.

The chosen model should not require video-specific, complex and tedious steps. The model should be able to not only represent video frame size distribution, but also the correlation between the frames. These attributes are important to achieve the desired results and to allow the analysis of our large collection of video traces. This pre-analysis step resulted in choosing three modeling methods: autoregressive (AR) model, seasonal autoregressive integrated moving average (SARIMA) model using the approach proposed in [63], and SAM model [64, 65]. All these models use maximum likelihood estimation to determine the model terms coefficients, and consider Akaike's Information Criterion (AIC) as their optimization goal. AIC can be computed as we explained in Chapter 4. AIC, as mentioned before, defines the goodness of fit for the models, considering both their accuracy and complexity defined by their number of parameters. Lower AIC values indicate better models in terms of their validity and simplicity. Each of the modeling methods is described briefly below.

6.3.1 AR Modeling

Autoregressive fitting takes into consideration the previous values of the fitted trace. An autoregressive model of order p can be written as:

$$x_t = \sum_{i=1}^{p} \varphi_i x_{t-i} + e_t$$

where φ_i is the i-th AR model parameter, and e_t is white noise. We used maximum likelihood estimation (MLE) to estimate the model parameters of the AR model. Using AR to fit the video traces is a considerably simple process, but it does not always yield accurate results. Additionally, each video trace has its own set of parameters in terms of their numbers and their coefficients values.

6.3.2 ARIMA Modeling using Automatic Approach

We used the automatic SARIMA estimation algorithm proposed in [63], which implements a unified approach to specify the model parameters using a step-wise procedure. It also takes into consideration the seasonality of the trace to allow representing seasonal data series. This approach also results in a separate set of parameters for each video trace in terms of their numbers and their values. For the rest of this chapter we will refer to this approach simply as ARIMA.

6.3.3 SAM Model

SAM provides a unified approach to model video traces encoded with different video codec standards using different encoding settings [64, 65]. The model was developed to model mobile video traces, and we investigate in this chapter its ability to model more resource-demanding HD video traces with higher resolutions and different encoding settings. SAM parameter values are determined using maximum likelihood estimation and optimized using Nelder-Mead method [87]. The four parameters are the coefficients of: autoregressive, moving average, seasonal autoregressive, and seasonal moving average parts.

6.4 Modeling Results

In this section we discuss the results of our modeling analysis performed over our collection of more than 50 HD video traces. We started with evaluating the achieved AIC results, first by simply comparing the sum of the AIC values for all the modeled video traces. Then, we calculated the number of video traces in which each model has scored the best AIC, i.e. lowest value. Additionally, we compared the three models using three statistical measures to evaluate how close the models values to the actual traces: the mean absolute error (MAE), mean absolute relative error (MARE), and root mean square error (RMSE). MAE and MARE are computed as discussed before in Chapter 4.

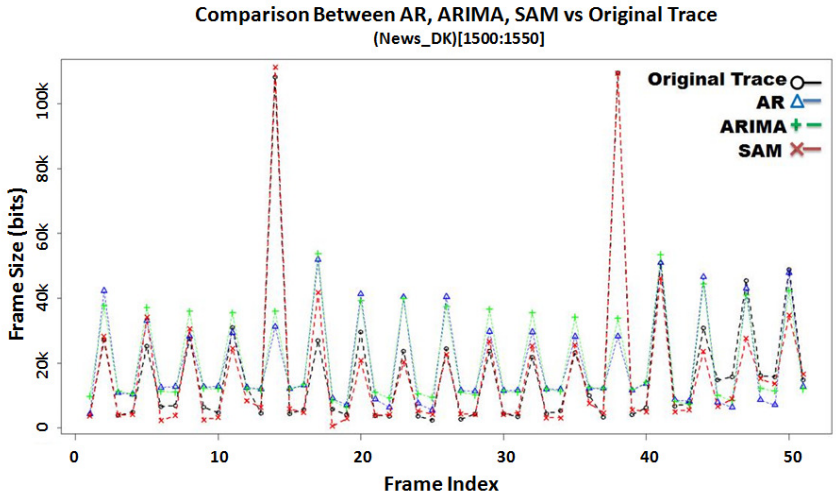
$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_i)^2}$$

where N is number of frames, and e_i is the modeling error at the i-th frame. The results are shown in Table 6-6. It can be noted that SAM achieved the best results, while AR and ARIMA came in second and last place respectively. The achieved results demonstrate two main points: SAM is superior to the other two modeling methods, and that the automated approach used with ARIMA modeling does not always yield the expected results.

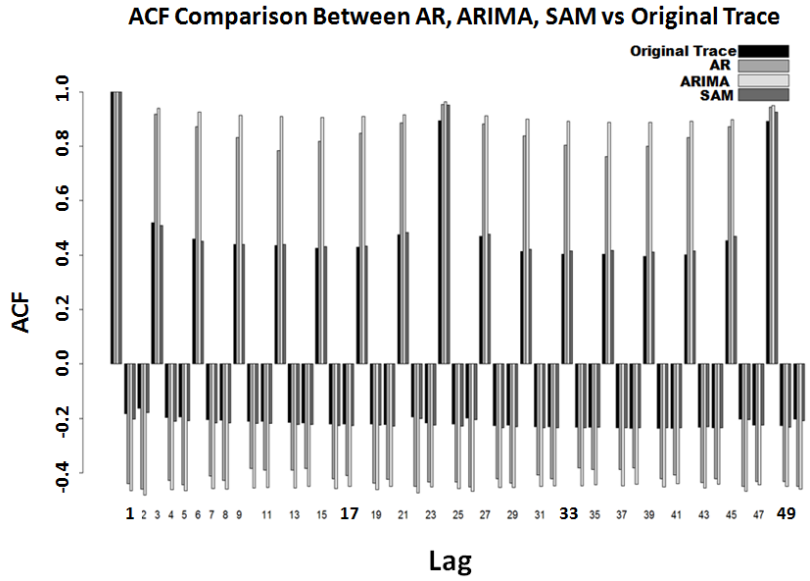
Table 6-6 Comparison between AR, ARIMA, and SAM Using AIC, MAE, MARE and RMSE

	AR	**ARIMA**	**SAM**
Total MAE	830753	894700	641897
Total MARE	200.12	220.47	126.28
Total RMSE	1583607	1644015	1114846
Total AIC	3473929	3492401	3344490
No. of Best AIC	6	3	43

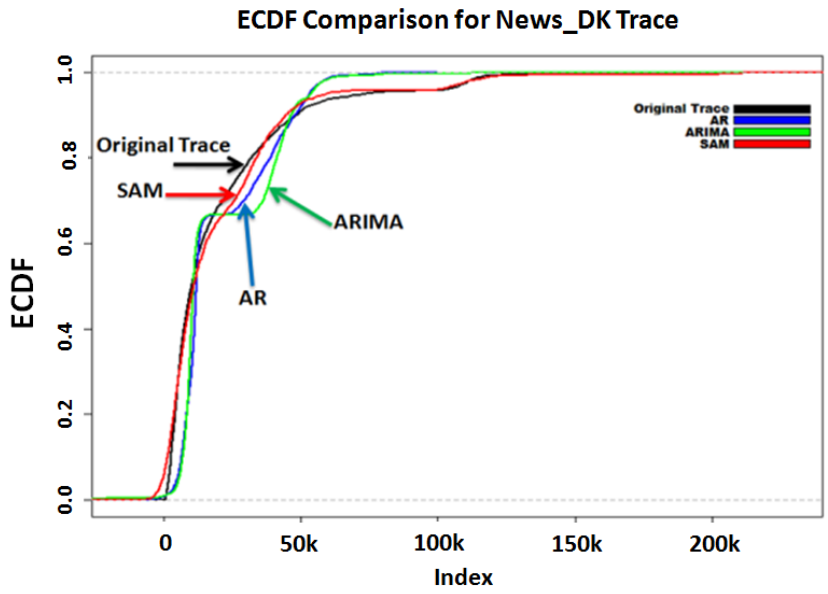
Additionally, we performed several graphical comparisons for all the video traces by comparing the original video traces, their auto correlation function (ACF) plots, and their empirical cumulative distribution function (ECDF) plots to ones achieved by the different models. Figure 6-6 shows an example of one of the compared video traces. As we can notice, SAM has better results and represents the traces statistical characteristics better than the other two models. For this example, modeling using AR required 12 parameters, using ARIMA required 7 parameters (two AR parameters and five MA parameters), and using SAM required only 4 parameters.



(a) A Close-up Trace Comparison (frames between 1500-1600)



(b) ACF Comparison (first 50 lags)



(c) ECDF Comparison

Figure 6-6 Graphical Comparisons Between AR, ARIMA, and SAM

The results show that SAM has a significant advantage over the other two modeling methods especially on the seasonal transition of the video trace. This advantage is also apparent in ACF and ECDF plots comparisons. All graphical comparison results for all the HD video traces are also available through our website [69].

6.5 Forecasting HD Video Traffic

Because of the variability exhibited in video traffic and especially in AVC encoded videos, static bandwidth allocation is considered not suitable to provide high utilization of the network resources. Thus, dynamic bandwidth allocation has been considered as an alternative approach [88]. The heart of the dynamic bandwidth allocation schemes is a traffic predictor that helps in making decisions for future bandwidth allocations.

In order to evaluate the different prediction methods, we characterize different requirements for the predictor in which to operate. These requirements are set to test the abilities of these models to operate under different network traffic scenarios. The first criterion is the model's ability to correctly estimate the traffic and to its capability of achieving long term prediction. The prediction process itself consumes network resources. Thus, it is preferable to run the predictor as few times as possible. On the other hand, we do not need the prediction interval

to be too large, because the video frame sizes changes frequently and do not follow a certain pattern for a long period of time that may results in severe prediction errors. Prediction errors results in either in inefficient use of network resources, or result in increased rate of dropped packets due to insufficient space in the receiving network buffers. We evaluate this criterion by comparing the three modeling methods using four different prediction interval lengths: 48, 72, 96, and 120 frames that translate to 2, 3, 4 and 5 seconds, respectively.

The second criterion is the ability of the predictor to capture the statistical characteristics of the movie trace by analyzing as few video frames as possible. We evaluate this criterion by comparing the prediction accuracy in the cases where the predictor has already processed 250, 500, 1000, and 1500 video frames. This translates to 10, 20, 40, and 60 seconds, respectively.

Evidently, we seek out the best predictor that can achieve the best prediction accuracy for the longest prediction window with the least number of frames to be analyzed. We chose the commonly used noise to signal (SNR^{-1}) ratio as our prediction accuracy metric. SNR^{-1} computes the ratio between the sum of squares of the prediction errors, and the sum of squares of the video frame sizes. SNR^{-1} can be computed as illustrated before in Chapter 4.

Figure 6-7 shows a summary of the main results. As seen in this figure, the prediction error is directly related to the increase of the prediction window size. It also shows that the increase of the predictor knowledge, as represented in the number of frames processed, provides better prediction accuracy. It is obvious from the figure that SAM provides significant improvements over the other two methods. Table 6-7 shows the improvements SAM provides over AR and ARIMA when 1000 frames are processed. SAM improves up to 55% over AR, and 53.3% over ARIMA.

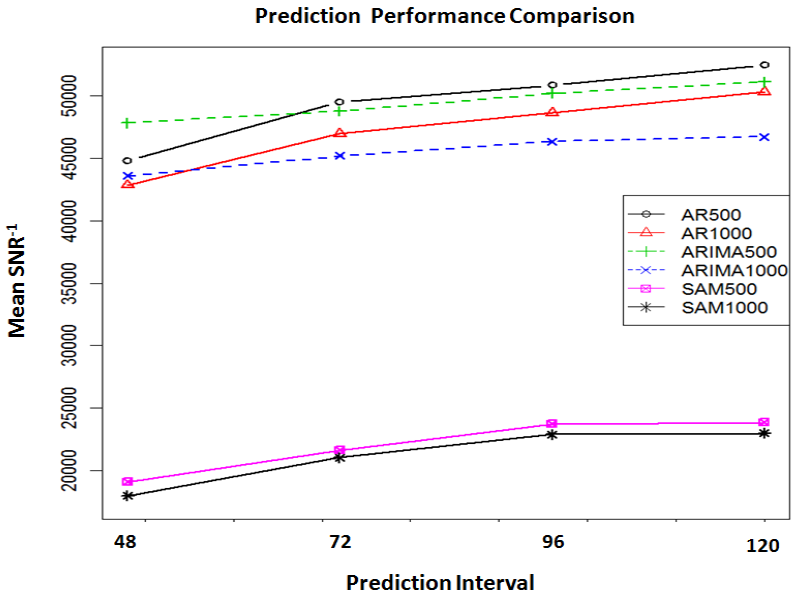


Figure 6-7 Comparisons between AR, ARIMA, and SAM in their SNR^{-1} Values

Table 6-7 SNR^{-1} Comparison between AR, ARIMA and SAM

	AR 1000	ARIMA 1000	SAM 1000
SNR^{-1} (avg)	47180	45457	**21220**
Improvement over AR	-	3.6 %	**55 %**
Improvement over ARIMA	-3.6 %	-	**53.3 %**

To better understand the reasons behind the observed improvement, we plot the three models predictions for a prediction window of 48 after processing 1000 of video frames. As shown in Figure 6-8, SAM not only manages to predict the video frames accurately, it is the only one that can predict the significant transitions of the frame sizes. SAM can also provide accurate results with relatively fewer numbers of frames. For instance, SAM results with 1500 preprocessed frames have only 4.7% improvement over SAM with 250 preprocessed frames [89].

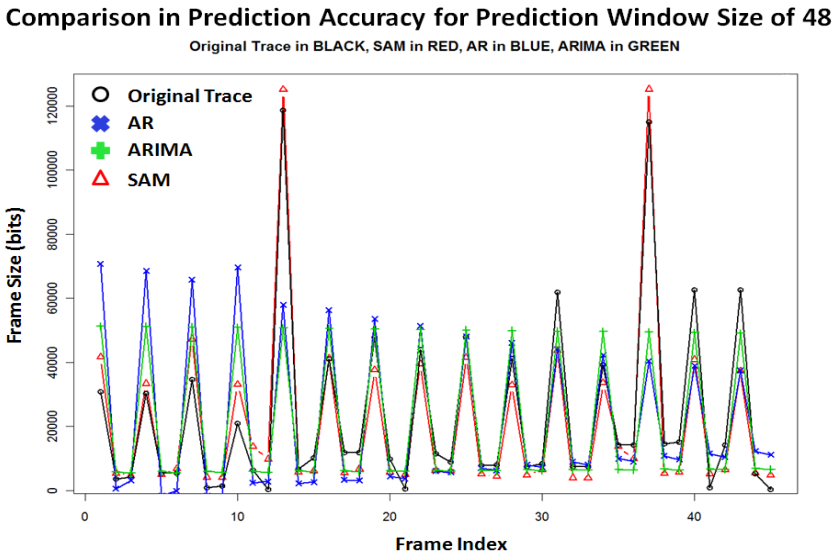


Figure 6-8 Prediction Comparison between AR, ARIMA, and SAM

We further investigated the possibility of using SAM with even fewer numbers of frames. Theoretically, SAM needs a minimum of 29 frames as suggested in [90]. Our results showed that we need at least 100 frames to achieve the desired accuracy. With SAM, using 1500 frames provided only 1% improvement over using 100 frames on average. Thus, based on our results, we recommend using SAM with 100 frames (~4 seconds) to predict the subsequent 120 frames (5 seconds). This means that a dynamic bandwidth allocation scheme needs only to negotiate the allocation once every 5 seconds.

In this chapter, we presented our work of encoding, analyzing, and modeling over 50 HD video traces that represent a wide spectrum of statistical characteristics. We collected over 50 HD video traces from YouTube website that represents a wide variety of video traces. We encoded these traces using AVC standard with the most common settings supported by experts' recommendations. These traces provide the research community with the means to test and research new methods to optimize network resources, and especially using dynamic bandwidth allocation. All the video traces and the developed tools are available to the research community through our website [87].

We performed a full statistical analysis to show the variance of the collected video traces using the most common quantitative measures. We performed a factor analysis to determine the principal components that defines a HD video trace. We concluded that both Mean and

Kurtosis values can be considered as the two main principal components. Our analysis has shown that both Kurtosis and Skewness values are important in defining a HD video trace, contrary to what has been considered before for MPEG1 encoded videos.

We performed a cluster analysis on our collection of HD videos using k-means clustering. Our results showed that we can group these movies into two main clusters. We supported our results using different graphical and non-graphical methods.

We compared three modeling methods in their ability to model our collection of HD video traces. Our results showed that SAM has a clear advantage over both AR and ARIMA models in both accuracy and simplicity as represented in its AIC values.

We have also compared these methods in their ability to forecast video traffic. Our prediction analysis was based on several factors to ensure that the chosen model is capable of providing the best results under the lowest requirements. Our results showed once again that SAM has a significant improvement over both AR and ARIMA, where it provided at least 50% better SNR^{-1} values.

Finally we illustrated the implementation of two of our developed tools. We showed the ability of the SAM-based generator of generating HD video traces that can be configured and used in different simulation scenarios. This contribution also aims to define our initial steps to achieve a better dynamic bandwidth allocation scheme designed to optimize bandwidth utilization with the presence of the high demanding HD video streams.

In the next chapter, we describe our dynamic resource allocation scheme based on our SAM model.

Chapter 7

7 SAM-based Dynamic Resource Allocation Scheme

In order to conserve the computational resources and have better control over the incoming video streams, per flow management and bandwidth allocation is done usually at the edges of the network, as shown in Figure 7-1. Such a position also allows the deployment of better admission control mechanisms. In such schemes, the emphasis is to increase the network resources utilization while maintaining the desired level of QoS.

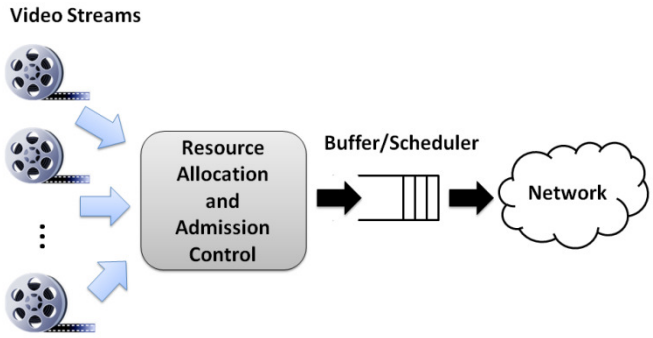


Figure 7-1 Dynamic Resource Allocation Scheme

Dynamic resource allocation (DRA) schemes are especially important for live streams, where the video stream characteristics are not known in advance. In order to provide an accurate estimation of the needed network resources for a certain flow, the chosen DRA scheme has to be able to predict the required bandwidth for future video frames. Such a prediction is preferably dependent only on the information available from the incoming video stream to meet the delay constraints and to keep the required computational power low. In addition to that, broadcasted information from the video source can either be unrepresentative of the video stream, or not available for live streams.

Dynamic resource allocation should be independent of the carrying protocol. Preferably, it should also be able to provide long term predictions. As shown in Figure 7-2, to adjust the bandwidth assignment for a certain video stream, DRA renegotiates the assigned bandwidth

for the flow. It is important to run the prediction mechanism as few times as possible to conserve the computational resources.

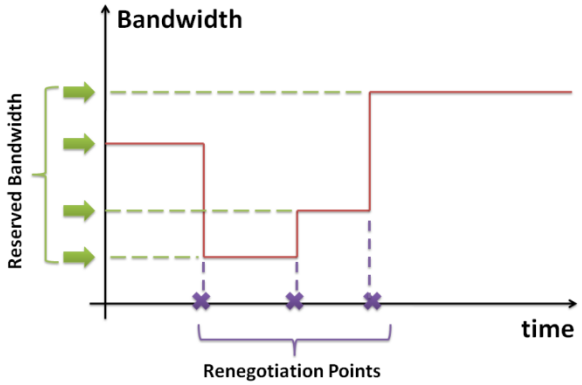


Figure 7-2 Renegotiating Bandwidth Resources Upon Traffic Changes

To summarize, the main goals for a DRA scheme for online video streams are: to predict the longest possible period with the least prediction errors, and to provide the best possible resource utilization with the lowest achievable frame delay. In order to achieve these goals, such schemes should provide the answers to the following two questions: when to renegotiate the allocated bandwidth for the flow (detect a change of traffic pattern), and how much to increase or decrease the allocated bandwidth (predict the future demand of the video stream).

Content based dynamic bandwidth allocation can be at the video frame level [91], group of pictures (GoP) level [91, 92], or scene level [93, 94]. In [94], the authors proposed a DRA scheme based on neural network (NN) prediction. The scheme contains several modules to provide an accurate prediction. These modules include: computing traffic description, extracting content features, classifying traffic patterns, and running principal component analysis for the feature selection process. The results were based on 13175-frame video encoded via MPEG-1 VBR that is divided into 177 shots. To achieve the desired results, the first 50 shots were used as training samples. The proposed complex approach reduces its applicability to support deadline requirements. In addition, the usage of NN to predict the traffic requires a significant set of data before starting the prediction process.

In [95], the authors proposed an object based video content classification scheme to map video scenes into their bandwidth resource requirements. The scheme is based on the

hierarchal object-based scene segmentation scheme, where the video scene is segmented into several smaller video objects. This mechanism divides the scene to a group of objects identified by their video object descriptor (VOD), and one scene type descriptor (STD). The number of scene types is not fixed and depends on the intended video application.

Video objects are identified by searching for spatial regions that have consistent video features like color, motion or/and texture. There are many features that can be extracted from the selected video objects. In [95], the authors identified three main features: object size, object spatial complexity, and object relative motion speed. The video scene classification scheme uses a multilayer hierarchy, where at each layer the video objects with their identified features, in addition to the identified scene type define the scene classification model. The results were obtained using a 0.5 hour MPEG-2 VBR video trace. As discussed, this approach is also considerably complex, especially for video scenes with high level of texture and motion levels.

In [92], the author used a fixed-size adaptive least mean-square (LMS) error linear predictor to determine the required bandwidth allocation based on the frame level prediction requiring a separate prediction process for each frame type, and a simpler GoP level prediction. This adaptive algorithm utilizes a fixed size adjustment to adapt to the traffic changes. The authors in [97] used the variable size LMS predictor proposed in [96] to predict the future bandwidth requirements based on the prediction of I-frames. Their results are based on four half-hour long MPEG video traces.

While these proposed approaches are substantially simpler, they are operating at the GoP level so they may provide lower prediction accuracy. These different approaches have been considered since modeling video traces at the frame level is considered a challenging task due to the different characteristics of each frame type.

The main problem with content-based predictions using a frame-based mathematical model is that they require precious computing resources, and may not be applicable to video traces encoded with different encoding standards or settings [94]. Our model, a simplified seasonal autoregressive integrated moving average(ARIMA) model, or SAM, has demonstrated its capability of modeling movies encoded with different encoding settings and standards [64,

65]. In addition, the simplification of SARIMA modeling as represented in SAM, as we show in the next section, allows it to be considered for real time implementations.

In this chapter, we present a dynamic resource allocation scheme to support real time video streaming based on our SAM model. In our design, we consider the applicability of the scheme and compare it to the variable size LMS predictor using more than 54 HD video traces. In the next section, we describe our SAM model simplification and show our approach to predict future video frames. Section 7.2 illustrates several methods to determine the trend changes in the video trace and our presented approach to achieve it. Section 7.3 demonstrates the design of our delay guaranteed SAM-based DRA scheme. Section 7.4 describes our simulation experiments comparing SAM to variable size LMS predictor.

7.1 Using SAM in Online Traffic Forecasting

In this section we describe our approach to use SAM to predict future video traffic. SAM can be represented using the simplified notation:

$$SAM = (1,0,1) \times (1,1,1)^s$$

where s here represents the seasonality of the video trace. This means that SAM requires only 4 coefficients to be estimated. These coefficients are: AR coefficient (φ), MA coefficient (θ), seasonal AR or SAR coefficient (Φs), and seasonal MA or SMA coefficient (Θs). SAM can be represented using SARIMA notation as:

$$\varphi(B)\Phi(B^s)\nabla_s x_t = \theta(B)\Theta(B^s)e_t$$

or

$$(1-\Phi_s B^s)(1-\varphi B)(1-B^s)x_t = (1-\theta B)(1-\Theta_s B^s)e_t$$

we can simplify this further:

$$(1-B-\varphi B+\varphi B^2-\Phi_s B^s+\varphi\Phi_s B^{s+1}+\Phi_s B^{s+1}-\varphi\Phi_s B^{s+2})x_t = (1-\theta B-\Theta_s B^s+\theta\Theta_s B^{s+1})e_t$$

and in difference equation form:

$$\begin{aligned} x_t &= x_{t-1}+\varphi x_{t-1}-\varphi x_{t-2}+\Phi_s x_{t-s}-\varphi\Phi_s x_{t-s-1}-\Phi_s x_{t-s-1}+\varphi\Phi_s x_{t-s-2} \\ &-\theta e_{t-1}-\Theta_s e_{t-s}+\theta\Theta_s e_{t-s-1}+e_t \end{aligned}$$

Since SAM requires only 4 parameters to be estimated, such convenient simplification allows it to be used in real time applications.

7.1.1 Model Parameters Estimation Methods

The estimations of SAM parameters can be done, as mentioned before, using ML, CSS, or CSS-ML methods. Most literature suggests ML as the preferred option to obtain the best parameters estimations [22, 23].

To support real time prediction, the estimation method should provide a good tradeoff between high accuracy and high computation speed. To determine the best suitable estimation method, we compared ML, CSS, and CSS-ML methods in modeling our collection of 54 HD video traces. The average trace length is 3212 frames, with a maximum of 9388 frames, and a minimum of 580 frames.

The results of the comparison are shown in Table 7-1 and Table 7-2. First we compared the three methods in the total time needed to model our collection of video traces. As shown in Table 7-1, CSS method has a clear advantage over both ML and CSS-ML. Using CSS it took only 0.22 (secs) on average to model a full video trace, compared to 39.54 (secs) using ML.

As mentioned before, high computation speed should not come at the expense of modeling accuracy. We used three statistical measures to compare the modeling accuracy of the three methods: Mean Absolute Error (MAE), Mean Absolute Relative Error (MARE), and Root Mean Square Error (RMSE). These methods can be computed as shown in Chapter 4 and Chapter 6.

Both Table 7-1 and Table 7-2 show that the difference between ML and CSS in terms of accuracy is less than 2.5%. We argue that such a degradation of accuracy is acceptable compared to the significant boost in computation speed.

Table 7-1 Estimation Methods Comparison Results

Comparison/Method	ML	CSS	CSS-ML
Total execution time (s)	2135.06	15.6	1202.68
Average time per video (s)	39.54	0.22	22.27
MAE (average)	12437.8	12568.1	12523.48
MARE (average)	2.395	2.447	2.417
RMSE (average)	21712.3	22162.3	22004.9

Table 7-2 Percentage of Improvement Between Estimation Methods

Comparison/Method	ML vs CSS	CSS-ML vs CSS	ML vs CSS-ML
MAE	1.04%	0.35%	0.68%
MARE	2.12%	1.27%	0.86%
RMSE	2.03%	0.71%	1.33%

Based on these results we recommend using CSS as the parameter estimation method for SAM. In this contribution, all our results are based on using CSS method.

7.1.2 Forecasting Using SAM

In Chapter 6, we showed that SAM requires only around 100 previous values to provide an accurate model. This observation is in accordance with the recommended guidelines for forecasting using ARIMA models [23]. In addition, in [94] the authors showed that most useful traffic information is presented in the short-term bandwidth statistics. This approach provides a valid method to achieve the desired forecasting without sacrificing performance since in practical applications the resulting forecasts are dependent significantly only on the recent values of the observed data series [70].

We achieve forecasting or prediction of future values of x_t directly from the previously mentioned SAM's difference equation. In this process the values of x_t and the estimation error terms are substituted as follows:

$$substituted\ x_{t+i} = \begin{cases} x_{t+i} & i \le 0 \\ \hat{x}_t(i) & i > 0 \end{cases}$$

$$substituted\ e_{t+i} = \begin{cases} e_{t+i} & i \le 0 \\ 0 & i > 0 \end{cases}$$

where $\hat{x}_t$ is the estimated frame size at time t. Such assumptions are valid since forecast values are unaffected by the small changes that are introduced by the estimation errors [70]. In this section, we explained our rationale to use SAM and our approach to model and forecast video frames.

7.2 Determining Trend Changes

It is important to determine the boundaries where the trend of the video frames changes in order to determine the best time to renegotiate the currently allocated resources. Such a change is usually associated with a change of the current video scene. The process of

determining the renegotiation-points belongs to one of the three categories: deterministic, traffic based and content based.

In deterministic approach, the renegotiation points are set every *n* frames, where these renegotiation points are empirically decided. As an example, we investigated the algorithm proposed in [100] for its simplicity. In this approach, a scene change is identified if the size of successive frames in the observed video stream exceeds a certain threshold in a sustained manner. Let's denote number of bits in n-th frame as x_n, and the scene threshold as J_{min}. Then the random sequence of bit-rate *jumps*, J_n, which exceed the J_{min} threshold, can be written as:

$$J_n = \begin{cases} 1 & |x_n - x_{n-1}| > J_{\min} \\ 0 & otherwise \end{cases}$$

To correctly identify the boundaries of the scenes, we need to make sure to have a sustained bit rate change. Thus, it is necessary to introduce minimum scene length parameter denoted by L_{min}. Following this logic, the function sequence indicator of the n-th frame, S_n can then be given by:

$$S_n = \begin{cases} 1 & if\ J_n = 1,\ J_{n-1} \neq 1,\ J_{n-2} \neq 1, \cdots,\ J_{n-L_{\min}} \neq 1 \\ 0 & otherwise \end{cases}$$

If the n-th frame has $S_n = 1$, then that frame indicates a scene change. This approach requires a mechanism to specify both J_{min} and L_{min} values in order to correctly determine the boundaries of a scene. For instance, in [20] it was determined that for *Star Wars* JPEG encoded video trace $J_{min} = 5000$, and $L_{min} = 2$. The obvious problem with this approach is that we need a separate empirical estimation for each incoming video flow. Such information is not only resource demanding, but also unavailable for live streams.

Traffic-based determination approach instead observes the changes of the bandwidth of the video flow. When the bandwidth required for that flow exceeds, or drops below a certain threshold relative to the previously negotiated bandwidth request, a new renegotiation process is initiated. The problem with this approach is that video traffic has significant variations between the video frame sizes in a short-term interval, due to the way video frames are encoded. For instance, I-frames are usually 3 times the size of B-frames [65]. In

addition to that, defining the thresholds is an empirical process and requires human intervention. Relying on such an approach may result in wasting bandwidth resources due to both incorrect reaction to the changes of bandwidth requirements, and the frequent renegotiation points.

The third approach is to determine the renegotiation points by analyzing the content of the video. This approach is the most appropriate for a fast-changing traffic like video streams. Our presented model is a seasonal ARIMA model. SARIMA models can be decomposed into three components: trend, seasonal and irregularities [70]. The trend component indicates the underlying direction and rate of change in SARIMA forecast. A simple way to detect a trend is to compute the average of frames over a certain time period and try to investigate the existence of a trend in the time series.

To identify a trend change, we propose to simply compute the aggregated video frames' sizes over the GoP length of the video trace. As shown in Figure 7-3, the aggregation gives a clear indication of the trend of the video stream. It is important to mention that the aggregated values (shown in red, with a thick line) are not represented by their actual sizes and are simply laid over the actual video trace (shown in grey, with a thin line) to demonstrate the trend of the video trace.

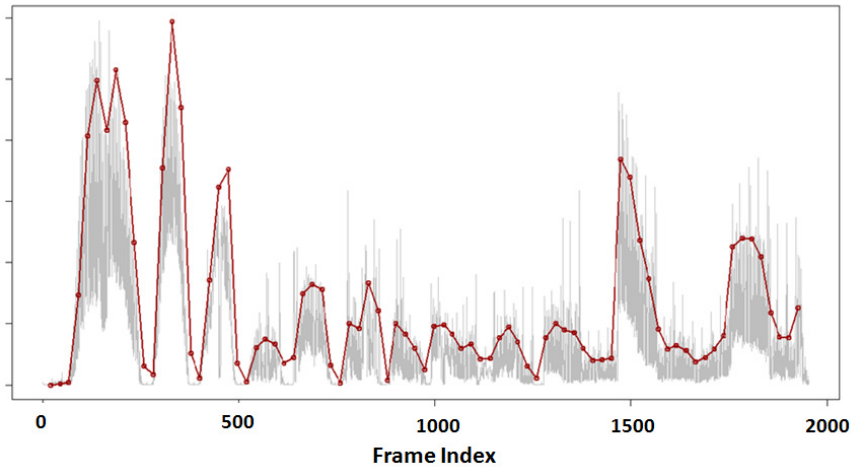


Figure 7-3 Detecting Traffic Trend Using GoP Aggregation

To determine the level of aggregation needed to convey the underlying structure of the video trace and to simplify the identification process, we compared the aggregation of frame sizes over one GoP, two GoPs, and over four GoPs. As can be noticed from Figure 7-4, by

increasing the aggregation size we might miss a trend change. Overlooking such changes may result in severe prediction errors. It is important to note that the GoP aggregation lines are not of their actual sizes, and they are laid over each other to simplify the graph and show the frame size trends.

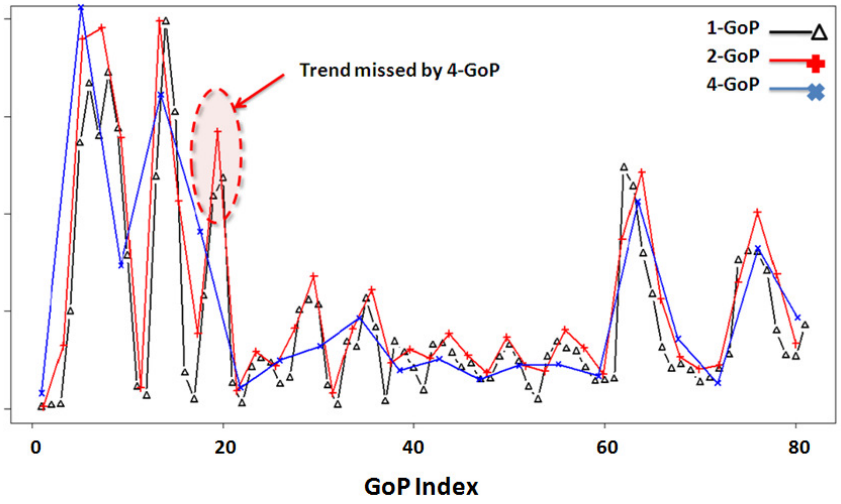


Figure 7-4 Comparing Different GoP Aggregation Levels

Thus we suggest using the aggregation of fewer than 4-GoPs as a tradeoff between accuracy and computation speed. Our experimental results in Section 4 use 1, 2, and 3 GoP aggregations.

7.3 SAM-based DRA Scheme

In this section, we discuss our approach to determine the required adjustment to the allocated bandwidth upon detecting a trend change. As shown in Figure 7-5, SAM predictor analyzes the observed traffic information and then it accordingly predicts the future incoming traffic. In case of a trend change, the dynamic bandwidth allocation scheme adjusts the previously allocated bandwidth based on the traffic predictor.

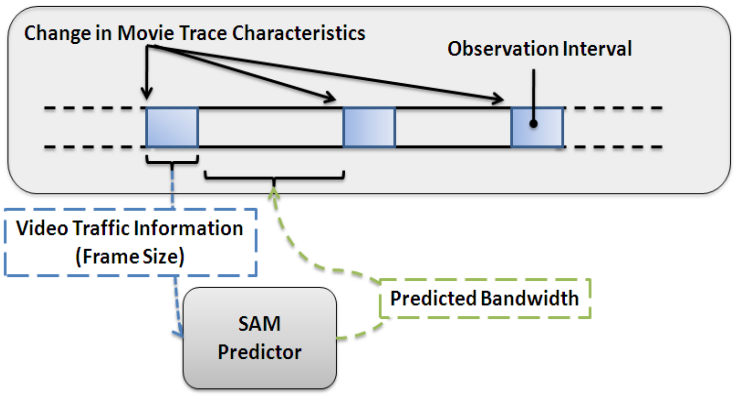


Figure 7-5 SAM-based Traffic Prediction

In case of a prediction error, if the traffic predictor allocates a higher bandwidth for the incoming flow than it requires, there will be a waste of network resources (link utilization). On the other hand, if the prediction process results in a lower allocated bandwidth; the difference should be buffered and then sent later within the acceptable delay limits for live video streams.

Figure 7-6, shows the presented SAM based DRA model. This represented model demonstrates a possible scheme to provide dynamic resource allocation based on our SAM model. The incoming video flow is processed through the SAM-based stream resource controller (SRC), where the prediction process is performed. The prediction difference due to the prediction errors is buffered. At the renegotiation points, a simple request/response mechanism is used to communicate with the network resource manager (NRM). Depending on the available network resources, NRM determines whether the requested increase, or a new flow request can be supported. In case the incoming flow cannot be supported by the network, SRC may send a feedback to the live-video source encoder to use a lower bit rate, as described in [101]. To simplify our analysis and comparisons, we assume that all bandwidth requests are granted.

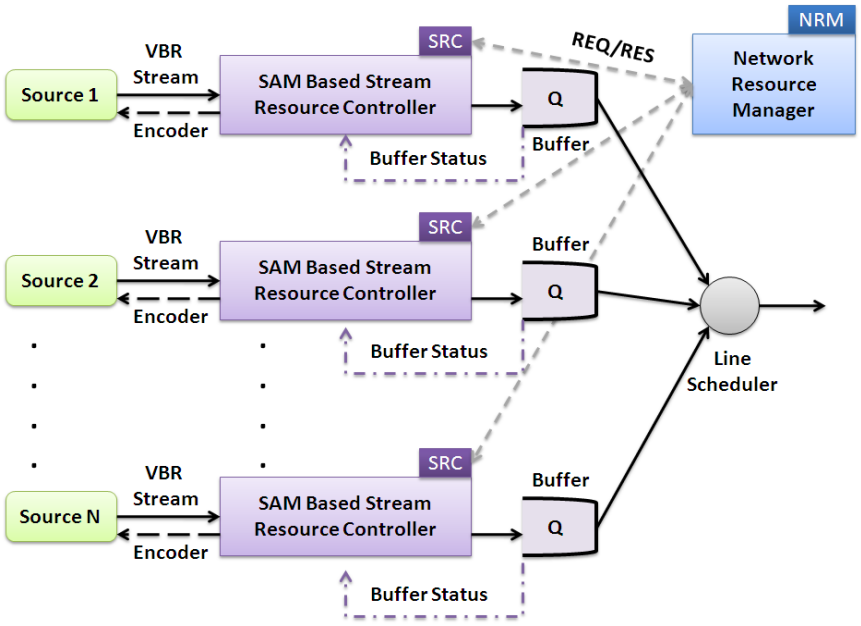


Figure 7-6 SAM-based DRA Model

As can be noticed from Figure 7-6, by using SAM to predict and reserve the bandwidth dynamically, the allocation problem has changed from supporting highly variant incoming video flows, to servicing the predicted allocations and buffering the possible prediction errors. Thus, the better the predictor, the better the system performance in servicing the existing flows and admitting new video flows.

As mentioned before, the dynamic allocation scheme should provide the best possible tradeoffs between network resource utilization and meeting the flow delay requirements. In other words, researchers should know that reducing the number of renegotiations might be at the expense of increasing the buffer occupancy and reducing the network utilization level.

7.4 Experimental Results

Online video traffic has high requirements for acceptable frame delay. For such traffic, it is important to guarantee these traffic parameters. Delayed video frames are discarded at the end nodes if they do not arrive before their specified display time. Cisco, for example, sets the requirement for online and interactive video delivery (e.g. video conferencing) to have the maximum one-way delay of 150 ms [102]. IPTV QoS requirements for MPEG-4 AVC encoded HDTV service, as described in DSL Forum technical report (TR-126) for triple play service [135], are set to a maximum of 200 ms for one-way delay.

SAM-based SRC aims to manage the dynamic bandwidth allocation while meeting the one way delay requirement. We use the proposed QoS guaranteed dynamic bandwidth allocation (QDBA) algorithm [97]. QDBA operates at the GoP level. Operating at the GoP level provides two advantages to the forecasting process: it simplifies the calculation of the incoming flows, and it also acts as a smoother for the variable video stream, allowing easier prediction [105]. Table 7-3 defines the QDBA parameters.

Table 7-3 QDBA Parameters

Parameter	Description
M	number of frames per GoP
F	number of frames to be transmitted per second
ρ	required bandwidth utilization
$\mu_{(i)}$	allocated bandwidth at time slot i
$Q_{(i)}$	queue size at the end of time slot i
$G_{0(i)}$	actual size of the GoP at a time slot i
$G_{p(i)}$	predicted size of the GoP at time slot i
$R_{p(i)}$	predicted bandwidth at time slot i
$R_{t(i)}$	required bandwidth at time slot i for delay constraints
T	maximum allowed delay
R	current transmission rate

and the QDBA algorithm at time slot i can be expressed as:

$$Q(i-1) = \max\{G_0(i-1) + Q(i-2) - \mu(i-1) \times \frac{M}{F}, 0\}$$

$$R_p(i) = G_p(i) \times \frac{F}{M}$$

$$R_t(i) = Q(i-1)/T$$

$$QDBA\ Bandwidth\ Allocation: \begin{cases} R - R_p(i) \geq (1-\rho) \times R & \rightarrow R = \max\{R_p(i), R_t(i)\}, \mu(i) = R \\ R < R_t(i) & \rightarrow \mu(i) = R_t(i), R = \mu(i) \\ otherwise & \rightarrow no\ change \end{cases}$$

Figure 7-7 QDBA Algorithm

QDBA algorithm compares the required bandwidth allocation at each time slot, taking into consideration the buffer status, against both the predicted and currently allocated bandwidth. If the current rate is higher than the required rate, considering the required link utilization, then it renegotiates a lower rate that is the maximum of the predicted and required rates. If the allocated rate is lower than the required rate, the renegotiated rate is set to the required rate. Otherwise, the allocated rate remains the same.

We compare our SAM-based traffic predictor to the traffic predictor based on the non-linear variable step-size adaptive (VSA) algorithm proposed in [96]. VSA is an improvement over the fixed step-size adaptive (FSA) algorithm [92]. VSA bases its prediction on the GoP size instead of the frame size. This approach was favored since modeling and predicting the GoP size is considered a simpler problem. By operating on the frame size level, SAM aims to provide better prediction results.

Let us consider G_i the i-th GoP size, then the p-th order of VSA predictor is:

$$\hat{G}_{i+1} = \sum_{j=0}^{p-1} w_j G_{i-1} = W^T G_i$$

where $\hat{G}_{i+1}$ is the forecasted size of the next GoP, p is the order of the predictor, w_j are the prediction filter coefficients for j=0,1,…,p-1. The predictor's order is chosen empirically to achieve the best results. The prediction error can be expressed as:

$$e_i = G_i - \hat{G}_i$$

The filter coefficients are updated after each new prediction using the following equation:

$$W_{i+1} = W_i + \frac{\mu_i e_i G_i}{\|G_i\|^2}$$

where $\|G_i\|^2 = G_i^T \times G_i$. Instead of using a fixed value for the updating coefficient μ, its value is updated to allow variable step-size adjustments. Increasing the value of μ results in a fast convergence but at the expense of higher prediction errors. Smaller μ value results in smaller prediction errors with slower convergence rate. For highly variable input stream like HD video traces, it is important to choose the correct value for μ to allow fast adaption to the stream variation with the lowest possible prediction errors.

The parameter μ_i is updated using:

$$\mu'_{i+1} = \alpha\mu_k + \gamma(q_1 e_i^2 + q_2 e_{i-1}^2)$$

$$\mu_{i+1} = \begin{cases} \mu_{\max} & if\ \mu'_{i+1} > \mu_{\max} \\ \mu_{\min} & if\ \mu'_{i+1} < \mu_{\min} \\ \mu'_{i+1} & otherwise \end{cases}$$

here α is the previous μ weight, γ is the collective error terms weight, q_1 and q_2 are the current and the previous prediction errors weights respectively. $\mu_{\max}$ and $\mu_{\min}$ are used to bound the step-size adjustment. $\mu_{\max}$ is chosen to ensure that the mean square error (MSE) remains bounded, while $\mu_{\min}$ is the same value chosen for FSA algorithm. As shown here, VSA requires 7 coefficients to be determined empirically before being deployed, which is considered a down-side to this approach. SAM, on the other hand, does not require any prior information or prior empirical calculations.

Following the suggestions in [104], for VSA we chose α = 0.98, γ =0.015, q_1 = 0.7, and q_2 = 0.3. Following the suggestions in [97], we set the initial value of the updating coefficient μ_0 = $\mu_{\min}$ = 0.009, $\mu_{\max}$ = 0.3, and the prediction order p = 12. In our analysis, we found out that to achieve the best prediction results we need to set $\mu_{\max}$ = 0.03, and p = 5.

As we stated before, one of the main targets in bandwidth allocation schemes design is to minimize the number of the renegotiations points. Therefore, we modified VSA to allow the prediction for more than one GoP. The modification simply allows VSA to operate on aggregation of multiple GoP sizes, instead of one GoP size.

The performance of the dynamic bandwidth predictors using QDBA scheme is measured using three parameters: renegotiation frequency, the total allocated bandwidth, and the total buffer usage or occupancy. A better predictor will result in fewer prediction errors to be buffered (smaller queue occupancy), better future prediction (fewer renegotiation points), and better utilization of the network resources under the defined delay requirements (lower bandwidth allocation rate).

7.4.1 Comparison Results Using 54 AVC Video Traces

Table 7-4 shows the average performance comparison results using our collection of 54 HD video traces for the maximum allowed delay of 100 ms, and required bandwidth utilization of $\rho = 0.9$. SAM-n indicates the usage of SAM with n-aggregated GoPs. For example, SAM-2 means two GoP sizes are aggregated.

Table 7-4 Percentage of Improvement for Using SAM over VSA

Comparison/Method	**SAM-1 Vs VSA-1**	**SAM-2 Vs VSA-2**	**SAM-3 Vs VSA-3**
Allocated Bandwidth	19.8%	8.6%	7.7%
Negotiation Frequency	0.5%	3.5%	5.77%
Queue Occupancy	25.2%	13.8%	13%

We notice that SAM outperforms VSA in all the performance comparisons due to its better ability to predict future traffic. Even with the low number of frames tested with average of ~3000 frames, SAM provides 7.7% (SAM-3) to 19.8% (SAM-1) bandwidth utilization improvement, and 13% (SAM-1) to 25.2% (SAM-1) queue occupancy improvement. By increasing the number of aggregated GoPs the difference between the two approaches becomes lower because it represents a smoother version of the video trace. The same observation is noticed in the queue occupancy comparison. It is important to mention that increasing data aggregation results in higher error rates and thus higher queue occupancy and higher bandwidth allocation rates. Table 7-5 shows the average percentage of increment for both queue occupancy and allocated bandwidth.

Table 7-5 Percentage of Increment for Queue Occupancy and Allocated Bandwidth

Comparison	**GoP-2 Vs GoP-1**	**GoP-3 Vs GoP-1**
Allocated Bandwidth	12%	14.8%
Queue Occupancy	10.6%	12.8%

The improvements of negotiation frequency increases in proportion to the number of aggregated GoPs since the total number of renegotiations are fewer with higher levels of aggregation. For instance, 0.5% reflects the improvement from 105 (VSA) to 104 (SAM), where 3.5% reflects the improvement from 55 (VSA) to 53 (SAM). Renegotiation frequency is almost cut in half when using GoP-2 aggregation. Thus, as a tradeoff between high accuracy and lower renegotiation frequency, GoP-2 aggregation can be used. Figure 7-8 shows the affect on the average queue size with varying the required bandwidth utilization

factor ρ, and the required maximum allowed delay (denoted by T). This comparison was done using *Star Wars IV* trace with GoP-1 aggregation. As it is shown, SAM has a clear advantage over VSA, especially for stricter delay constraints.

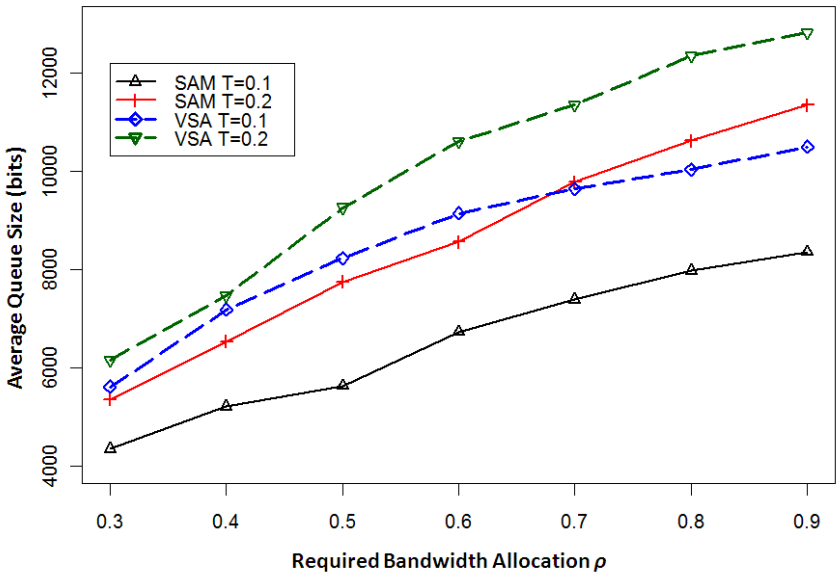
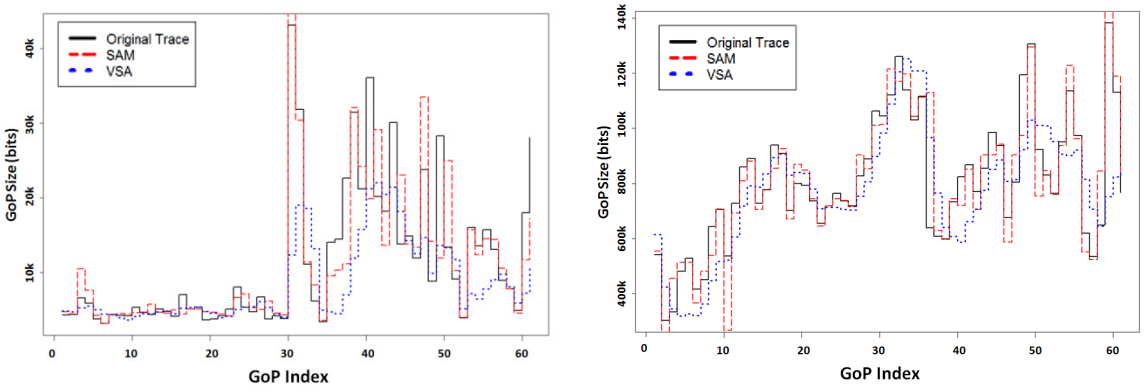


Figure 7-8 Average Queue Size against Different Utilization and Delay Requirements

Figure 7-9 illustrates a close-up comparison between the VSA-1 and SAM-1 in their ability to predict future traffic using two of the simulated video trace samples. It is clear that SAM predicts the traffic better especially in the case of sudden events since it operates on frames level.



(a) Star Size video prediction comparison (b) Travel_Cleveland video prediction comparison

Figure 7-9 SAM versus VSA Prediction Rate Comparison

7.4.2 Comparison Results Using Long AVC Video Traces

To support our assumption that the difference between SAM and VSA will be even more substantial using longer video traces, we compared the two methods using three long traces obtained from [107]. The three selected traces are chosen to represent various video characteristics: Silence of the Lambs (~30 min, 52384 frames), Tokyo Olympics (~74 min, 131520 frames), and Star Wars IV(~30 min, 52384 frames), representing action, thriller, and sports video genres, respectively. All these movies are encoded using AVC main profile, with resolution of 352×288 (CIF size), a frame rate of 30 fps, a GoP size of 16, 7 B-frames per GoP, and a quantization level of 28 for all frames.

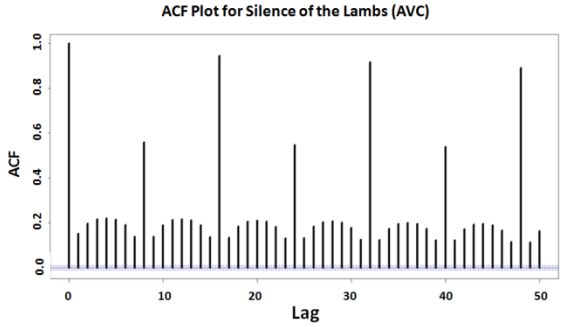
Table 7-6 shows the performance analysis results for the three movies. Again, SAM has a clear advantage over VSA in all the tested GoP aggregations. We can also see that the performance gain has increased using longer traces. Therefore, using SAM especially for live and continuous streams applications like IPTV will result in a better utilization of network resources.

Table 7-6 Percentage of Improvement for Using SAM over VSA for Long Video Traces

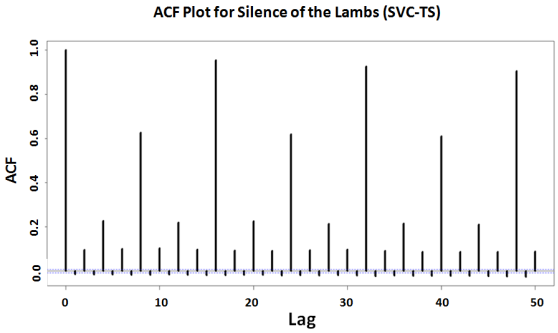
Video Trace	Comparison/Method	SAM-1 Vs VSA-1	SAM-2 Vs VSA-2	SAM-3 Vs VSA-3
Star Wars IV	*Allocated Bandwidth*	21.7%	25.3%	17.7%
	Negotiation Frequency	1.68%	5.8%	7.95%
	Queue Occupancy	22.8%	26.6%	19.3%
Silence of the Lambs	*Allocated Bandwidth*	22.4%	15.5%	22.6%
	Negotiation Frequency	3.9%	5.65%	8.04%
	Queue Occupancy	23%	16%	23.8%
Tokyo Olympics	*Allocated Bandwidth*	25%	24.7%	25.8%
	Negotiation Frequency	3.1%	8.7%	9.4%
	Queue Occupancy	27.9%	27.2%	28.2%

7.4.3 Comparison Results Using Long SVC-TS/SVC-SS Video Traces

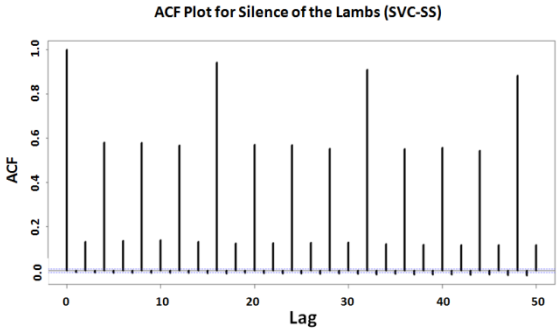
Additionally, we test the capability of our SAM-DRA to predict both scalable video codec with temporal scalability (SVC-TS), and SVC with spatial scalability (SVC-SS) using different encoding settings. SVC-TS splits the video bit stream into a base layer and a hierarchy of enhancement layers. A higher frame rate can be achieved when processing more of the temporal enhancement layers. SVC-SS also uses a hierarchy of enhancement layers, where each layer instead corresponds to a spatial resolution. Due to using different techniques of encoding, different correlations among the video frames resulted, as shown in Figure 7-10. We used the same three movies chosen in the previous step. For SVC-TS, we used GoP size of 16, 7 B frames, and quantization parameter of 28, with three temporal enhancement layers. For SVC-SS, we used GoP size of 16, 3 B frames, and quantization parameter of 28, with one base layer to support QCIF size (176×144), and one enhancement layer to support CIF size (352×288). The SVC-SS videos also incorporate temporal scalability with one base layer and two enhancement layers.



(a) Autocorrelation for Silence of the lambs using AVC



(b) Autocorrelation for Silence of the lambs using SVC-TS



(c) Autocorrelation for Silence of the lambs using SVC-SS

Figure 7-10 Frames Autocorrelation when Using AVC, SVC-TS and SVC-SS Encodings

Table 7-7 shows the improvements of using SAM to provide dynamics allocation over VSA for SVC-TS encoded videos. Table 7-8, shows the improvements for SVC-SS encoded videos. As the results show, SAM provide up to 26.9% improvement (SAM-1) in bandwidth utilization, up to 10.2% in negotiation frequency (SAM-3), and up to 29.9% in queue occupancy for SVC-TS videos. For SVC-SS videos, the achieved improvement is up to 32.4% (SAM-1) in bandwidth utilization, up to 12% in negotiation frequency (SAM-3), and up to 36.4% (SAM-1) in queue occupancy.

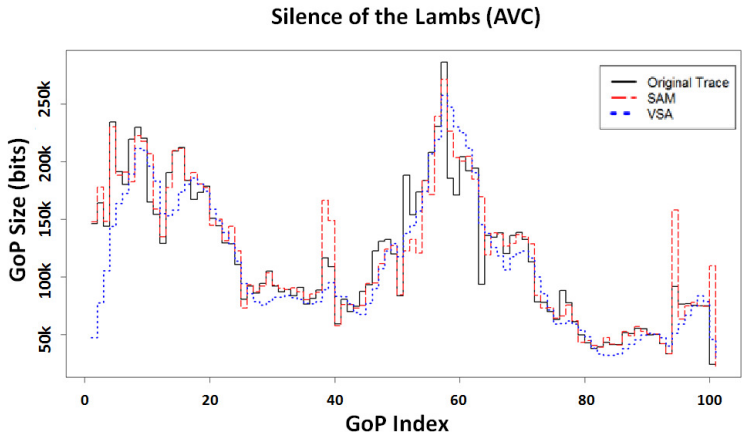
Table 7-7 Percentage of Improvement for Using SAM over VSA for SVC-TS Videos

Video Trace	Comparison/Method	SAM-1 Vs VSA-1	SAM-2 Vs VSA-2	SAM-3 Vs VSA-3
Star Wars IV	*Allocated Bandwidth*	23.7%	24.5%	19.1%
	Negotiation Frequency	4.2%	7.4%	6.3%
	Queue Occupancy	25.9%	26.3%	20.5%
Silence of the Lambs	*Allocated Bandwidth*	26.9%	26.8%	23.4%
	Negotiation Frequency	4.0%	9.67%	10.2%
	Queue Occupancy	29.9%	29.3%	25.7%
Tokyo Olympics	*Allocated Bandwidth*	25.6%	24.2%	18.1%
	Negotiation Frequency	2.2%	7.5%	8.5%
	Queue Occupancy	28.8%	26.8%	19.7%

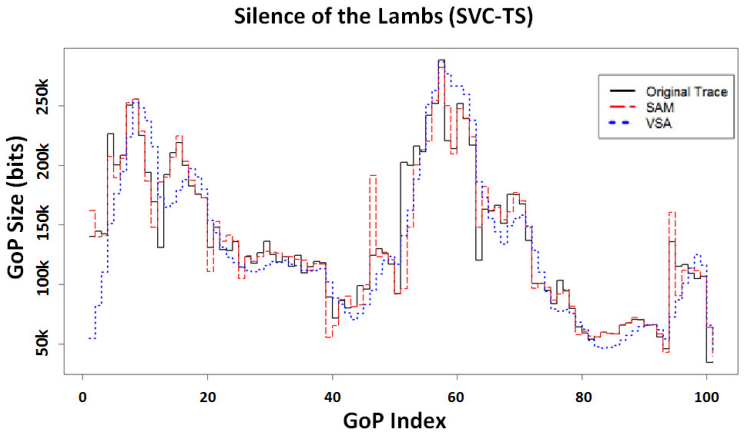
Table 7-8 Percentage of Improvement for Using SAM over VSA for SVC-SS Videos

Video Trace	Comparison/Method	SAM-1 Vs VSA-1	SAM-2 Vs VSA-2	SAM-3 Vs VSA-3
Star Wars IV	*Allocated Bandwidth*	23.6%	24.0%	19.1%
	Negotiation Frequency	5.09%	5.44%	7.86%
	Queue Occupancy	26.1%	26.0%	20.7%
Silence of the Lambs	*Allocated Bandwidth*	23.9%	24.9%	19.4%
	Negotiation Frequency	3.0%	7.5%	10.1%
	Queue Occupancy	26.5%	27.2%	21.0%
Tokyo Olympics	*Allocated Bandwidth*	32.4%	25.9%	26.1%
	Negotiation Frequency	6.3%	11.9%	12.0%
	Queue Occupancy	36.4%	28.7%	29.2%

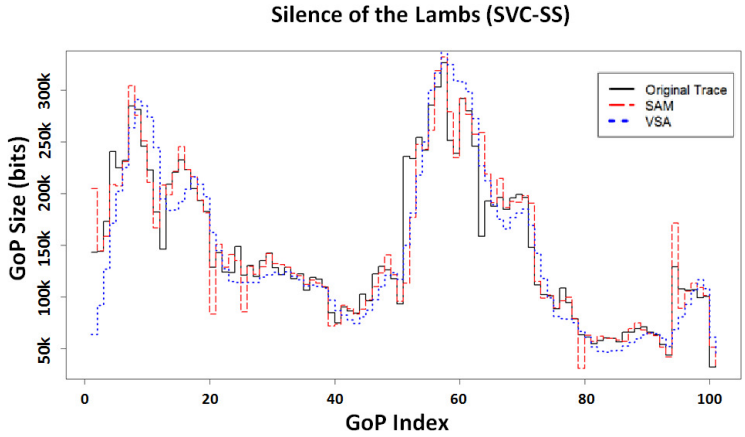
It is obvious from the results that using SAM improves the utilization of the resources. This conclusion supports the suggestion that improving the modeling accuracy, through providing a better model, can improve the utilization of the network resources. This indicates a strong relationship between the accuracy of the model and the expected performance enhancement. Figure 7-11 shows how SAM and VSA abilities to predict the video traces for the same video frames encoded with AVC, SVC-TS, and SVC-SS.



(a) Silence of the lambs using AVC



(b) Silence of the lambs using SVC-TS

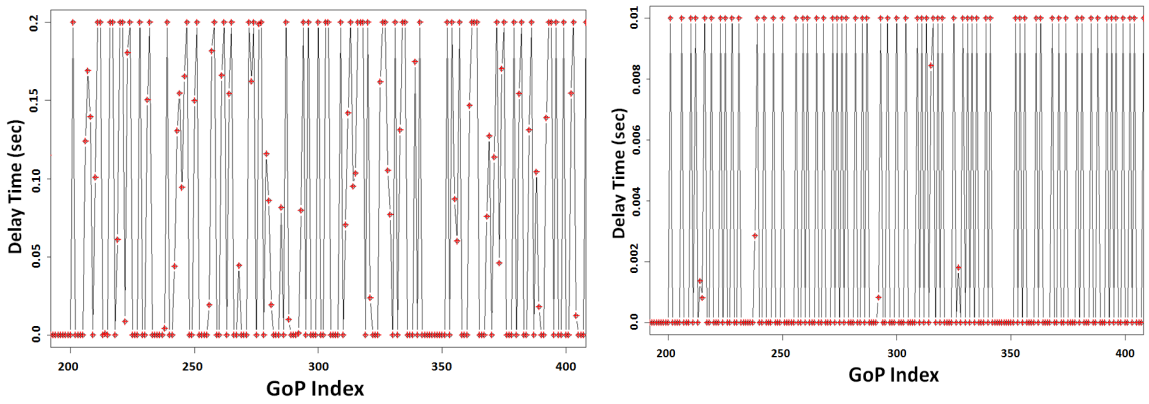


(c) Silence of the lambs using SVC-SS

Figure 7-11 SAM versus VSA Prediction Rate Comparison for the Silence of the Lambs Video Trace

As we stated before, our design takes into consideration the maximum allowed bandwidth delay for the incoming video flows. Using QDBA algorithm enforces the acceptable deadline delay and provides a good supports for the QoS requirements. Figure 7-12 (a) shows how the deadline requirement is met when the maximum delay requirement is set to T = 200 ms,

and the required bandwidth utilization is set to $\rho = 0.9$ for *Silence of the Lambs* video trace. Figure 7-12 (b) shows the effect on actual frame delay when a stringent deadline requirement of 10 ms is applied.



(a) Actual delay when T= 200ms for Silence of the lambs

(b) Actual delay when T= 10ms for Silence of the lambs

Figure 7-12 Meeting Delay Requirements in SAM-based DRA

As can be noted, the SAM-based DRA scheme meets the deadline requirements while maximizing the utilization of the available network resources by providing better prediction performance.

In this chapter, we presented a dynamic resource allocation scheme based on our SAM video model to provide a better support for online video streams. We illustrated the mechanism of using SAM model to forecast the incoming video frames depending only on the short-term history of the previously observed frames.

We compared the SAM modeling accuracy using three parameter-estimation algorithms (CSS, ML, and CSS-ML) to achieve higher computational performance. Using CSS algorithm provides a significant boost in computation speed, 0.22 seconds per video versus 39.54 seconds using ML on average, with less than 2.5% loss of accuracy in our comparisons using MAE, MARE, and RMSE.

We showed the impact of aggregating frame sizes on their corresponding GoP size, and the results of aggregating frame size over different multiples of GoP sizes. Our results showed that up to 3-GoP aggregation is acceptable without missing a significant number of video frames trend changes.

Based on our presented DRA scheme, that utilizes QDBA algorithm to achieve the best performance while meeting the delay deadlines, we compared SAM and VSA using our collection of 54 HD video traces. Based on the three performance measures, allocated bandwidth, renegotiation frequency, and queue occupancy, we showed that SAM outperforms VSA in all of them, providing up to 7.7% (SAM-3) to 19.8% (SAM-1) improvement in bandwidth utilization, and up to 13% (SAM-1) to 25.2% (SAM-1) improvement in queue occupancy on average.

We extended our analysis results by comparing VSA and SAM using three long video traces representing different video genres. The results confirmed our assumption that SAM has a clear edge over VSA especially for long video traces.

We tested the capability of SAM-DRA to support video encoded with scalable video codec extensions for both temporal and spatial scalability. The achieved results show significant improvements in all the performance measures when using SAM. These results prove the capability of SAM in handling video traces encoded with different encoding settings and standards. Thus, we believe that SAM will provide significant network resource utilization improvements for continuous video stream applications like IPTV.

The dynamic bandwidth allocation performance improvement using our SAM-based DRA scheme and its real time applicability makes it a strong candidate for real time deployment to support live video streams and improve network resources utilization.

In the next chapter, we summarize our contributions presented in this dissertation and provide an insight to our future work.

Chapter 8

8 Summary

Video streaming traffic has been surging in the last few years. The introduction of web-based video applications and the rising of video on demand services have been driving network researchers with a high motivation to seek better solutions to accommodate the growth of user demands and their expectations. The spread of broadband wireless networks, as represented in WiMAX and LTE technologies, have tremendous impact on the future of the video streaming over the wireless medium.

In this dissertation, we have described our methodology to research the characteristics of video traffic over both wireless and wired networks. We have also discussed the steps to model a variety of video traces encoded with the most common encoding standards. We targeted three of the latest and most used standards in video encoding: MPEG4-Part2, MPEG4-Part10/Advanced Video Codec (AVC) also known as H.264, and AVC's extension to support scalability, viz., scalable video codec (SVC).

One of the main goals of these analyses is to research the possibility of achieving a general mathematical approach that is capable of representing different movie traces encoded with the most used video standards. As a result, we presented our Simplified Seasonal ARIMA Model (SAM). Using various encoding settings, we showed that SAM is capable of capturing the statistical characteristics of mobile video traces encoded with the most common video standards. The simplicity of our model and its general approach are the basis of our model-based trace generator. Our trace generator is capable of producing video traffic that resembles different video traces with different encoding settings. SAM improves the ability to predict successive video traffic patterns, and thus provides a better support for both admission control and resource allocation schemes. As we have shown in our presented results, there is a direct correlation between the accuracy of the video model and the achieved resources utilization level achieved with dynamic bandwidth allocation.

In addition, we tried to shed light on the inter-correlation between video frames and their unique statistical characteristics. Using principal component analysis (PCA) and exploratory

factor analysis (EFA) techniques, we performed multivariate factor analyses on our collection of over 50 HD video traces. In addition, we aimed to better identify video traces by grouping them, depending on their statistical characteristics, into clusters using k-means clustering.

We compared three modeling methods (SAM, AR, and ARIMA) in their ability to model our collection of HD video traces. Our results showed that SAM has a clear advantage over both AR and ARIMA models in accuracy and in simplicity as represented in their AIC values.

We have also compared the ability of these methods to forecast video traffic. Our prediction analysis was based on several factors to ensure that the chosen model can provide the best results under the lowest requirements. Our results showed that SAM provides a significant improvement over both AR and ARIMA by achieving around 50% improvement in SNR^{-1} values.

We created a library of HD video traces available to fellow researchers. This video traces collection can be used not only to verify our results, but also to provide the means for the research community to develop and evaluate their contributions. In fact, all our contributions are available to the research community through our website, including our results and the developed codes and tools.

In addition, we proposed a dynamic resource allocation scheme based on our SAM model to provide a better support for online video streams. We illustrated the mechanism of using the SAM model to forecast incoming video frames depending only on the short-term history of the previously observed frames.

Using our collection of 54 HD video traces we compared SAM and VSA prediction schemes based on our proposed DRA scheme, which utilizes QDBA algorithm to achieve the best performance while meeting the delay deadlines,

Our comparisons were based on the three performance measures: allocated bandwidth, renegotiation frequency, and queue occupancy. Our results showed that SAM outperforms VSA in all of these measures. SAM provides up to 19.8% improvement in bandwidth utilization, and up to 25.2% improvement in queue occupancy on average. Additionally, we achieved a similar improvement using videos encoded with scalable video codec (SVC)

extension to support both temporal and spatial scalabilities. For temporal scalability, we achieved up to 26.9% improvement in bandwidth utilization. When using both spatial and temporal scalabilities, we achieved up to 32.4% improvement in bandwidth utilization. The results confirmed our assumption that SAM has a clear edge over VSA, especially for long video traces.

The dynamic bandwidth allocation performance improvement using our proposed SAM-based DRA scheme and its real time applicability makes it a strong candidate for real-time deployment to support live video streams and improve network resources utilization.

We aim through our future work to test SAM model using different encoding standards, including open video standards. Our analyses are to include more sophisticated systems to achieve a better understanding of the network variables that affect the final outcome of the network resource utilization.

References

[1] YouTube, YouTube web streaming website, (online) URL=[http://www.youtube.com] (March 2010).

[2] Vimeo, Vimeo website video sharing for you, (online) URL=[http://www.vimeo.com] (March 2010).

[3] C. Albrecht, "Survey: Online Video Up to 27% of Internet Traffic", (online) URL=[http://tinyurl.com/yzpzoew] (March 2010).

[4] Comscore Press Release in Aug.2009, (online), URL=[http://tinyurl.com/l4o3rs] (March 2010).

[5] Comscore Press Release in Nov.2009, (online), URL=[http://news.websitegear.com/view/149267] (March2010).

[6] Hulu Website, free online video service that offers hit TV shows, (online), URL=[http://www.hulu.com] (March 2010).

[7] Netflix website, DVD Rental and High definition video streaming service, (online) URL=[http://www.netflix.com] (March 2010).

[8] R. Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling," Wiley- Interscience, New York, NY, April 1991, ISBN:0471503361.

[9] C. So-In, R. Jain, and A. Al-Tamimi, "Scheduling in IEEE 802.16e Mobile WiMAX Networks: Key Issues and a Survey," IEEE Journal on Selected Areas in Communications (JSAC), Vol. 27, No. 2, Feb 2009, URL=[http://www.cse.wustl.edu/~jain/papers/sched.htm].

[10] R. Jain, C. So-in, and A. Al-Tamimi, "System Level Modeling of IEEE 802.16e Mobile WiMAX Networks: Key Issues," IEEE Wireless Communications, Vol. 15, No. 5, October 2008, URL=[http://www.cse.wustl.edu/~jain/papers/slm.htm].

[11] C. So-In, R. Jain, and A. Al-Tamimi, "Scheduling in WiMAX: Baseline Multi-class Simulations," WiMAX Forum Application Working Group meeting, Washington DC, November 19-20, 2007.

[12] WiMAX Forum, "WiMAX System Evaluation Methodology V2.1," Jul. 2008, 230 pp. URL=[http://www.wimaxforum.org/technology/documents].

[13] Media Wiki-Wikipedia, "MPEG Standards," URL[http://nostalgia.wikipedia.org/wiki/MPEG].

[14] N. Ansari et al. "On Modeling MPEG Video Traffics," IEEE Transactions on Broadcasting, pages 337-347. 2002.

[15] M. Krunz, and H. Hughes, "A Traffic Model for MPEG-Coded VBR Streams," ACM SIGMETRICS Performance Evaluation Review archive Volume 23, Issue 1, pages: 47 - 55. 1995.

[16] P. Seeling, F. Fitzek, and M. Reisslein, "Video Traces for Network Performance Evaluation: A Comprehensive Overview and Guide on Video Traces and Their Utilization in Networking Research," Springer; 1 edition (December 22, 2006), 274 pages.

[17] P. Seeling, M. Reisslein, and B. Kulapala, "Network Performance Evaluation Using Frame Size And Quality Traces of Single and Two Layer Video: A Tutorial," IEEE Communications Surveys and Tutorials, Third Quarter 2004, Vol. 6, No. 2, pp58-78.

[18] G. Auwera, T. Prasanth, and M. Reisslein, "Traffic Characteristics of H.264/AVC Variable Bit Rate Video," IEEE Communications Magazine, Volume 46 Issue 11, pp 164-174.

[19] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockammer and T. Wedi, "Video Coding with H.264/AVC: Tools, Performance, and Complexity", IEEE Circuits and Systems, Vol. 4, No. 1, 2004.

[20] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[21] A. Kouadio, M. Clare, L. Noblet and V. Bottreau, "SVC – a higly scalable version of H.264/AVC," European Broadcasting Union, EBU Technical Review, Edition 2008 Q2.

[22] C. Chatfield, "The Analysis of Time Series: An Introduction," Sixth Edition, Chapman & Hall/CRC 2003.

[23] D. C. Montgomery, L. A. Johnson, J. S. Gardiner, "Forecasting and Time Series Analysis", Second Edition, McGraw-Hill Companies; 2 Sub edition (July 1990), 381 pages.

[24] M . Parulekar and A.M. Makowski, "M/G/∞ input processes: A Versatile class of models for network traffic," INFOCOM 97, pages 419-426 vol.2.

[25] A.M. Dawood, and M. Ghanbari, "Content-based MPEG video traffic modeling," IEEE Transactions on Multimedia, pages 77-87, 1999.

[26] D.P. Heyman and T.V. Lakshman, "Source Model for VBR Broadcast-Video Traffic," IEEE/ACM Transactions on Networking, pages 40-48, 1996.

[27] H. Liu, N. Ansari, and Y.Q. Shi, "Modeling VBR Video Traffic by Markov-Modulated Self-Similar Processes," The Journal of VLSI Signal Processing, pages 101-113, 2004.

[28] O. Lazaro, D. Girma, and J. Dunlop, "H.263 Video Traffic Modeling for Low Bit Rate Wireless Communication," 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. Volume 3, Issue, 5-8 Sept. 2004, pp. 2124-2128.

[29] A. Lombardo et al. , "A Fast Simulation model of MPEG Video Traffic," GLOBECOM 98, pages 702-707 vol.2. 1998.

[30] H. Liu, N. Ansari and Y.Q. Shi, "A Simple Model for MPEG Video Traffic," IEEE International Conference on Multimedia and Expo 2000: pp. 553-556.

[31] C. Ma, and C. Ji, "Modeling Video Traffic Using Wavelets," IEEE International Conference on Communications, pages 559-562 vol.1. 1998.

[32] O. Lazaro, D. Girma, J. Dunlop, "Real Time Generation of Synthetic MPEG-4 Video Traffic Using Wavelets," Proceedings of IEEE VTS 54th Vehicular Technology Conference, Fall 2001 (VTC 2001),Volume 1, pp. 418 - 422.

[33] T. Borsos, "A Practical Model for VBR Video Traffic with Application," Lecture Notes in Computer Science pages 85-95. 2001.

[34] X. Huang, Y. Zhou and R. Zhang, "A Multiscale Model for MPEG-4 Varied Bit Rate Video Traffic," IEEE transactions on broadcasting. Volume:50, Issue: 3, pp. 323- 334.

[35] L. de la Cruz et al., "Self-Similar Traffic Generation Using a Fractional ARIMA Model Application to the VBR MPEG Video Traffic," ITS 98 Proceedings. 102-107 vol.1.

[36] A. Al-Tamimi, R. Jain, and C. So-In, "SAM: A Simplified Seasonal ARIMA Model for Mobile Video over Wireless Broadband Networks," IEEE International Symposium on Multimedia 2008, pp 178-183.

[37] B. Nau, "Duke University, Decision 411 Course," Duke University, URL=[http://www.duke.edu/~rnau/Decision411CoursePage.htm],Feb-28-2009.

[38] G. Box, G. Jenkins, and G. Reinsel,"Time series analysis forecasting and control," Prentice Hall; 3rd edition (February 28, 1994), 592 pages.

[39] J. Aston, et al., "New ARIMA Models for Seasonal Time Series and Their Application to Seasonal Adjustment and Forecasting", U.S. Census Bureau, Research Report, October, 2007.

[40] R. Tsay , "Analysis of Financial Time Series", Wiley-Interscience, 1st edition (October 15, 2001). 472 pages.

[41] JM Reference Software, URL=[http://iphome.hhi.de/suehring/tml/]. March 2010

[42] R project, "The project R of statstical computing," URL=[http://www.r-project.org/], Feb-28-2009.

[43] SAS: Business Analytics and Business Intelligence Software, URL=[http://www.sas.com], July 2009.

[44] YUV Video Sequences, Commonly used video trst sequences in 4:2:0 YUV format for both CIF and QCIF sizes. URL=[http://trace.eas.asu.edu/yuv/index.html].

[45] J. Ozer, "Producing H.264 Video for Flash: An Overview" ,URL=http://www.streaminglearningcenter.com/articles/41/1/Producing-H264-Video-for-Flash-An-Overview/Page1.html.

[46] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile Video Transmission Using Scalable Video Coding," IEEE Transactions on Circuits and Systems for Video Technology, Volume: 17, Issue: 9 On page(s): 1204-1217.

[47] G. Van der Auwera, P. David, and M. Reisslein, "Traffic and Quality Characterization of Single-Layer Video Streams Encoded with the H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding Extension," IEEE Transactions on Broadcasting, Volume 54, Issue 3, pp 698-718.

[48] Fraunhofer HHI, "The scalable Video Coding Amendment of the H.264/AVC Standard," Instititue Nachrichtentechnik Heinrich-Hertz-Institute, 2008.

[49] Y. Sun, and J.N. Daigle, "A Source Model of Video Traffic Based on Full-Length VBR MPEG4 Video Traces," IEEE Global Telecommunications Conference, 2005. GLOBECOM'05, Volume 2, 28 Nov.-2 Dec. 2005, 5 pp.

[50] O. Lazaro, D. Girma, J. Dunlop "A Study of Video Source Modeling for 3G Mobile Communication Systems," Proceedings of First International Conference on 3G Mobile Communication Technologies, 2000, Conf. Publ. No. 471, pp. 461-465.

[51] Y. Shu, M. Yu, J. Liu, and O.W.W. Yang "Wireless Traffic Modeling and Prediction Using Seasonal ARIMA Models," Proceedings of ICC'03, Volume 3, 11-15 May 2003, pp. 1675-1679.

[52] O. Briet, "gsarima: Two functions for Generalized SARIMA time series simulation," GSARIMA package for R. URL=[http://cran.fyxm.net/web/packages/gsarima/index.html], Feb-28-2009

[53] ARMA defnintion from the business dictionary , URL = [http://www.businessdictionary.com/definition/autoregressive-moving-average-ARMA-model.html].

[54] RFC-3550, "RTP: A Transport Protocol for Real-Time Applications".July 2003

[55] WiMAX Forum, "WiMAX System Evaluation Methodology V2.1," Jul. 2008, 230 pp. URL=[http://www.wimaxforum.org/technology/documents].

[56] IEEE P802.16Rev2/D2, "DRAFT Standard for Local and metropolitan area networks," Part 16: Air Interface for Broadband Wireless Access Systems, Dec. 2007, 2094 pp.

[57] B. Kim, C. So-In, J. Yun, R. Jain, Y. Hur, and A. Al-Tamimi, "Capacity Estimation and TCP Performance Enhancement over Mobile WiMAX ," IEEE Wireless Comm. Mag., vol. 47, no. 6, pp. 132-141, Jun. 2009.

[58] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," in Proc. ACM SIGCOMM Communication Review, 1995, vol 25, no. 4, pp. 231-242.

[59] C. So-In , R. Jain and A. Al-Tamimi, "A Deficit Round Robin with Fragmentation Scheduler for IEEE 802.16e Mobile WiMAX," in Proc. IEEE Sarnoff Symposium., 2009, pp. 1-7.

[60] M. Andrews, "Probabilistic end-to-end delay bounds for earliest deadline first scheduling," in Proc. IEEE Computer Communication Conf., Israel, 2000, vol. 2, pp. 603-612.

[61] WiMAX Forum, "The Network Simulator NS-2 MAC+PHY Add-On for WiMAX," Aug. 2007. Available to WiMAX Forum members.

[62] R. Jain, D.M. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems," in DEC Research Report TR-301, 1984, 38 pp., URL=[http://www.cse.wustl.edu/~jain/papers/fairness.htm]

[63] R. Hyndman, Y. Khandakar, "Automatic Time Series Forecasting: The forecast Package for R", Journal of Statistical Software, Vol. 27, Issue 3, Jul 2008

[64] A. Al-Tamimi, R. Jain, and C. So-In, "Modeling and generation of AVC and SVC-TS mobile video traces for broadband access networks," In Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems (Phoenix, Arizona, USA, February 22 - 23, 2010). MMSys '10. ACM, New York, NY, 89-98. DOI= http://doi.acm.org/10.1145/1730836.1730848.

[65] A. Al-Tamimi, C. So-In, C., and R. Jain, "Modeling and Resource Allocation for Mobile Video over WiMAX Broadband Wireless Networks," IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Wireless Video Transmission, vol. 28, no. 3, pp. 354-365, April 2010.

[66] P. Manzoni, P. Cremonesi, G. Serazzi, "Workload Models of VBR Video Traffic and Their Use in Resource Allocation Policies," IEEE/ACM Transactions on Networking (TON),Volume 7,Issue 3 (June1999),pp.387-397.

[67] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," 20th Conference on Local Computer Networks, 1995.,Volume , Issue , 16-19 Oct 1995 pp.397-406.

[68] I. Lanfranchi,"MPEG-4 AVC traffic analysis and bandwidth prediction for broadband cable networks", MS Thesis, 2008, Georgia Tech.

[69] SAM model Traces website, URL=[http://www.cse.wustl.edu/~jain/sam/index.html]

[70] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, "Time Series Analysis: Forecasting and Control," Wiley Series in Probability and Statistics. 4th Edition. ISBN-10: 0470272848..

[71] The Top 5 video streaming websites, URL=[http://www.techsupportalert.com/top-5-video-streaming-websites.htm], fetched on March-20-2010.

[72] YouTube HD video section, URL=[http://www.youtube.com/HD].

[73] MediaInfo, MediaInfo supplies technical and tag information about your video or audio files, URL=[http://mediainfo.sourceforge.net/en].

[74] Digital Rapids, URL= [http://www.digital-rapids.com/downloads/docs/DR_AVC_Flash9.pdf], March 2010.

[75] FFMPEG Coding Library, Cross-platform solution to record, convert and stream audio and video. URL=http://ffmpeg.org.

[76] x264 Encoder, URL=[http://www.videolan.org/developers/x264.html]. March 2010.

[77] JM Reference Software, URL=[http://iphome.hhi.de/suehring/tml/]. March 2010.

[78] B. Everitt , "An R and S-Plus® Companion to Multivariate Analysis," Springer 2007.

[79] H. T. Kaiser, "The Application of Electronic Computers to factor analysis", Educ. Psychol. Meas. 20:141-51, 1960.

[80] R. B. Cattel, "The scree test for the number of factors," Multivariate Behavioral Research, Vol. 1, No. 2. (1966), pp. 245-276.

[81] G. Raîche, M. Riopel, J. Blais, "Non Graphical solutions for the cattell's scree test", Annual meeting of the Psychometric Society, Montréal, Canada.

[82] G. Kootstra, "Project on exploratory Factor Analysis applied to foreign language learning," 2004.

[83] M. Norusis, "SPSS 17.0 Statistical Procedures Companion," Prentice Hall, 2009.

[84] C. Ding, X. He, "K-means clustering via principal component analysis," ACM Inter. Conf. Proceeding Series; Vol. 69, 2004.

[85] Machaon Clustering and Validation Environment, Cluster Validity Algorithms, URL=[http://machaon.karanagai.com/validation_algorithms.html], March-2010.

[86] H. Feng and Y. Shu, "Study on network traffic prediction techniques", International Conference on Wireless Communications, Networking and Mobile Computing, 2005,pp.1041–1044, September 2005.

[87] J. A. Nelder, R. Mead "A simplex algorithm for function minimization," Computer Journal 7, pp. 308–313.1965.

[88] H. Zhao, N. Ansari, Y. Shi, "Efficient Predictive Bandwidth Allocation for Real Time Videos", IEICE Transactions on Communications., vol. E86-B, No.1 2003.

[89] A. Al-Tamimi, R. Jain, C. So-In, "Modeling and Prediction of High Definition Video Traffic: A Real-World Case Study", The Second International Conferences on Advances in Multimedia, in press, MMEDIA 2010.

[90] R. Hyndman, A. Kostenko, "Minimum Sample Size Requitrements for Seasonal Forecasting Models", Foresight: the International Journal of Applied Forecasting (2007), 6, 12-15.

[91] J.-Y. Choi, J. Shin, "Content-Aware Packet-Level Interleaving Method for Video Transmission over Wireless Networks," Lecture Note in Computer Science. 2005, ISSU 3510, pages 149-158.

[92] A. M. Adas, "Using adaptive linear prediction to support real-time VBR video under RCBR network service model," IEEE/ACM Trans. Netw. 6, 5 (Oct. 1998), 635-644. DOI= http://dx.doi.org/10.1109/90.731200.

[93] H. Liu, N. Ansari, and Y. Shi, Y., "Dynamic Bandwidth Allocation for VBR Video Traffic Based on Scene Change Identification. In Proceedings of the international Conference on information Technology: Coding and Computing (Itcc'00) (March 27 - 29, 2000). ITCC. IEEE Computer Society, Washington, DC, 284. 7.

[94] M. Wu et al., "Dynamic resource allocation via video content and short-term traffic statistics," IEEE Transactions on Multimedia, vol. 3, no. 2, pp. 186-199, 2001.

[95] P. Bocheck, and S.-F Chang, "Content-based VBR traffic modeling and its application to dynamic network resource allocation" , Res. Rep. 48c-98-20, Columbia University, New York, NY, USA, 1998.

[96] R.H. Kwong , and Johnston E.W. "A variable step size LMS algorithm," IEEE Trans. Signal Processing, vol. 40, pp. 1633–1642, July 1992.

[97] H. Zhao, N. Ansari, N. and Y.Q. Shi, "A Fast Non-Linear Adaptive Algorithm for Video Traffic Prediction," In Proceedings of the international Conference on information Technology: Coding and Computing (April 08 - 10, 2002). ITCC. IEEE Computer Society, Washington, DC, 54.

[98] The Top 5 video streaming websites 2010, (online) URL=[http://www.techsupportalert.com/top-5-video-streamingwebsites.htm], (March 2010).

[99] JM Reference Software, (online) URL=[http://iphome.hhi.de/suehring/tml/], (March 2010).

[100] B. Melamed, and D. E. Pendarakis, "Modeling full-length VBR video using markov-renewal-modulated TES models," IEEE J. Select. Areas Commun., vol. 16, pp. 600–611, June 1998.

[101] S. G.-M., and M. Wu, "Efficient bandwidth resource allocation for low-delay multiuser MPEG-4 video transmission," In IEEE International Conf. on Communications, volume 3, pages 1308-1312, 2004.

[102] T. Szigeti, "End-to-End QoS Network Design: Quality of Service in LANs, WANs. and VPNs," Cisco Press. Illustrated Edition. ISBN-10: 1587051761, 2004.

[103] DSL Forum Technical Report, "Triple-play Service Quality of Experience (QoE) Requirements," TR-126. (online) URL=[http://www.broadband-forum.org/technical/download/TR-126.pdf] (April 2010).

[104] H. Zhao, N. Ansari, and Y.Q. SHI, "Delay guaranteed bandwidth allocation for real-time video delivery," IEEE Proc. Commun., 2004, 151, (6), pp. 553–558.

[105] G. J. Janacek, "Practical Time Series," Arnold Texts in Statistics Series. ISBN-10: 0340719990, 2001.

[106] Mobile video traces, Mobile Devices: Video Traces, Arizona State University (ASU) and Aalborg University (AAU) (online) URL=[http://mobiledevices.kom.aau.dk/research/traffic and channel measurements/video traces/]. (April 2010).

[107] G. Van der Auwera, P. T. David, M. Reisslein, and L. J. Karam, "Traffic and quality characterization of the H.264/AVC scalable video coding extension," Adv. MultiMedia 2008, 2 (Jan. 2008), 1-27. DOI= http://dx.doi.org/10.1155/2008/164027.

[108] TestQuest white paper, "Ensuring Quality for Wireless Multimedia Services Through Automated Testing," URL=[http://www.testquest.com/objects/Library/appnote_mvtt.pdf]

Curriculum Vitae

Abdel-Karim Al-Tamimi

Date of Birth August 25, 1981

Place of Birth Hebron, Palestine.

Degrees

Ph.D. Computer Engineering, Washington University, August 2010.
M.S. Computer Engineering, Washington University, May 2007.
Bachelor in Computer Engineering, Yarmouk University, Jordan, May 2004.

Journal Papers

A. Al-Tamimi, C. So-In, and R. Jain, "**Modeling and Resource Allocation for Mobile Video over WiMAX Broadband Wireless Networks**," IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Wireless Video Transmission, vol. 28, no. 3, pp. 354-365, April 2010.

A. Al-Tamimi, R. Jain, C. So-In, "**High Definition Video Streams Analysis, Modeling and Prediction**," submitted to IEEE Transactions on Multimedia, Special Issue on Multimedia Streaming, March 2010.

A. Al-Tamimi, R. Jain, C. So-In, "**Content-based Dynamic Resource Allocation for AVC and SVC HD Video Streams**," submitted to ACM Transactions on Multimedia Computing, Communications and Applications, May 2010.

B. Kim, J. Yun, Y. Hur, ,C. So-In, R. Jain, A. Al-Tamimi, "**Capacity estimation and TCP performance enhancement over mobile WiMAX networks**," IEEE Communications Magazine, special issue on Mobile WiMAX, Vol. 47, Issue 6, June 2009, pp. 132-141

C. So-In, R. Jain, and A. Al-Tamimi, "**Scheduling in IEEE 802.16e Mobile WiMAX Networks: Key Issues and a Survey**," IEEE Journal on Selected Areas in Communications (JSAC), Vol. 27, No. 2, Feb 2009

R. Jain, C. So-In, and A. Al-Tamimi , "**System Level Modeling of IEEE 802.16e Mobile WiMAX Networks: Key Issues** ," IEEE Wireless Comm. Mag., Oct. 2008.

C. So-In, R. Jain, and A. Al-Tamimi, "**Capacity Evaluation for IEEE 802.16e Mobile WiMAX**," Journal of Computer Systems, Networks, and Communications, Special issue on WiMAX, LTE, and WiFi Interworking, Vol. 1, No. 1, April 2010.

Conference Papers

A. Al-Tamimi, R. Jain, and C. So-In, "**Modeling and generation of AVC and SVC-TS mobile video traces for broadband access networks**". In Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems (Phoenix, Arizona, USA, February 22 - 23, 2010). MMSys'10.

A. Al-Tamimi, R. Jain, C. So-in, "**SAM: A Simplified Seasonal ARIMA Model for Mobile Video over Wireless Broadband Networks**," Proceedings of IEEE Tenth International Symposium on Multimedia, ISM2008, pp 178-183, ISBN: 978-0-7695-3454-1.

A. Al-Tamimi, R. Jain, C. So-in, "**Statistical Analysis and Modeling of High Definition Video Traces**", International Conference on Multimedia and Expo, ICME 2010, in press.

A. Al-Tamimi, R. Jain, C. So-In, "**Modeling and Prediction of High Definition Video Traffic: A Real-World Case Study**", The Second International Conferences on Advances in Multimedia, MMEDIA 2010, in press.

A. Al-Tamimi, R. Jain, C. So-In, "**Dynamic Resource Allocation Based on Online Traffic Prediction for HD Video Streams,**" submitted to The Proceedings of the Eighteen ACM International Conference on Multimedia, (25-29 October *2010*, Florence, Italy), MM'10.

A. Al-Tamimi, C. So-In, and R. Jain, "**Modeling and Resource Allocation for HD Videos over WiMAX Broadband Wireless Networks**," Invited to IEEE COMSOC MMTC E-Letter, Vol.5, No.3, May 2010.

C. So-In, R. Jain, and A. Al-Tamimi, "**A Deficit Round Robin with Fragmentation Scheduler for IEEE 802.16e Mobile WiMAX**," Proceedings 2009 IEEE Sarnoff Symposium, Princeton, NJ, Mar 30-Apr 1, 2009

C. So-in, R. Jain, A. Al-Tamimi, "**OCSA: An algorithm for Burst Mapping in IEEE 802.16e Mobile**," Proceedings of the 15th Asia Pacific Conference on Communications 2009.

C. So-in, R. Jain, A. Al-Tamimi, "**SWIM: A Scheduler for Unsolicited Grant Service (UGS) in IEEE 802.16e Mobile WiMAX Networks**," Proceedings of 2009 Fourth International Conference on Access Networks 2009.

C. So-In, R. Jain, A. Al-Tamimi, "**eOCSA: An Algorithm for Burst Mapping with Strict QoS Requirements in IEEE 802.16e Mobile**

WiMAX Networks," Proceedings of the Second IFIP Wireless Days Conference 2009.

C. So-In, R. Jain, A. Al-Tamimi, "**Generalized Weighted Fairness and its Application for Resource Allocation in IEEE 802.16 WiMAX Networks**," Accepted to appear in Proc. the 2nd Int. Conf. on Computer and Automation Engineering, Singapore, Feb. 2010.

M. Al-Jarrah, A. Al-Tamimi, "**A Thin Security Layer Protocol over IP Protocol on TCP/IP Suite for Security Enhancement**," Innovations in Information Technology, 19-21 Nov. 2006.

Book Chapters

Chakchai So-In, Raj Jain, and Abdel-Karim Al-Tamimi, "**Resource Allocation in IEEE 802.16e Mobile WiMAX**," Orthogonal Frequency Division Multiple Access Fundamentals and Applications, Editors: Tao Jiang; Lingyang Song; Yan Zhang, CRC Press, 9781420088243, 608 pp, April 2010.

Abdel-Karim Al-Tamimi and Raj Jain, "**Security in Wireless Data Networks**," Handbook on Security and Networks Edited by Prof. Yang Xiao, Prof. Frank H. Li, and Prof. Hui Chen, to be published by World Scientific Publishing Co., ISBN:9814273031, 1000 pp, Dec 2010.

Technical Papers in Submission

C. So-In, R. Jain, and A. Al-Tamimi, "**Virtual ID: ID/Locator Split in a Mobile IP Environment for Mobility, Multihoming, and Location Privacy**," in submission to *Int. J. of Internet Protocol Technology (IJIPT*), Feb. 2010.

C. So-In, R. Jain, and A. Al-Tamimi, "**Deficit Round Robin with Fragmentation Scheduling to Achieve Generalized Weighted Fairness for Resource Allocation in IEEE 802.16 WiMAX Networks**," in submission to *J. Future Internet*, March 2010.

C. So-In, R. Jain, and A. Al-Tamimi, "**SWIM: A Scheduler for Unsolicited Grant Service (UGS) in IEEE 802.16 WiMAX Networks**," in submission to *IEEE Systems J.*, Feb. 2010. (extended version)

WiMAX Forum Contributions

A. Al-Tamimi, R. Jain, and C. So-In, "**Mobile TV Workload Characterization**," Presented at WiMAX Forum Application Working Group meeting at Washington, DC, November 19-20, 2007.

C. So-In, R. Jain, and A. Al-Tamimi, "**Scheduling in WiMAX: Baseline Multi-class Simulations**," WiMAX Forum Application Working Group meeting, Washington DC, November 19-20, 2007.

A. Al-Tamimi and R. Jain, "**Video Traffic Modeling Using Seasonal ARIMA Models**," WiMAX Forum Face-to-Face Meeting, Hawaii, January 30, 2008.

A. Al-Tamimi, R. Jain, "**A Simple and General Model for Mobile Video Workload Generation**," WiMAX Forum Application Working Group meeting, Macau, September 24, 2008.

C. So-in, R. Jain, A. Al-Tamimi, "**A Deficit Round Robin with Fragmentation Scheduler for Mobile WiMA**X," WiMAX Forum Application Working Group Meeting, Orlando, FL, February 4-9, 2009.

other HD Video Traces Library, SAM model website and results, http://www.cse.wustl.edu/~jain/sam

August 2010