Playback Delay and Interruption Analysis of Multiuser Video Streaming Systems

by
Mehmet Emre Özfatura

Submitted to the Graduate School of Engineering
and Natural Sciences in partial fulfillment of the requirements
for the degree of Master of Science

Sabancı University
January 2015

Playback Delay and Interruption Analysis of Multiuser Video Streaming Systems

by Mehmet Emre Özfatura

APPROVED BY

Assoc. Prof. Dr. Özgür Erçetin (Thesis Advisor)	....
Assoc. Prof. Dr. Hazer İnaltekin (Thesis Co-Advisor)	....
Assoc. Prof. Dr. Özgür Gürbüz	....
Assoc. Prof. Dr. Müjdat Çetin	....
Assoc. Prof. Dr. Albert Levi	....

DATE OF APPROVAL: January 05, 2015

"Ever tried. Ever failed. No matter. Try Again. Fail again. Fail better."

Samuel Beckett

To my family

Playback Delay and Interruption Analysis of Multiuser Video Streaming Systems

Mehmet Emre Özfatura

MSc Thesis, 2015

Thesis Advisor: Assoc. Prof. Dr. Özgür Erçetin
Thesis Co-Advisor: Assoc. Prof. Dr. Hazer İnaltekin

Keywords: ***Wireless Scheduling, Rate Splitting policy, Video on Demand, Interruption, jitter , multiuser***

Playback delay control is an important mechanism to avoid jitter in video streaming systems. This work introduces a playback delay minimization problem for multiuser video streaming systems providing a jitter-free video streaming service to end users in the system. We first analyze the case where the video requests of the users arrive to server at the same time in other words video requests are synchronized. Then we extend our analysis to case of asynchronised video requests where the number of users varies by the time

In particular, a necessary condition on the playback delay for jitter-free streaming is obtained. Then, based on the derived necessary condition, an optimum rate splitting algorithm that splits available rate to all users is proposed. The proposed algorithm is optimum in the sense that it achieves the minimum system delay, which is defined as the maximum of all initial playback delays, while ensuring jitter-free streaming service to all users. Finally, using these results, an expression for the minimum system delay as a function of system parameters such as total rate, arrival times and playback curves of requested video files is also derived.

Further, in this work we investigated random channel model and a closed form expression for interruption probability is presented. In addition it is verified that nearest deadline first algorithm is the optimal scheduling policy for video streaming over heterogeneous channels in a sense that it provides the minimum system jitter probability.

Acknowledgments

I cannot express enough thanks to my supervisor, Dr. Özgür Erçetin for his continued support, encouragement and patience. I must emphasize that my completion of this thesis could not have been accomplished without the support of Dr. Özgür Erçetin. Although, sometimes I felt that I have been stuck and can not move any further in my studies, I have not been lost thanks to his priceless understanding and concern. I am not sure many graduate students have an opportunity to work with such a kind and knowledgeable supervisor. Therefore, I count myself lucky to have worked with him. For everything you have done for me, Dr. Özgür Erçetin , I honestly thank you.

I am also grateful to Dr. Hazer İnaltekin for his great contribution my thesis. I really appreciate his guidance in my research. His support is not limited with research but also in the writing process of the research outcomes. I must emphasize that Dr. Hazer İnaltekin has been always kind to me and he have widen my knowledge with his valuable ideas. Without his endless support and helpful comments it would be very difficult to deal with problems that I have encountered.

I also want to thank TÜBİTAK for supporting my master education with BİDEB schoolarship.

I also would like to express my gratitude to my M.Sc oral examination committee members Dr. Özgür Gürbüz, Dr. Müjdat Çetin, and Dr. Albert Levi for taking their time serving on defense exam committee, and I thank them for kindly reading the thesis and their valuable comments.

Contents

Abstract **vi**

Contents **vii**

List of Figures **x**

List of Tables **xi**

1 Introduction **1**

1.1 Motivation . . . 1
1.2 Problem Formulation . . . 3
1.3 Contributions . . . 4
1.4 Organization of Work . . . 5

2 Literature Review **6**

2.1 Statistical Delay Bound . . . 6
2.2 Rate Distortion Analysis . . . 7
2.3 Interruption Probability . . . 9
2.4 Dynamically Adaptive Streaming over HTTP . . . 10
2.5 Energy Efficiency . . . 10

3 Multiuser Video Streaming: Playback Analysis **11**

3.1 Introduction . . . 11
3.1.1 Cumulative Rate Curve . . . 12
3.1.2 Playback Curve . . . 13
3.1.3 Receiver Curve and System Delay . . . 13
3.2 Multiuser Video Streaming for Synchronized Video Requests . . . 15
3.2.1 Optimal Rate Splitting Policy and Minimum System Delay . . . 15
3.2.2 Numerical Analysis . . . 20
3.3 Multiuser Video Streaming for Asynchronized Video Requests . . . 22
3.3.1 Preliminaries . . . 24
3.3.2 Delay Optimal Rate Splitting Policy . . . 25
3.3.3 Numerical Results . . . 31

4 Multiuser Video Streaming: Interruption Analysis **34**

4.1 Jitter Probability for Single User . . . 34
4.1.1 Upper Bound for Jitter Probability . . . 35
4.1.2 Buffer Starvation . . . 36

4.1.3 Exact Calculation of Jitter Probability . . . 36
4.1.4 Approximation for the Jitter Probability . . . 37
4.1.5 Numerical Results . . . 38
4.2 Video Transmission to Multi-end Users Over Heterogeneous Channels . . 44
4.2.1 Introduction . . . 44
4.2.2 Preliminaries . . . 44
4.2.2.1 Binomial Identities . . . 44
4.2.2.2 Nearest Deadline First Algorithm . . . 46
4.2.3 Optimal Scheduling Policy . . . 47
4.2.4 Numerical Results . . . 48

5 Conclusion and Future Works **50**
5.1 Conclusions . . . 50
5.2 Future Works . . . 51

A NDF Algorithm **52**
A.1 Theorem 3 . . . 52

Bibliography **54**

List of Figures

3.1 Multiuser video streaming system 11
3.2 Receiver, Playback and Playout curves 12
3.3 Cumulative Rate Curve and Cumulative Playout Curve 18
3.4 Experiment Setup 20
3.5 Required Channel Capacity for Playback Delay of 30 Seconds 21
3.6 Required Channel Capacity for Playback Delay of 60 Seconds 22
3.7 Example case with 3 Users 22
3.8 Analysis of user acceptance and total video transfer for channel rate of 1Mb/s 32
3.9 Analysis of user acceptance and total video transfer for channel rate of 2Mb/s 33

4.1 R_{best} Curve 35
4.2 Minimum slope $\gamma_{\min}$ 38
4.3 Playback Curves for Different Video Durations 39
4.4 Jitter Probability Analysis For Video Duration of 20 Seconds 40
4.5 Jitter Probability Analysis For Video Duration of 40 Seconds 41
4.6 Jitter Probability Analysis For Video Duration of 40 Seconds for fixed ρ . 42
4.7 Jitter Probability Analysis For Video Duration of 100 Seconds 43
4.8 Playout curves at time slot l_c 46
4.9 Simulation Result Vs Theoretical Result 49

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recently, there is an increasing demand for multimedia streaming applications thanks to the ubiquity of internet access, the availability of online content and the growing usage of wireless hand-held devices. The predictions of Cisco Visual Networking Index [1] indicate that IP video traffic will constitute 79 percent of all consumer internet traffic in 2018. Especially, on demand video providers have gained importance in the current decade. For instance, in 2014, YouTube and Netflix accounts for up to 49 percent of fixed access Internet traffic in North America [2]. Furthermore, 20 percent of the mobile internet traffic in North America is solely based on YouTube [2]. The upward trend in the interest for the video on demand (VOD) services such as IPTV is expected to accelerate further during the next decade [3].

As the VOD global market is growing fast, certain system design issues have to be rethought for video streaming systems such as appropriation of video server bandwidth. In particular, for a commercialized and efficient video streaming system, the service providers do not only have to assure that the network infrastructure has the sufficient resources to serve the users according to their desired level of service quality such as video quality and playback delay, but also allocate available communication resources optimally among the users according to their desired level of service quality. It may even be necessary for the VOD service provider to control admissions to the network so as to prevent existing users from observing lower than promised level of quality of service.

In the literature, there are manifold studies regarding video streaming systems. In addition, various QoS metrics are proposed to quantify the service quality provided to end user. One of the prevalent QoS metrics is the statistical delay bound which describes the probability that packet delay exceeds the predefined delay-bound [4]. In [5–16] it is investigated how to guarantee a statistical delay bound for end users while increasing rate (throughput), minimizing resource utilization, maximizing the video quality or minimizing energy consumption. Statistical delay-bound is a powerful metric to analyze wireless networks, however presented works do not explicitly underline the relationship between the delay-bound and video quality at the end user.

Hence, it is investigated to quantify the distortion in the video stream due to rate varying structure of the channel. In these studies quantified distortion in the video quality is considered as a QoS metric and an utility function is defined based on the average rate, throughput or packet loss. [17–25]. Then it is aimed to maximize the utility function for users. Hence, these works are concentrated on rate-distortion analysis for the video stream. In [26–29], layered structure of the encoded video file is also take in to the consideration to measure the distortion. Different from the traditional utility-based approaches, delay-constrained video transmission schemes have been considered in [30–33]. Although, papers ,discussed so far, provide different techniques for scheduling, rate allocation, admission control and rate adaptation to maximize the utility of the users based on the video quality, these works could not decently identify the received video quality from the user perspective. For instance, continuous fluctuations in the video quality is also an undesirable situation for users, even though average video quality is high during the streaming process. Therefore, recent works aim to identify the distortion in video file from the user's aspect. Therefore, notion of the subjective video quality is analyzed in [34],[35],[36]. However, in our work we ensure that video quality is preserved during the whole streaming session. In other words, encoded video files are sent with fixed frame configuration and quantization step size.

Minimum playback delay for user to display video without an interruption is introduced in [37]. Further, there are several works aim to derive a closed form expression for the interruption probability [37–42]. However, all these works concentrate on the single user case, whereas we consider a video streaming system with multiple end users. Moreover, we propose a rate splitting policy for minimum system delay and a scheduling policy to minimize interruption probability. Scheduling policy to prevent interruption in a multiuser streaming is proposed in [43]. Nevertheless, the proposed algorithm is a best effort method and unable to provision the interruption probability. Although the playback delay optimization problem for multiuser video streaming systems was investigated in [44], the authors assume that clients subscribe to different portions of scalable video streams. In our system, on the other hand, each user is able to request a different video stream from the server.

1.2 Problem Formulation

For maximum end-user experience, we expect each user to display the requested video stream without an interruption. An interruption occurs when the required data for video is not available in the user buffer on time. This interruption event is called jitter. A simple and efficient way to resolve jitter is to start displaying the video stream after a sufficiently long playback delay. Note that a long playback delay is not preferred by users. Hence, the playback delay should be minimized as far as possible whilst satisfying the promised level of quality of service to all users.

In the first part of our work, we assume that the rate of the transmission channel is varying over time, but such channel variations during the video streaming session are known in advance. The maximum possible latency of a user is the determining factor of QoS. Hence, we focus on minimizing the initial playback delay for an uninterrupted video streaming service in a multiuser setting for a known but dynamic service bandwidth, and for a given set of users' video requests. In this setting, we concentrate on the transmission of pre-encoded video files to a multitude of end users. In a general multiuser video streaming system, video requests from different users may arrive at the server at different times. However, for the sake of simplicity we start our analysis from the case of synchronized video requests.

We first find a necessary condition for the proposed multiuser video streaming system to provide jitter-free video streaming to all users simultaneously. Next, we propose a rate splitting control policy for the server such that when the derived necessary condition is satisfied, then all users receive jitter-free video streaming service. Finally, by utilizing the proposed rate splitting policy, we determine the minimum possible playback delay while still achieving jitter-free video streaming to all users.

Thereafter, we investigate the case of asynchronised video requests. Remark that the number of users in the system is changing as time goes by due to newly arriving users. To deal with this, we consider the whole video streaming process as a concatenation of mutually exclusive time intervals (frame). Each frame corresponds to a system with a constant number of users such that video transmission for each user starts at the same time. Then, we utilize our inferences from the case of synchronized video requests to determine the minimum possible playback delay while still achieving jitter-free video streaming to all users for asynchronised video requests. An important practical implication of these results is that they indicate the service providers if and when their network infrastructure is adequate for meeting certain level of QoS requirements measured in terms of playback delays for jitter-free multiuser video streaming.

In the second part of this work, we assume that time is divided into fixed length time slots. We further assume that transmission channel experience independent and identically distributed (iid) block fading such that channel gain is constant over a time slot and it varies between the time slots independently. The random behaviour of the channel is defined by the packet drop rate. In this case, we consider the playback delay is a default system parameter.

We first examine the different methods used for provisioning the jitter probability when there is a single user in the system [42],[37],[38]. Then, we introduce a recursive method to measure exact jitter probability for given channel statistic, playback delay and playback curve. Next, we propose a closed form expression for the approximate jitter probability. Finally, we compare the performance of all these methods for different video lengths and varying playback delayes.

Further, we extend our analysis to multiuser video streaming over heterogeneous channels. Our main purpose is to identify the optimal scheduling policy for a multiuser system in a sense that for the presented scheduling policy the minimum system jitter probability is achieved. In here, minimum system jitter probability defines the probability that for at least one user video interruption occurs. In this work, we show that nearest deadline first algorithm provides the optimal scheduling policy.

1.3 Contributions

The main contribution of our work is providing a novel framework to solve the delay optimal resource allocation problem in the multiuser video streaming system over a common transmission channel. We establish a necessary condition for providing interruption-free video streaming to all users, and we find the minimum possible playback delay to avoid interruption during the video streaming process. We further extend our study to a random channel model where only the information of packet drop rate is available. Then we present a closed form expression for the interruption probability when there is a single user in the system. Finally, we verify that nearest deadline first (NDF) algorithm is the optimal scheduling policy for video streaming over heterogeneous channels in the sense that it minimizes the system interruption probability.Further we point out that it is possible to provision the system jitter probability under the assumption that all transmission channels are identical.

1.4 Organization of Work

This work is organized in the following way. Second chapter is allocated to literature review. In Chapter 3, we introduce the playback delay analysis of the multiuser video streaming system. We explain the system model for multiuser video streaming system in 3.1. Synchronous and asynchronous video requests are investigated in 3.2 and 3.3 respectively. In Chapter 4, we introduce the Interruption analysis of the multiuser video streaming system. Methods for quantifying the jitter probability are provided in 4.1 and in 4.2 we examine multi-user video streaming over heterogeneous channels. Finally, in Chapter 5 we provide the conclusions and mention the future works.

Chapter 2

Literature Review

2.1 Statistical Delay Bound

Quality of service (QoS) provisioning in wireless networks has been an active research area. Manifold studies interest in statistical (QoS) due to varying structure of the wireless channel. Hence, the concept of effective capacity is a powerful tool to characterize the wireless channel in terms of functions that provides a statistical link-layer QoS metric. The concept of the effective capacity is constructed upon on the theory of effective bandwidth. Effective bandwidth provides a statistical delay bound for packets [45] in a queuing system. The concept of effective bandwidth considers constant channel capacity with varying source rate whereas effective capacity deals with constant source rate and varying channel capacity. To clarify the effective capacity method, let $D(t)$ denotes the delay experienced by a packet arriving at time t. Then, it claims that probability of $D(t)$ exceeding a delay bound D_{max} is given as

$$\sup_{t} Pr\left\{D(t) \geq D_{max}\right\} \approx \gamma(\mu) \exp^{-\theta(\mu) D_{max}} \tag{2.1}$$

where $\{\gamma(\mu), \theta(\mu)\}$ are function of source rate μ [4]. Hence, for a given delay-bound violation probability ϵ maximum value of μ can be calculated.

In [5],[6], authors consider a multiuser downlink network and propose a scheme consisting of admission control, resource allocation and scheduling algorithms. In these papers proposed schemes, the scheduling algorithm is a combination of Knoop and Hublet's (K&H) scheduling and round robin (RR) scheduling . Through proposed scheduling algorithms, it is aimed to increase multiuser diversity while providing a QoS guarantees to users in the network. The main contribution of these works are enabling explicit provisioning of statistical QoS for multi end-user by virtue of the effective capacity method. Different dynamic resource allocation schemes for wireless networks adopting the effective capacity model is studied in [14],[15],[16]. The proposed schemes aim to increase the throughput performance of network while supporting the QoS requirements.

Cross-layer modeling is introduced in [7] and [8] to characterize the QoS performance at the data link layer utilizing the effective capacity concept. The significant feature of the introduced model is that physical layer service process is modeled as a finite state Markov chain (FSMC). In [9], the authors proposed an adaptive resource allocation scheme for downlink heterogeneous wireless networks based on the cross-layer model. According to the proposed scheme, assigned power levels and time slots to mobile users are dynamically updated according to the statistical QoS delay-bound. In [10] QoS provision for multilayered video files is investigated and an adaptive transmission scheme is proposed to minimize resource consumption. However, all these works aim to maximize the rate (throughput) or minimizing the resource consumption which are not directly related to received video quality. On the other hand, in [11] authors aim to maximize the sum of video quality while delay-bounds for end-users are satisfied.

In addition to above previous work, the relationship between the energy efficiency and QoS constraints is also investigated in the literature. In particular, this relationship is monitored and characterized in [12] and an energy efficient power allocation scheme guaranteeing QoS constraints is proposed in [13].

2.2 Rate Distortion Analysis

In the literature, the problem of maximizing the video quality, or equivalently the problem of reducing distortion in the received video file delivered over a wireless medium, is also well-investigated. In [17], an adaptive rate allocation strategy is studied to assure certain QoS requirements considering the packet loss events for multi-layered video transmission. In addition, different distributed scheduling and distributed rate allocation algorithms are proposed for multiuser video streaming [18–24]. There is a strong relation between the rate of a wireless or wireline link and the level of video distortion. In these papers, this relationship is first described. Then, to optimize the system performance, either the weighted sum of the video distortion functions of users is minimized or the weighted sum of the utility function of users is maximized. In [20–24] video distortion is modeled according to parametric model given in [46]. In [18] a distortion parameter assigned to each packet according to their priorities to model the deterioration in the video quality. On the other hand, a rate based utility function is used in [19].

A similar scheduling problem regarding the group of picture (GoP) structure and delay deadlines is investigated in [25]. Each GoP contains finite certain number of data units with different distortion impacts. The proposed method aims to maximize accumulated video quality of all users. One of the distinguishing features of this work is the consideration of long term video quality delivered to users.

Recent studies on optimizing the received video quality in a wireless video streaming systems consider scalable video files. In [26], an adaptive scheduling algorithm for single user is proposed to minimize video distortion where the video distortion is measured by using mean square error in the received data units. Multiuser video streaming systems are also investigated in [27],[28],[29]. In order to quantify the system performance, the mean square error method for video distortion calculation is used in [27],[28], whereas a throughput based utility function is used in [29].

Traditional utility-based approaches often did not explicitly account for stringent deadlines. Therefore, the delay-constrained video transmission schemes have been considered in the literature by using different techniques. In particular, scheduling and resource allocation policies are proposed in [30–33]. Particularly, the authors in [31] authors identify a lower bound on the deadline violation probability, and propose simple policies that achieve the lower bound. In [32], a reward is assigned to each packet meeting its deadline, and the function of total reward accumulated from all packets is maximized. On the other hand, in [33] transmission delay deadlines of each sender's video packets are considered as monotonically-decreasing weight distribution in the time horizon and the proposed non-stationary resource allocation policy schedule the users according to these weights.

The common feature of the works explained above is that video quality is defined through rate, throughput, packet dropping or video layer. Hence, it is aimed to maximize average video quality. However, from the users perspective consistency of the video quality is also an important parameter which is ignored in these papers. Therefore, to visualize the video quality experienced by end users, notion of the subjective video quality is introduced [34],[35],[36]. In [35] quality of experience is characterized by using the second order empirical cumulative distribution function. On the other hand, in [36] it is assumed that frame rate and quantization step-size are two determinant factors of the subjective video quality.

2.3 Interruption Probability

In the literature, many works consider interruption probability as a quality of experience (QoE) metric since the experienced video quality is directly related to occurrence of interruption event. Hence, there are several research on provision of the interruption probability. In [37], the authors first define the minimum initial delay to prevent interruption event when the channel rate is fully characterized. The random channel case where the fluctuations in the channel rate is not known a priori at the receiver is investigated in both [37],[38] and an upper bound for interruption probability is provided. In these works, the random channel is modeled with probability of successful packet transmission.

Another approach to derive the interruption probability is based on the diffusion approximation. According to the diffusion approximation approach [47],[48], if the packet inter-arrival times follow a given but unknown distribution with mean $\frac{1}{\lambda}$ and variance υ_a and similarly packet inter-departure times follow a given but unknown distribution with mean $\frac{1}{\mu}$ and variance υ_s, then the buffer size $X(t)$ is modeled as a Brownian motion such that

$$dX(t) = X(t + dt) - X(t) = \beta dt + G\sqrt{\alpha dt}, \tag{2.2}$$

where G is a random variable with standard normal distribution, $\beta = \lambda - \mu$ is the drift coefficient and $\alpha = \lambda^3 \upsilon_a - \mu^3 \upsilon_s$ is the diffusion coefficient. In [39], an interruption probability is derived for infinite video length using the diffusion approximation and similarly an upper bound for interruption probability is obtained in [40]. Other methods for provisioning the interruption probability are proposed in [41] and [42]. In [41], it is claimed that for a given arrival process with rate R, initial buffer size D and uniform departure process with rate 1, then the interruption probability is given as

$$p(D) = \exp^{-I(R)D}, \tag{2.3}$$

where $I(R)$ is the largest root of $\gamma(r) = r + R(\exp^{-r} - 1)$. In [42], the buffer level of an user is modeled as a M/M/1 queue with Poisson arrival process and Poisson service process. Then, using the Ballot Theorem, a closed form expression for interruption probability is derived. On the other hand, in [49],[50] and [43] scheduling and rate adaptation algorithms are proposed to prevent interruption event. In [49], interruption probability is predicted for short time intervals utilizing the large deviation theory then playout rate is adjusted accordingly. Similarly, in [50] buffer occupancy is measured to adjust the video rate to prevent rebuffer. On the other side, in [43] proposed algorithm control the users's buffer levels and if there is a certain amount of reduction in the buffer levels then decide on the next user to be scheduled.

2.4 Dynamically Adaptive Streaming over HTTP

Besides the above works focusing on the server side of the video streaming systems, there is an increasing interest towards client side of the video streaming applications recently. Thanks to the abundance of Web platforms and broadband connections, HTTP streaming has become the cost effective way of delivering multimedia content [51], [52]. Further, the dynamic adaptive streaming over HTTP (DASH) [52][53] has been proposed. In a DASH system, multiple copies of the pre-encoded videos corresponding to different video quality levels are stored in segments. In HTTP streaming, the Web server usually has very little knowledge about client/network status. Therefore, client decides on the content delivery, and for each segment the client request the appropriate quality version of the video fragment. In the recent works, quality of experience QoE is aimed to be maximized using the DASH standard [54–60].

2.5 Energy Efficiency

Design issues related to video streaming systems are not only based on the quality of service or quality of experience. Increasing mobile data traffic and dense deployment of base stations have made energy efficient implementations increasingly more important for video streaming systems. Besides, it is well known that consumers highly care about battery life times of wireless hand-held devices. Thus, there are many works aim to reduce energy consumption both in the base station side and in the mobile device side [61–66].

Chapter 3

Multiuser Video Streaming: Playback Analysis

3.1 Introduction

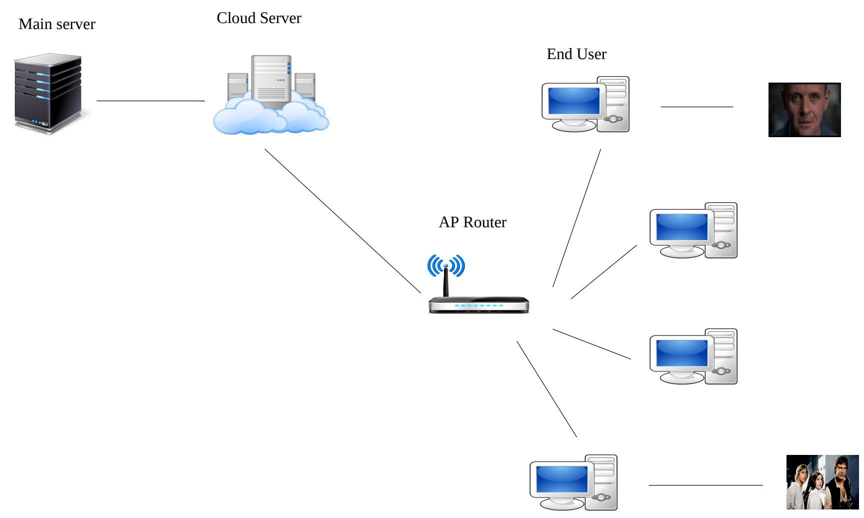


FIGURE 3.1: Multiuser video streaming system

In this chapter, we focus on discovering underlying principles for interruption free delivery of video streams to a multitude of end users. To this end we consider an extension of the video streaming system introduced in [37] to a multiuser setup. In particular, there are N different users requesting N different video streams from a common video streaming server in our case, as shown in Fig.3.1.

The routing between the video server and the router is fixed and determined prior to the network operation. The bottleneck of the network is between the router and the clients.

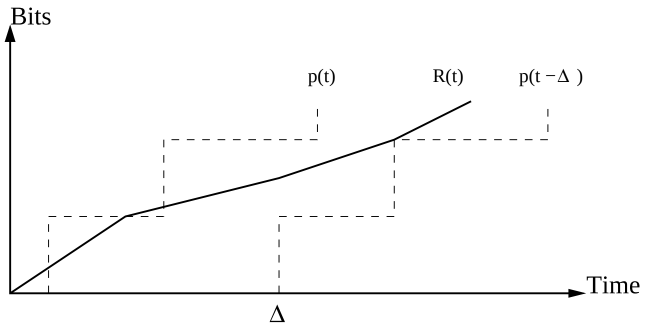


FIGURE 3.2: Receiver, Playback and Playout curves

Router has an infinite buffer space, but we will consider the case for finite buffer as a future work. The end-to-end transmission is performed over UDP protocol, which requires the reliability to be ensured by the application. We consider a cross-layer network stack, where the application layer has direct communication with MAC sub-layer. The effects of any forward error correction and/or application-specific error-correction used in the system are reflected in packet error probability[67], which is independent and identical among time slots. Once a packet is not received correctly, a NAK is sent by the client, which in turn triggers a retransmission by the router. There is no limit on how many times the packet is retransmitted. Otherwise, client sends an ACK which in turn allows the router to drop the packet from its outgoing queue. Our proposed resource allocation algorithms are run at the MAC layer of the bottleneck router serving the clients.

In this chapter, for the playback delay analysis, we assume that packet length is sufficiently small thus time axis is considered as continuous. Before analyze the multiuser video streaming system in regard to the relation between the interruption event and initial waiting time of user, we introduce some system parameters to obtain a rigorous mathematical model.

3.1.1 Cumulative Rate Curve

The data transmission channel can be characterized by a deterministic *rate curve* $r(t)$ that specifies the instantaneous error-free data of the channel. Since the server starts streaming *only after* a video request arrives, it is assumed that $r(t) = 0$ for $t \leq 0$. We construct the *cumulative rate curve* $R(t)$ such that

$$R(t) = \int_0^t r(\tau)d\tau, \tag{3.1}$$

to visualize the total amount of error free data that can be transmitted over the channel up to time t. Obviously, $R(t) = 0$ for $t \leq 0$ and $R(t)$ is monotonically increasing function of t

3.1.2 Playback Curve

In this multiuser streaming setup, it is important to be able to quantitatively characterize the additional rate requirement introduced by each video stream requested by end users. To this end, we introduce the notion of *playback curve* such that the video stream requested by user i is characterized by a playback curve $p_i(t)$, where $i = 1, \ldots, N$ The playback curves describe the minimum amount of data(e.g. measured in bits) that needs to be decoded up to time t in the user buffer. In general, these curves are monotonically increasing. We assume that $p_i(t) = 0$ for $t \leq 0$ to indicate that no data are requested before the arrival of the ith user video request. Remark that we particularly focus on VOD services such that users in the system request a video stream from a certain video library. Furthermore, we assume that quality of the video stream remains constant during the video streaming process, hence it is proper to assume that playback information is known in advance. Therefore, exact information of the playback curves are available in the server, before the transmission of the corresponding video files(e.g., header file of the video stream contains the playback curve information).

A video file is considered as a combination of mutually exclusive fragments that contain different frames of the same video file. In [68], the authors call the video units *group of pictures* (GOP). Note that frames of the same GOP are decoded and encoded together. To display a frame, all information related to the corresponding GOP needs to be available in the buffer. Hence, we assume all playback curves are right-continuous step functions. We call any time instant $t > 0$ with $p(t^-) \neq p(t)$ an *increment point*. The t^- notation denotes any time instant $t - \epsilon$, $\epsilon > 0$, that comes arbitrarily close to t from the left-hand side.

3.1.3 Receiver Curve and System Delay

The *receiver curve* $R_i(t)$ is defined as the overall error-free data allocated to user i up to time t. As before, we assume that $R_i(t) = 0$ for $t \leq 0$. Since information transmitted to a user cannot be retrieved back from this user, each receiver curve is monotonically increasing. Note that in a video streaming system with a single user, the cumulative rate curve and the receiver curve turn out to be the same function , as all communication resources allocated to only one user. In a multiuser setting, on the other hand, we need to handle the issue of distributing the available communication resources to a plurality of end users. Hence, we define the *rate splitting policy* π for further analysis of the proposed multiuser system.

Definition 1. A rate splitting policy π is a method for dividing the cumulative rate curve $R(t)$ to obtain a receiver curve for each user such that

$$R(t) = \sum_{i=1}^{N} R_i(t) \tag{3.2}$$

for all $t \geq 0$, where $R_i(t)$ denotes the receiver curve of the user i under rate splitting policy π.

Throughout the section we assume that for a given cumulative rate curve $R(t)$ and a rate splitting policy π, there exists a sufficiently large constant $T > 0$ such that the following equality

$$R_i(T) = p_i(T_i), \tag{3.3}$$

holds for all $i \in \{1, \dots, N\}$. This assumption indicates that for any rate splitting policy π, each user can download the corresponding video stream in a finite duration. In this chapter, our primary purpose is to identify the conditions under which we can provide uninterrupted video streaming service to all users in the system. We remark that if a user does not receive the required amount of data on time during the video streaming process, an interruption arises while playing out the video at the end user. In the literature this phenomenon is also called *jitter*. One common technique to avoid jitter during video streaming is to start to display the video stream with a delay, after some Δ seconds from the arrival of the video streaming request. Δ here is called the *playback delay*.

We first consider a system with single user, where the receiver curve is equal to the cumulative rate curve $R(t)$. Authors of [37] demonstrate that for jitter-free video streaming, the total number of bits received by the user up to time t should be equal to or larger than $p(t - \Delta)$ bits for any time instant $t > 0$ i.e.,

$$R(t) \geq p(t - \Delta). \tag{3.4}$$

Here, the curve $p(t-\Delta)$ is called *playout curve* and denoted by $p^{\Delta}(t)$. When the receiver curve $R(t)$ and the playback curve $p(t)$ are known before the transmission at the server, then the minimum initial delay to avoid jitter is given as

$$\Delta_{\text{min}} = \max_{t} \left\{ R^{-1}(p(t)) - t \right\}, \tag{3.5}$$

where $R^{-1}(x) = \max \{t : x \leq R(t)\}$ is the so called the pseudo-inverse function of the monotonically increasing receiver curve $R(t)$. The pseudo-inverse function of the playback curve $p^{-1}(t)$ and the receiver curve $R(t)$ are used to determine minimum playback delay in [37]. However, the method given in [37] only provides the minimum playback delay for strictly increasing $R(t)$. On the other hand, the minimum playback delay given in (3.4) is valid for all monotonically increasing receiver curves. Note that Δ_{min} is the minimum value satisfying the inequality $R(t) \geq p(t - \Delta)$ for all $t > 0$

3.2 Multiuser Video Streaming for Synchronized Video Requests

In this section we consider the case where there are multiple end users in system and all the video requests reach to server at the same time particularly at $t = 0$. In contrast to the single user case, there are N users in our system model and each user may have a different initial delay. A *delay vector* $\mathbf{\Delta} = (\Delta_1, \ldots, \Delta_N)$ is used to denote all initial delay values for all users in the system. We say that a multiuser video streaming system is jitter free if (3.4) holds for all users in the system, i.e.,

$$R_i(t) \geq p_i(t - \Delta_i), \tag{3.6}$$

for all t for all i. We denote the playout curve of the user i by $p_i^{\Delta}(t)$ to simplify the notation. For a given rate splitting policy π and a cumulative rate curve $R(t)$, (3.5) provides the minimum playback delay for user i with $p_i(t)$ i.e.,

$$\Delta_i^{\pi} = \max_t \left\{ (R_i)^{-1} (p_i(t)) - t \right\}. \tag{3.7}$$

The minimum playback delay vector $\mathbf{\Delta^{\pi}}$ is a vector that consists of minimum playback delays for the given rate splitting policy π, i.e.,

$$\mathbf{\Delta^{\pi}} = (\Delta_1^{\pi}, \ldots, \Delta_N^{\pi}) \tag{3.8}$$

Definition 2. For a given playback delay vector $\mathbf{\Delta}$, the system delay Δ_s is defined to be the maximum of playback delays, i.e., $\Delta_s = \max_{1 \leq i \leq N} \Delta_i$.

3.2.1 Optimal Rate Splitting Policy and Minimum System Delay

In this subsection, we formally introduce our design problem for the multiuser video streaming system in question as the minimization of the system delay Δ_s while all users receive jitter-free video streaming service simultaneously. That is, we want to find the optimum rate splitting policy $\pi^\star$ minimizing Δ_s over all feasible rate splitting policies π while inequality (3.6) holds for all $i = 1, \dots, N$ and for all t. It turns out that finding the optimum jitter-free rate splitting policy minimizing the system delay is equivalent to the following minimization problem

$$\operatorname*{argmin}_{\pi} \{\|\boldsymbol{\Delta}^{\boldsymbol{\pi}}\|_\infty\}, \tag{3.9}$$

where $\|x\|_\infty = \max\{| x_1 |, \dots, | x_n |\}$ is the maximum norm. We note that this objective ensures min-max fairness among users in the system, while playback delays are minimized.Before analyzing the minimum achievable system delay we define two additional terms $P(t) = \sum_1^N p_i(t)$ and $P^{\boldsymbol{\Delta}}(t) = \sum_1^N p_i(t - \Delta_i)$ to simplify our notations. The first one is interpreted as the *cumulative playback curve* and the second one is interpreted as the *cumulative playout curve*. The next lemma establishes the relationship between the $R(t)$ and $P^{\boldsymbol{\Delta}^{\boldsymbol{\pi}}}(t)$ for a rate splitting policy and the corresponding minimum playback delay vector $\boldsymbol{\Delta}^{\boldsymbol{\pi}}$. We remark that, for the sake of simplicity in the notation $P^{\boldsymbol{\Delta}}(t)$ is used instead of $P^{\boldsymbol{\Delta}^{\boldsymbol{\pi}}}(t)$.

Lemma 1. For any rate splitting policy π and corresponding minimum initial delay vector $\boldsymbol{\Delta}^{\boldsymbol{\pi}}$, the following inequality between the cumulative rate curve $R(t)$ and the cumulative playout curve $P^{\boldsymbol{\Delta}}(t) = \sum_1^N p_i(t - \Delta_i^\pi)$, holds for all $t \geq 0$.

$$R(t) \geq P^{\boldsymbol{\Delta}}(t), \tag{3.10}$$

Proof: For a given rate splitting policy π and the delay vector $\boldsymbol{\Delta}^{\boldsymbol{\pi}}$, the inequality (3.6) holds for all $i = 1, \dots, N$ and for all t. Summing over all of these inequalities gives the following inequality

$$\sum_{i=1}^{N} R_i(t) \geq \sum_{i=1}^{N} p_i(t - \Delta_i^\pi), \tag{3.11}$$

for all t. By using (3.2) and the definition of $P^{\boldsymbol{\Delta}}(t)$, we observe that the summation in the left hand side is equal to the cumulative rate curve $R(t)$ and the summation in the right hand side is equal to the cumulative playout curve $P^{\boldsymbol{\Delta}}(t)$.

An important corollary of Lemma 1 is that inequality (3.10) is a necessary condition for providing jitter-free video streaming to all users. We now utilize Lemma 1 to obtain a lower bound on the minimum system delay attainable by solution (3.9). Later, we will show that this lower bound is achievable by constructing a feasible rate splitting policy attaining the lower bound and ensuring jitter-free video service to all users in the system. To this end, let $\Delta^\star$ be equal to

$$\Delta^\star = \max_t \left\{R^{-1}\left(P(t)\right) - t\right\}. \tag{3.12}$$

Hence, $\Delta^\star$ can be considered as the minimum playback delay in the single user case with the playback curve $P(t)$ and receiver curve $R(t)$. We note that $\Delta^\star$ is only a function of $P(t)$ and $R(t)$, and therefore it is fixed number for any given $P(t)$ and $R(t)$.

Lemma 2. For any minimum initial delay vector $\boldsymbol{\Delta^\pi}$, the system delay Δ_s satisfies the inequality,

$$\Delta^\star \leq \Delta_s. \tag{3.13}$$

Proof: Assume there exist a rate splitting policy π with a corresponding $\boldsymbol{\Delta^\pi}$ such that $S(\boldsymbol{\Delta^\pi}) < \Delta^\star$. Then, Lemma 1 implies that the cumulative rate curve and the cumulative playout curve satisfy (3.10), which leads to

$$R(t) \geq P^{\boldsymbol{\Delta}}(t) = \sum_{i=1}^{N} p_i(t - \Delta_i^\pi),$$

for all $t > 0$. From the definition of the system delay, $\Delta_i^\pi \leq \Delta_s$ for all i. Further, since $\Delta_s < \Delta^\star$, we can also conclude that $\Delta_i^\pi < \Delta^\star$ for all i. Then, using the monotonically increasing property of the playback curve, we have

$$R(t) \geq \sum_{i=1}^{N} p_i(t - \Delta_s) = P(t - \Delta_s),$$

for all $t > 0$. Hence, $\Delta_s < \Delta^\star$ and satisfies $R(t) \geq P(t - \Delta)$ for all $t > 0$, which contradicts with the fact that $\Delta^\star$ is the minimum Δ value that satisfies $R(t) \geq P(t - \Delta)$ for all $t > 0$.

Now we search for the rate splitting policy π that provides the minimum system delay Δ_s while all users are provided with jitter-free video streaming. Recall that in the proposed system model, users request the corresponding video stream $p_i(t)$ at $t = 0$ simultaneously. Assume we have playback delay vector $\boldsymbol{\Delta}$. Let there be K increment points in the corresponding cumulative playout curve $P^{\boldsymbol{\Delta}}(t)$. Each increment point is denoted by t_k where $k \in \{1, \dots, K\}$. Given that Q_k is the total additional data demand between t_{k-1} and t_k, t_k can be expressed as

$$t_k = (P^{\boldsymbol{\Delta}})^{-1}\left(\sum_{j=1}^{k} Q_j\right). \tag{3.14}$$

For each increment point, we have a corresponding *sufficiency point* τ_k such that τ_k is the video streaming server time instant at which the video streaming server is able provision the total additional data demand required by the users at the increment point t_k. Equivalently, it is the first time instant at which the cumulative rate curve reaches the value of $\sum_{j=1}^{k} Q_j$.

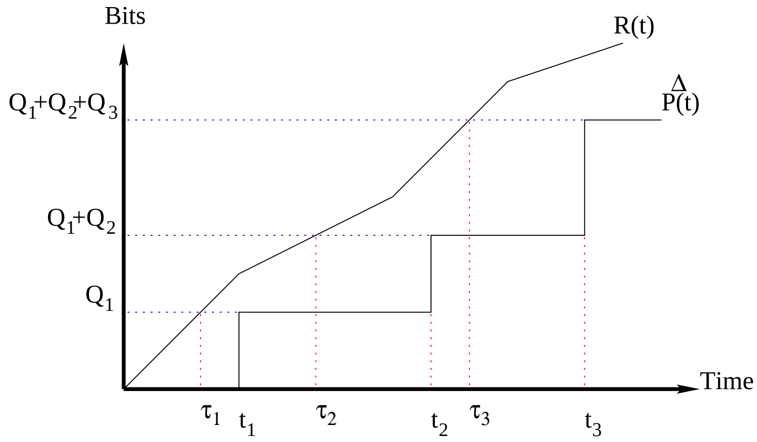


FIGURE 3.3: Cumulative Rate Curve and Cumulative Playout Curve

---

Algorithm 1

Input: $\mathbf{\Delta}$, $R(t)$, $\{p_i(t)\}_{i=1}^N$
Output: $\{R_i(\tau_k)\}_{i=1}^N$
if $k = 1$ **then**
  $Q_1 = P^{\mathbf{\Delta}}(t_1)$
else
  $Q_k = P^{\mathbf{\Delta}}(t_k) - P^{\mathbf{\Delta}}(t_{k-1})$
end if
for $k = 1 : K$ **do**
  $\tau_k = R^{-1}\left(\sum_1^k Q_j\right)$
  if $k = 1$ **then**
    $q_i^1 = p_i^{\Delta}(t_1)$,
    $R_i(\tau_1) = q_i^1$,
  else
    $q_i^k = p_i^{\Delta}(t_k) - p_i^{\Delta}(t_{k-1})$,
    $R_i(\tau_k) = R_i(\tau_{k-1}) + q_i^k$,
  end if
end for

We propose Algorithm 1 to address the rate allocation problem for jitter-free multiuser video streaming. Algorithm works as follows. For any given $\mathbf{\Delta}$, $R(t)$ and $\{p_i(t)\}_{i=1}^N$, it first determines the all sufficiency points. Then, in between each sufficiency point τ_{k-1} and τ_k, Algorithm allocates $p_i^{\Delta}(t_k) - p_i^{\Delta}(t_{k-1})$ bits to user i. We note that $p_i^{\Delta}(t_k) - p_i^{\Delta}(t_{k-1})$ is the share of the user i in the increment of the cumulative playouut curve at t_k.

Algorithm 1, in turn, utilizes the knowledge of $p_i^{\Delta}(t_k) - p_i^{\Delta}(t_{k-1})$ for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$ to determine the values of receiver curve $R_i(t)$ at τ_k. for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$. Note that Algorithm 1 does not construct $R_i(t)$ for all t. It only provides the critical values $\{R_i(\tau_k), k = 1, \ldots, K\}_{i=1}^N$ that all receiver curves need to satisfy for jitter-free video streaming. Due to the step function property of the playback curves, any rate splitting policy achieving these critical values at τ_k, $k = 1, \ldots, K$ provides jitter-free video streaming service to all users. Thus, provides a family of rate splitting policies that achieve critical rate values at the sufficiency points.

This family of rate splitting policies is not necessarily singleton, and we will show below that any rate splitting policy in this family provides jitter-free streaming service to all users in the system. Let $\{R_i(\tau_k), k = 1, \ldots, K\}_{i=1}^{N}$ be the output of Algorithm 1 for given $R(t)$, $\{p_i(t)\}_{i=1}^{N}$ and $\mathbf{\Delta}$. Then, all rate splitting policies with the receiver curves achieving the critical values $\{R_i(\tau_k), k = 1, \ldots, K\}_{i=1}^{N}$ form a set of rate splitting policies that is denoted by $\Pi^{\mathbf{\Delta}}$.

Lemma 3. For given delay vector $\mathbf{\Delta_1} = (\Delta_1, \ldots, \Delta_N)$ and $\mathbf{\Delta_2} = (\Delta_1 + \Delta_d, \ldots, \Delta_N + \Delta_d)$ with any $\Delta_d > 0$, Algorithm 1 gives the same set of rate splitting policies.

Proof: If we shift the all playout curves by Δ_d, then cumulative playout curve is shifted by same amount and the shape of the curve is preserved. Therefore, although the increment points t_k changes, values of the Q_i, q_i^k and τ_k remains the same. Consequently we have the same critical values $R_i(\tau_k)$ for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$. As a result, we have the same set of rate splitting policies achieving these critical values.

Note that if the necessary condition (3.10) in Lemma 1 is satisfied, then we have to have $t_k \geq \tau_k$ for all k due to monotonically increasing property of $R(t)$ and $P^{\mathbf{\Delta}}(t)$. Now, we will use this observation to obtain a family of jitter-free rate splitting policies in the next Lemma.

Lemma 4. For given $R(t)$, $\{p_i(t)\}_{i=1}^{N}$ and $\mathbf{\Delta}$ satisfying (3.10), any rate splitting policy $\pi \in \Pi^{\mathbf{\Delta}}$ provides jitter-free video streaming to all users.

Proof: By construction, for any $\pi \in \Pi^{\mathbf{\Delta}}$ we have

$$R_i(t_k) \geq R_i(\tau_k) = p_i(t_k - \Delta_i), \tag{3.15}$$

for all increment points $t_1, \ldots, t_K$. Since $R_i(t)$ is monotonically increasing and $p_i(t_k - \Delta_i)$ is constant over time intervals $\{[t_{k-1}, t_k)\}_{k=1}^{K}$, we conclude then $R_i(t) \geq p_i(t)$ for all t. Hence, all users in the system are provided with a jitter-free video streaming service under the rate splitting policy π.

Recall that for each rate splitting policy π, we have a corresponding minimum playback delay vector $\mathbf{\Delta^\pi}$ that consists of minimum single playback delays for the given rate splitting policy π. We search for the rate splitting policy $\pi^\star$ providing jitter-free video streaming to all users with the minimum system delay, i.e., $\pi^\star$ belongs to the set

$$\underset{\pi}{\operatorname{argmin}} \left\{ \|\mathbf{\Delta^\pi}\|_\infty \right\}.$$

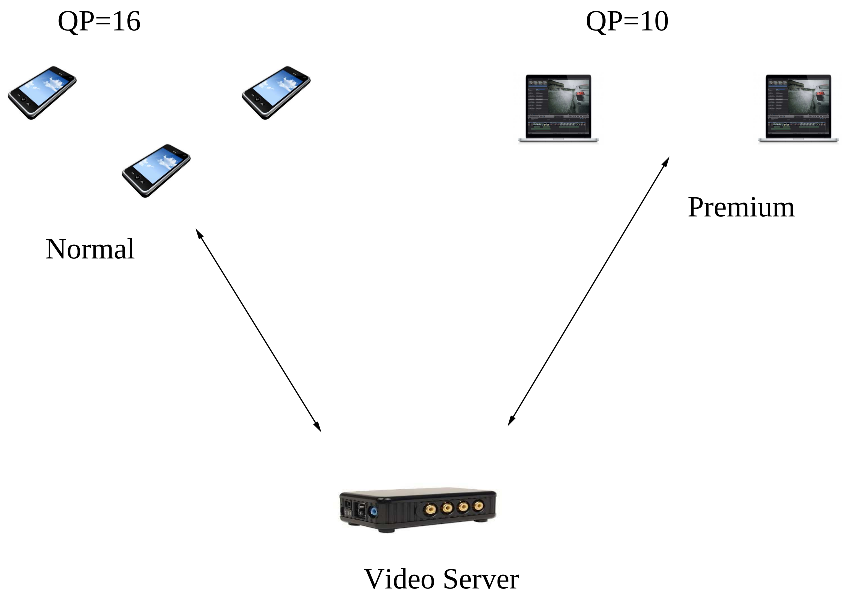


FIGURE 3.4: Experiment Setup

The following theorem indicates that $\Delta^\star = \max\left\{R^{-1}\left(P(t)\right) - t\right\}$ is the minimum achievable system delay. Moreover, any rate splitting policy $\pi \in \Pi^{\mathbf{0}}$ achieves the minimum system delay $\Delta^\star$ where $\mathbf{0} = (0, \ldots, 0)$.

Theorem 1. For given $R(t)$ and $\{p_i(t)\}_{i=1}^N$, any rate splitting policy $\pi \in \Pi^{\mathbf{0}}$ is a solution to minimization problem (3.9).

Proof: Consider the playback delay vector $\mathbf{\Delta}^\star$ that is given as $\{\Delta^\star, \ldots, \Delta^\star\}$. Notice that $\mathbf{\Delta}^\star$ satisfies (3.10) for all t. Hence, for any rate splitting policy $\pi \in \Pi^{\mathbf{\Delta}^\star}$ and the playback delay vector $\mathbf{\Delta}^\star$, the jitter-free video streaming service is provided to all users by Lemma 4. Therefore, we have the following inequality,

$$\|\mathbf{\Delta}^{\boldsymbol{\pi}}\|_\infty \leq \Delta^\star \tag{3.16}$$

for any $\pi \in \Pi^{\mathbf{\Delta}^\star}$. On the other hand, Lemma 2 implies that $\Delta^\star \leq \|\mathbf{\Delta}^{\boldsymbol{\pi}}\|_\infty$. Hence, $\Delta_s = \Delta^\star$ for the rate splitting policy $\pi \in \Pi^{\mathbf{\Delta}^\star}$ and the lower bound on the system delays with jitter free streaming in (3.13) is achieved by any $\pi \in \Pi^{\mathbf{\Delta}^\star}$. Therefore, π is solution to the minimization problem in (3.9). Since $\Pi^{\mathbf{0}}$ and $\Pi^{\mathbf{\Delta}^\star}$ corresponds to same set by Lemma 3, we conclude that any rate splitting policy $\pi \in \Pi^{\mathbf{0}}$ achieves the system delay $\Delta_s = \Delta^\star$ and be a solution to given minimization problem in (3.9).

3.2.2 Numerical Analysis

We perform numerical experiments to investigate the relation between the number of users and the channel capacity for the given QoS level (i.e., playback delay). The

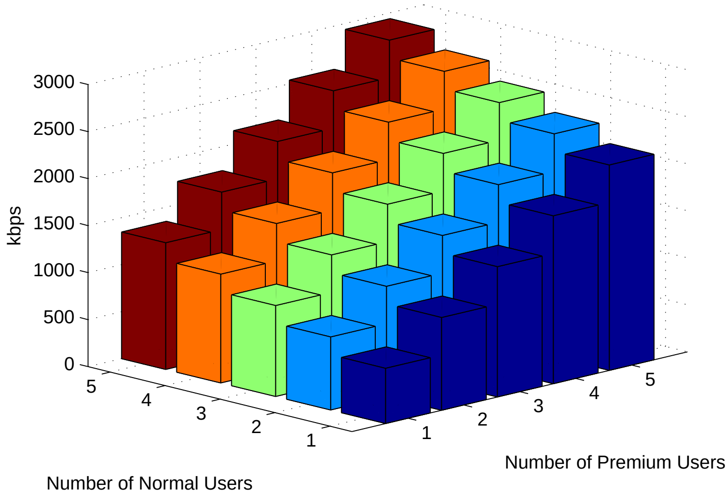


FIGURE 3.5: Required Channel Capacity for Playback Delay of 30 Seconds

video file used in the experiments are encoded according to single layer H.264/AVC standard [69]. We consider approximately 30 minutes of *Silence of Lambs* with CIF resolution(352x288) at a 30 frames per second. A GOP size of 16 and frame configuration of 3 frames in between I/P frames key pictures is used [69]. In the simulation setup, we consider two different user classes, users with normal and premium service. Users from respectively different classes request the same video content with different quantization parameters (QP). A quantization parameter is used to determine the quantization level of transform coefficients in H.264/AVC. An increase of 1 unit in the quantization parameter means an increase of quantization step size by approximately 12 percent which in turns means 12 percent reduction in the video rate [70]. We assume that users with normal service, stream the video file with $QP = 16$ and users with premium service stream the video file with $QP = 10$. In the experiment, we observe the necessary capacity for satisfying different playback delay requirements. Fig. 3.5 and Fig. 3.6 show the results for playback delay of 30 seconds and 60 seconds respectively. Our results indicate that the required channel capacity increases linearly with increasing number of users for a given QoS level. However when the number of users is kept constant, increasing the playback delay does not decrease the required channel capacity linearly.

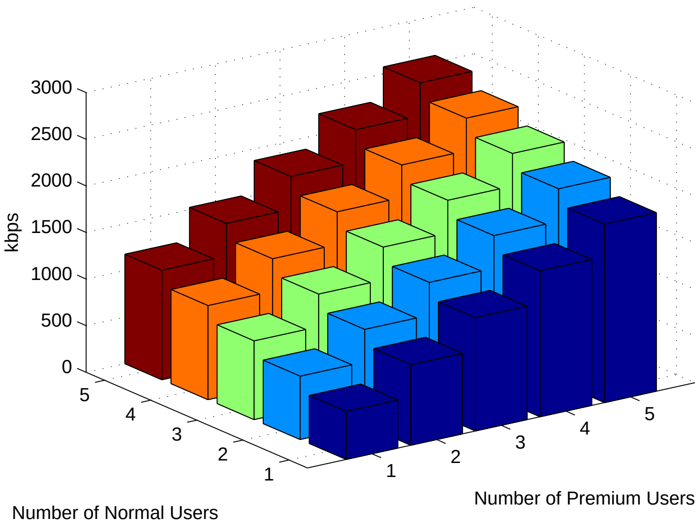


FIGURE 3.6: Required Channel Capacity for Playback Delay of 60 Seconds

3.3 Multiuser Video Streaming for Asynchronized Video Requests

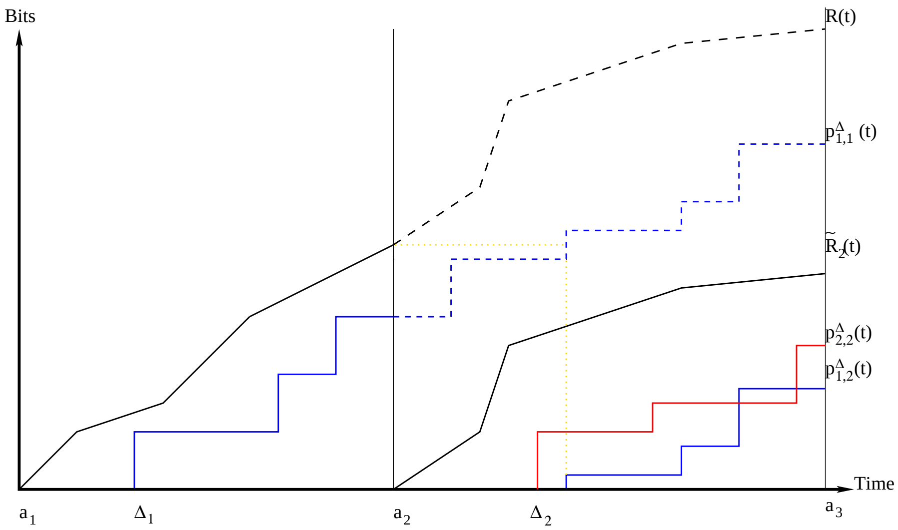


FIGURE 3.7: Example case with 3 Users

In a general multiuser video streaming system, users are not obliged to request the corresponding video files synchronously. In other words, different from the case examined in the previous section, video requests may not reach the server at the same time particularly at $t = 0$. We consider the system for N users where they consecutively request different video streams from the server. Let a_i be the request time of video stream belonging to user i and without loss of generality let $a_1 = 0$. For ease of notation we assume that users are indexed according to their order of arrival to the system. The server starts to allocate rate to a user, when a request for the video stream has arrived. This implies that for all i, $R_i(t) = 0$ and $p_i(t) = 0$ when $t < a_i$. We first explain the system with an example with three users and then generalize it to system with N users. Eventually, we want to show that Algorithm 1 ,introduced in the previous section, achieves the minimum system playback delay for asynchronised video requests as well as synchronized video requests.

In this system model, it is difficult to find minimum achievable system delay immediately, due to newly arriving users. Therefore, to simplify our analysis we consider each mutually exclusive time interval (*frame*) $(a_i, a_{i+1}]$ as an independent rate allocation problem such that each frame corresponds to a system with a constant number of users simultaneously requesting video streams. Hence, to examine the each frame separately without causing a sub-optimality and to sustain the independency, we develop the notion of residual playout curve and residual cumulative rate curve. The *residual playout curve* $p^{\Delta}_{i,j}(t)$ is interpreted as the playout curve of the user i and the *residual cumulative rate curve* $\tilde{R}_j(t)$ is interpreted as the cumulative rate curve when the rate allocation process is completed in the time interval $(a_{j-1}, a_j]$. In each frame $(a_j, a_{j+1}]$, we divide the cumulative rate between first arriving j users and construct $R_{i,j}(t)$ as the *residual receiver curve* for user i. A simple case with 3 users is visualized in Fig.3.7 . Observe that between a_1 and a_2, server allocates all the rate to user 1 i.e., $R_{1,1}(a_2) = \tilde{R}_1(a_2) = R(a_2)$. For $t > a_2$ the residual playout curve of user 1 is $p^{\Delta}_{1,2}(t) = \left[p^{\Delta}_{1,1}(t) - R_{1,1}(a_2)\right]^+$, where $[x]^+ = \max(x, 0)$ and the residual cumulative rate curve is $\tilde{R}_2(t) = R(t) - R(a_2)$. At $t = a_2$ user 2 joins the system with the playout curve $p^{\Delta}_2(t)$. Hence, during the time interval $(a_2, a_3]$ the server splits the cumulative rate between user 1 and 2 such that $R_{1,2}(a_3) + R_{2,2}(a_3) = \tilde{R}_2(a_3)$. Similarly, we determine the residual playout curves $p^{\Delta}_{1,3}(t)$, $p^{\Delta}_{2,3}(t)$ and residual cumulative rate curve $\tilde{R}_3(t)$ for $t > a_3$. This process can be generalized for arbitrary $N > 1$. In general the residual cumulative rate curve is given as

$$\tilde{R}_j(t) = R(t) - R(a_j) \tag{3.17}$$

for $t > a_j$ and residual playout curve is given as

$$p^{\Delta}_{i,j}(t) = \left[p^{\Delta}_{i,j-1}(t) - R_{i,j-1}(a_j)\right]^+ \tag{3.18}$$

for $t > a_j$ and for all i. We remark that initially $p^{\Delta}_{i,j}(t) = p^{\Delta}_i(t)$ for $t > a_i$. In addition, receiver curve of the user i in the time interval $(a_j, a_{j+1}]$ is given as

$$R_i(t) = R_i(a_j) + R_{i,j}(t) \tag{3.19}$$

and $R_i(t) = 0$ for $t \leq a_i$, since user i has not arrived yet. Note that $p^{\Delta}_{i,j}(t)$ for $t > a_j$ is recursively updated in the following way,

$$p^{\Delta}_{i,j}(t) = \left[\left[\left[p^{\Delta}_i(t) - R_{i,i}(a_{i+1})\right]^+ - R_{i,i+1}(a_{i+2})\right]^+ \ldots - R_{i,j-1}(a_j)\right]^+ . \tag{3.20}$$

The expression in the right hand side of the (3.20) is equivalent to

$$\left[p^{\Delta}_i(t) - \sum_{k=i}^{j-1} R_{i,k}(a_{k+1})\right]^+ . \tag{3.21}$$

Hence, using (3.19) $p^{\Delta}_{i,j}(t)$ simplified in to

$$p^{\Delta}_{i,j}(t) = \left[p^{\Delta}_i(t) - R_i(a_j)\right]^+ . \tag{3.22}$$

In the previous section, we have a single condition to ensure jitter-free video streaming. However, in this section we aim to investigate the jitter event for each frame separately. Hence, we need to convert the given condition for jitter-free streaming in to multiple conditions such that each corresponds to a particular frame $(a_j, a_{j+1}]$.

3.3.1 Preliminaries

In the following lemma, utilizing the notion of residual playout curve and residual receiver curve, we redefine the sufficient condition for jitter-free video streaming.

Lemma 5. System provides jitter-free video streaming to all users i.e.,

$$R_i(t) \geq p^{\Delta}_i(t)$$

holds for all i and all t, if and only if the following inequality

$$R_{i,j}(t) \geq p^{\Delta}_{i,j}(t) \tag{3.23}$$

is satisfied for any frame $(a_j, a_{j+1}]$ and for all i.

Proof: First assume that (3.23) holds for all i and for each frame $(a_j, a_{j+1}]$. Then, we need to show that

$$R_i(t) \geq p^{\Delta}_i(t)$$

holds for all i and all t. Without loss of generality consider the particular frame $(a_j, a_{j+1}]$. Then, using (3.22) we rewrite (3.23) in the following way

$$R_{i,j}(t) \geq \left[p^{\Delta}_i(t) - R_i(a_j)\right]^+ , \tag{3.24}$$

where $a_j < t \leq a_{j+1}$. Further, we add $R_i(a_j)$ to both sides of (3.24) such that

$$R_{i,j}(t) + R_i(a_j) \geq \left[p_i^{\Delta}(t) - R_i(a_j)\right]^+ + R_i(a_j). \tag{3.25}$$

The term in the left hand side of (3.25) is equal to $R_i(t)$ from (3.19), thus we have the following inequality

$$R_i(t) \geq \left[p_i^{\Delta}(t) - R_i(a_j)\right]^+ + R_i(a_j), \tag{3.26}$$

for $a_j < t \leq a_{j+1}$ and for all i. We note that $\left[p_i^{\Delta}(t) - R_i(a_j)\right]^+ + R_i(a_j) \geq p_i^{\Delta}(t)$. Hence, we show that

$$R_i(t) \geq p_i^{\Delta}(t) \tag{3.27}$$

holds for all i. This inequality implies that the system provides jitter-free video streaming to all users. Now assume that system provides jitter-free video streaming to all users i.e.,

$$R_i(t) \geq p_i^{\Delta}(t)$$

holds for all i and for all t. Then, if we extract $R_i(a_j)$ from both side of the equation in the above we obtain the following inequality

$$R_{i,j}(t) \geq p_i^{\Delta}(t) - R_i(a_j), \tag{3.28}$$

for $a_j < t \leq a_{j+1}$ and for all i. Notice that $R_{i,j}(t) \geq 0$ for $t > a_j$, thus we can replace right hand side of (3.28) with $\left[p_i^{\Delta}(t) - R_i(a_j)\right]^+$. Finally, from (3.22) we conclude that

$$R_{i,j}(t) \geq p_{i,j}^{\Delta}(t)$$

holds for each frame $(a_j, a_{j+1}]$ and for all i.

3.3.2 Delay Optimal Rate Splitting Policy

In this subsection, we search for the rate splitting policy π that provides the minimum system delay Δ_s while all users are provided with jitter-free video streaming i.e.,

$$\underset{\pi}{\operatorname{argmin}} \left\{\|\boldsymbol{\Delta}^{\boldsymbol{\pi}}\|_{\infty}\right\}.$$

Before searching for the delay optimal rate splitting policy π, recall that for a given rate splitting policy π and cumulative rate curve $R(t)$ minimum playback delay for user i with $p_i(t)$ is given as

$$\Delta_i^{\pi} = \max_t \left\{(R_i)^{-1}\left(p_i(t)\right) - t\right\}. \tag{3.29}$$

In addition, playback delay vector $\boldsymbol{\Delta}^{\boldsymbol{\pi}} = (\Delta_1^{\pi}, \ldots, \Delta_N^{\pi})$ is defined as minimum playback delay vector for the corresponding rate splitting policy π.

Recall that Algorithm 1 provides the critical values $\{R_i(\tau_k), k = 1, \ldots, K\}_{i=1}^{N}$ for given $R(t)$, $\{p_i(t)\}_{i=1}^{N}$ and $\boldsymbol{\Delta}$. However, in the case of asynchronised video requests each user have a different arrival time a_i and it is assumed that $R_i(t) = 0$ for $t \leq a_i$ in other words server starts to allocate rate to user i at a_i. Thus, algorithm 1 can not be directly executed for given $R(t)$, $\{p_i(t)\}_{i=1}^{N}$ and $\boldsymbol{\Delta}$. As a result of that, for given $R(t)$, $\{p_i(t)\}_{i=1}^{N}$, $\boldsymbol{\Delta}$ and $\{a_i\}_{i=1}^{N}$ we execute the Algorithm 1 for each exclusive time interval $(a_j, a_{j+1}]$ separately. Notice that if we are dealing with the frame $(a_j, a_{j+1}]$, then $\left\{p_i^{\Delta}(t)\right\}_{i=1}^{j}$ and $\tilde{R}_j(t)$ will be the inputs of the Algorithm 1 and as an output we obtain the sufficiency points τ_k in the interval $(a_j, a_{j+1}]$. Collection of these sufficiency points τ_k, defines the critical values for receiver curves. Hence, we have $\{R_i(\tau_k), k = 1, \ldots, K\}_{i=1}^{N}$ as an overall output. Then, all the rate splitting policies with the receiver curves achieving the the critical values $\{R_i(\tau_k), k = 1, \ldots, K\}_{i=1}^{N}$ form a set of rate splitting policies that is denoted by Π^{Δ}.

Note that difference between the asynchronised case and synchronized case, in terms of the outputs provided by Algorithm 1, is the location of the sufficiency points. In spite of that values of the receiver curves at the sufficiency points are same for the both cases. In this section, we further introduce the term $P_j^{\boldsymbol{\Delta}}(t)$ such that

$$P_j^{\boldsymbol{\Delta}}(t) = \sum_{i=1}^{j} p_{i,j}^{\Delta}(t). \tag{3.30}$$

$P_j^{\boldsymbol{\Delta}}(t)$ is interpreted as *residual cumulative playout curve* for $t > a_j$. The next corollary establishes the relationship between the $\tilde{R}_j(t)$ and $P_j^{\boldsymbol{\Delta}}(t)$ for a given rate splitting policy π and the corresponding minimum delay vector $\boldsymbol{\Delta}^{\boldsymbol{\pi}}$

Corollary 1. For any rate splitting policy π and the corresponding minimum playback delay vector $\boldsymbol{\Delta}^{\boldsymbol{\pi}}$, residual cumulative rate curve $\tilde{R}_j(t)$ should be over the residual cumulative playout curve $P_j^{\boldsymbol{\Delta}}(t) = \sum_{i=1}^{j} p_{i,j}^{\Delta}(t)$ i.e.,

$$\tilde{R}_j(t) \geq P_j^{\boldsymbol{\Delta}}(t). \tag{3.31}$$

for each frame $(a_j, a_{j+1}]$.

Proof: From definition of the $\boldsymbol{\Delta}^{\boldsymbol{\pi}}$ and Lemma 5, we know that

$$R_{i,j}(t) \geq p_{i,j}^{\Delta}(t) \tag{3.32}$$

for each frame $(a_j, a_{j+1}]$ and for all i. Summing over all of the inequalities $i = 1 : j$ gives the following inequality

$$\tilde{R}_j(t) \geq P_j^{\boldsymbol{\Delta}}(t),$$

for $a_j < t \leq a_{j+1}$.

Notice that given corollary defines the necessary condition for jitter-free video streaming. In the remaining part of this subsection, we first introduce the properties of the rate splitting policies that are constructed according to Algorithm 1. Then, we show that there exist a lower bound for the system delay Δ_s. Finally, we prove that any rate splitting policy $\pi \in \Pi^{\mathbf{0}}$ is a delay optimal rate splitting policy such that any rate splitting policy $\pi \in \Pi^{\mathbf{0}}$ is a solution to minimization problem in (3.9). In the next Lemma, we underline the relationship between the rate splitting policy π and the residual cumulative playout curve $P_j^{\boldsymbol{\Delta}}(t)$. Lemma 6 asserts that for given $R(t)$, $\left\{p_i(t)_{i=1}^N\right\}$ and $\boldsymbol{\Delta}$ any rate splitting policy $\pi \in \Pi^{\boldsymbol{\Delta}}$ achieves the minimum residual cumulative playout curve $P_j^{\Delta}(t)$ for each frame $(a_j, a_{j+1}]$.

Lemma 6. For given $R(t)$, $\left\{p_i(t)_{i=1}^N\right\}$ and $\boldsymbol{\Delta}$ the following inequality

$$P_{j+1}^{\boldsymbol{\Delta}}(t) \geq \left[P_j^{\boldsymbol{\Delta}}(t) - \tilde{R}_j(a_{j+1})\right]^+ + p_{j+1}^{\Delta}(t), \tag{3.33}$$

holds for any rate splitting policy π and equality is achieved for any $\pi \in \Pi^{\boldsymbol{\Delta}}$.

Proof: Assume that there are n users in the video streaming system. Notice that for the time frame $(a_n, a_{n+1}]$, video streaming system is equivalent to a video streaming system where users simultaneously request the video streams at $t = a_n$ with corresponding playout curves $p_{1,n}^{\Delta}(t), \ldots, p_{n,n}^{\Delta}(t)$. Further, we assume that a new user arrive to the system at $t = a_{n+1}$ with playout curve $p_{n+1}^{\Delta}(t)$. From (3.30) and (3.18), the residual cumulative playout curve $P_{n+1}^{\boldsymbol{\Delta}}(t)$ for $t > a_{n+1}$ is given as

$$P_{n+1}^{\boldsymbol{\Delta}}(t) = \sum_{i=1}^{n} \left[p_{i,n}^{\Delta}(t) - R_{i,n}(a_{n+1})\right]^+ + p_{n+1}^{\Delta}(t). \tag{3.34}$$

Note that value of the $R_{i,n}(a_{n+1})$ depend on the rate splitting policy π, however we know that $\sum_{i=1}^{n} R_{i,n}(a_{n+1}) = \tilde{R}_n(a_{n+1})$. Therefore, following inequality holds

$$\sum_{i=1}^{n} \left[p_{i,n}^{\Delta}(t) - R_{i,n}(a_{n+1})\right]^+ \geq \left[\sum_{i=n}^{N} p_{i,n}^{\Delta}(t) - \tilde{R}_n(a_{n+1})\right]^+, \tag{3.35}$$

for $t > a_{n+1}$. The term $\sum_{i=n}^{N} p^{\Delta}_{i,n}(t)$ in the right hand side of the (3.35) is replaced by $P^{\mathbf{\Delta}}_{n}(t)$ and we have the following inequality

$$\sum_{i=1}^{n} \left[p^{\Delta}_{i,n}(t) - R_{i,n}(a_{n+1})\right]^{+} \geq \left[P^{\mathbf{\Delta}}_{n}(t) - \tilde{R}_{n}(a_{n+1})\right]^{+}. \tag{3.36}$$

From (3.36) and (3.34) we conclude that,

$$P^{\mathbf{\Delta}}_{n+1}(t) \geq \left[P^{\mathbf{\Delta}}_{n}(t) - \tilde{R}_{n}(a_{n+1})\right]^{+} + p^{\Delta}_{n+1}(t). \tag{3.37}$$

Let assume cumulative rate is distributed among the users according to a rate splitting policy $\pi \in \Pi^{\mathbf{\Delta}}$. Further, assume that there are K increment point in the residual cumulative playout curve $P^{\mathbf{\Delta}}_{n}(t)$ and without loss of generality let $\tau_k \leq a_{n+1} \leq \tau_{k+1}$ for some k. Then, Algorithm 1 implies that following inequality

$$p^{\Delta}_{i,n}(t_k) \leq R_{i,n}(a_{n+1}) \leq p^{\Delta}_{i,n}(t_{k+1}) \tag{3.38}$$

should hold for all $i \leq n$. Likewise, we have $P^{\mathbf{\Delta}}_{n}(t_k) \leq \tilde{R}_{n}(a_{n+1}) \leq P^{\mathbf{\Delta}}_{n}(t_{k+1})$. Now, we want to show that inequality (3.36) is satisfied with equality for any $\pi \in \Pi^{\mathbf{\Delta}}$. Consider two possible cases where $t < t_{k+1}$ and $t \geq t_{k+1}$. When $t < t_{k+1}$, both sides of the (3.36) will be 0 since $p^{\Delta}_{i,n}(t) - R_{i,n}(a_{n+1}) \leq 0$ and $P^{\mathbf{\Delta}}_{n}(t) - \tilde{R}_{n}(a_{n+1}) \leq 0$ for $t < t_{k+1}$. On the other hand, when $t \geq t_{k+1}$

$$[p^{\Delta}_{i,n}(t) - R_{i,n}(a_{n+1})]^{+} = p^{\Delta}_{i,n}(t) - R_{i,n}(a_{n+1}). \tag{3.39}$$

Hence, we have the following equality

$$\sum_{i=1}^{n} \left[p^{\Delta}_{i,n}(t) - R_{i,n}(a_{n+1})\right]^{+} = \sum_{i=1}^{n} p^{\Delta}_{i,n}(t) - \sum_{i=1}^{n} R_{i,n}(a_{n+1}) = P^{\mathbf{\Delta}}_{n}(t) - \tilde{R}_{n}(a_{n+1}), \tag{3.40}$$

for $t > a_{n+1}$. Since we monitor that (3.36) is satisfied with equality, we conclude that (3.33) is also satisfied with equality for any rate splitting $\pi \in \Pi^{\mathbf{\Delta}}$.

Now, let us introduce the term $P_j(t)$, where $P_j(t)$ is the residual cumulative playout curve for the given playback delay vector $\mathbf{\Delta} = \{0, \ldots, 0\}$ and the rate splitting policy $\pi \in \Pi^{\mathbf{0}}$ i.e.,

$$P_j(t) = \left[P_{j-1}(t) - \tilde{R}_{j-1}(a_j)\right]^{+} + p_j(t). \tag{3.41}$$

In the following lemma we want to reveal the relation between the $P_j(t)$ and $P^{\mathbf{\Delta}}_{j}(t)$ for given $\mathbf{\Delta}$ and $\pi \in \Pi^{\mathbf{\Delta}}$.

Lemma 7. For given $R(t)$,$\{p_i(t)\}_{i=1}^N$ and $\mathbf{\Delta} = (\Delta, \dots, \Delta)$, any rate splitting policy $\pi \in \Pi^{\mathbf{\Delta}}$ provides the equality

$$P_j^{\mathbf{\Delta}}(t) = P_j(t - \Delta). \tag{3.42}$$

Proof: In the proof of lemma, we use the induction method. The base case $j = 1$ is trivial since $P_1^{\mathbf{\Delta}}(t) = p_1(t - \Delta)$ and $P_1(t) = p_1(t)$. Then, assume for j equality $P_j^{\mathbf{\Delta}}(t) = P_j(t - \Delta)$ holds. Now, we want show that equality (3.42) holds for $j + 1$. Recall, $P_{j+1}(t)$ is given as

$$\left[P_j(t) - \tilde{R}_j(a_{j+1})\right]^+ + p_{j+1}(t). \tag{3.43}$$

Notice that under the assumption of $P_j(t) - \tilde{R}_j(a_{j+1}) \neq 0$, (3.43) converted in to following expression

$$\left[P_j(t) + p_{j+1}(t) - \tilde{R}_j(a_{j+1})\right]^+. \tag{3.44}$$

$P_j(t) - \tilde{R}_j(a_{j+1}) \neq 0$ is a valid assumption for our system model. Because if it is not the case and $P_j(t) - \tilde{R}_j(a_{j+1}) \leq 0$, then it simply means that video streaming process for the first j user is completed before the arrival of the $j + 1$th user. However, this case is not in the scope of our interest. In a similar fashion $P_{j+1}^{\mathbf{\Delta}}(t)$ is given as

$$\left[P_j^{\mathbf{\Delta}}(t) + p_{j+1}^{\Delta}(t) - \tilde{R}_j(a_{j+1})\right]^+. \tag{3.45}$$

Since, we assume that $P_j^{\mathbf{\Delta}}(t) = P_j(t - \Delta)$ holds and $p_{j+1}^{\Delta}(t) = p_{j+1}(t - \Delta)$ from the definition, we conclude that $P_{j+1}^{\mathbf{\Delta}}(t) = P_{j+1}(t - \Delta)$.

Until now, we examine the properties of the rate splitting policies that are provided by Algorithm 1. We now utilize Lemma 6, Lemma 7 and corollary to obtain a lower bound on the minimum system delay. Later , we will show that this lower bound is achievable by constructing a feasible rate splitting policy attaining the lower bound and ensuring jitter-free video streaming to all users in the system. To this and, let $\Delta_{\max}$ be equal to $\Delta_{\max} = \max\left\{\Delta_j^\star\right\}$ where

$$\Delta_j^\star = \max_t \left\{\tilde{R}_j^{-1}\left(P_j(t)\right) - t\right\}. \tag{3.46}$$

Lemma 8. For given any minimum playback delay vector $\mathbf{\Delta}^{\boldsymbol{\pi}}$, the system delay Δ_s satisfy the inequality

$$\Delta_{\max} \leq \Delta_s. \tag{3.47}$$

Proof: Assume that there exist a rate splitting policy π with a corresponding $\mathbf{\Delta}^{\boldsymbol{\pi}}$ such that $\{\|\mathbf{\Delta}^{\boldsymbol{\pi}}\|_\infty\} < \Delta_{\max}$. From corollary we know that for given π and $\mathbf{\Delta}^{\boldsymbol{\pi}}$ the following inequality $\tilde{R}_j(t) \geq P_j^{\boldsymbol{\Delta}}(t)$ holds for each frame $(a_j, a_{j+1}]$. Due to monotonically increasing property of playout curves the same inequality should be also valid for the playback delay vector $\boldsymbol{\Delta} = (\Delta_s, \ldots, \Delta_s)$ where each user is equally delayed with Δ_s. From Lemma 6 we know that for given playback delay vector $\boldsymbol{\Delta}$ any rate splitting policy $\pi \in \Pi^{\boldsymbol{\Delta}}$ provides the minimum residual cumulative playout vector $P_j^{\boldsymbol{\Delta}}(t)$. Hence, for each frame $(a_j < t \leq a_{j+1}]$, the inequality $\tilde{R}_j(t) \geq P_j^{\boldsymbol{\Delta}}(t)$ should hold for playback delay vector $\boldsymbol{\Delta} = (\Delta_s, \ldots, \Delta_s)$ and for any rate splitting policy $\pi \in \Pi^{\boldsymbol{\Delta}}$. From lemma 7, $P_j^{\boldsymbol{\Delta}}(t)$ can be replaced with $P_j(t - \Delta_s)$. Notice that $\Delta_j^\star$ is the minimum playback delay satisfying the inequality $\tilde{R}_j(t) \geq P_j(t - \Delta)$. Further, we know that there exist at least one j satisfying $\Delta_j^\star = \Delta_{\max}$. On the other hand, it contradicts with the observation $\tilde{R}_j(t) \geq P_j(t - \Delta_s)$ holds for each frame where $\Delta_s < \Delta_{\max}$

Theorem 2. For given $R(t)$ and $\{p_i(t)\}_{i=1}^N$, any rate splitting policy $\pi \in \Pi^{\mathbf{0}}$ is a solution to minimization problem (3.9) with a corresponding system delay $\Delta_{\max}$.

Proof: Consider the playback delay vector $\boldsymbol{\Delta}_{\mathbf{max}} = (\Delta_{\max}, \ldots, \Delta_{\max})$. Notice that for given playback delay vector $\boldsymbol{\Delta}_{\mathbf{max}}$ and any rate splitting policy $\pi \in \Pi^{\boldsymbol{\Delta}_{\mathbf{max}}}$ we have the following equality $P_j^{\boldsymbol{\Delta}_{\mathbf{max}}}(t) = P_j(t - \Delta_{\max})$ for all j from Lemma 6. Further, from definition of the $\Delta_{\max}$ it is ensured that $\tilde{R}_j(t) \geq P_j(t - \Delta_{\max})$. Hence, we have

$$\tilde{R}_j(t) \geq P_j^{\boldsymbol{\Delta}_{\mathbf{max}}}(t). \tag{3.48}$$

for all j. Then Lemma 4 implies that for any $\pi \in \Pi^{\boldsymbol{\Delta}_{\mathbf{max}}}$ following inequality

$$R_{i,j}(t) \geq p_{i,j}^{\Delta}(t) \tag{3.49}$$

holds for all i and all j. According to Lemma 5 the last inequality is the sufficient condition for providing jitter-free video streaming to all users. Hence, we conclude that

$$\{\|\boldsymbol{\Delta}^{\boldsymbol{\pi}}\|_\infty\} \leq \Delta_{\max} \tag{3.50}$$

for any $\pi \in \Pi^{\boldsymbol{\Delta}_{\mathbf{max}}}$. On the other hand Lemma 8 implies that $\Delta_{\max} \leq \{\|\boldsymbol{\Delta}^{\boldsymbol{\pi}}\|_\infty\}$. Thus, $\Delta_s = \Delta_{\max}$ for any rate splitting policy $\pi \in \Pi^{\boldsymbol{\Delta}_{\mathbf{max}}}$, and the lower bound on system delays with jitter free streaming in (3.47) is achieved by any $\pi \in \Pi^{\boldsymbol{\Delta}_{\mathbf{max}}}$. Therefore, π is a solution to the minimization problem in (3.9). Since $\Pi^{\boldsymbol{\Delta}_{\mathbf{max}}}$ and $\Pi^{\mathbf{0}}$ corresponds to the same set by Lemma 3, we conclude that any rate splitting policy $\pi \in \Pi^{\mathbf{0}}$ achieves the minimum system delay $\Delta_{\max}$ and is a solution to the minimization problem.

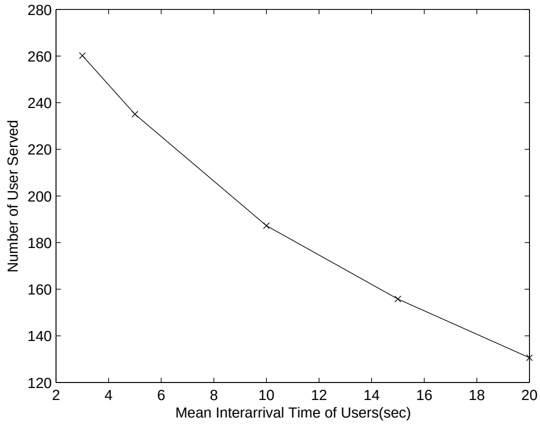
3.3.3 Numerical Results

In this subsection we perform numerical experiments to investigate the relation between the user acceptance rate, mean inter-arrival time of user and the channel capacity for the given QoS level (i.e., playback delay). The video file used in the experiments are encoded according to single layer H.264/AVC standard [69]. We consider 3 different video fragments from different video files *Silence of Lambs*,*Tokyo Olympics* and *Stars Wars* with CIF resolution(352x288) at a 30 frames per second. Each of the video fragments has a duration of 60 seconds. A GOP size of 16 and frame configuration of 3 frames in between I/P frames key pictures is used [69] and quantization parameters (QP) is set to 24. In the simulation setup, we consider two different channel rate, 1Mb/s and 2Mb/s. We assume that arrival process of the users behave according to Poisson distribution, thus the time interval between the two consecutive arrival is exponentially distributed.

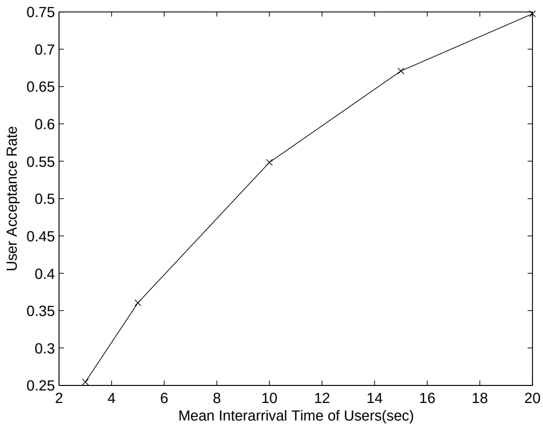
$$f(x;\ \lambda) = \begin{cases} \frac{1}{\lambda}\exp^{\frac{-x}{\lambda}} & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases}$$

Further, it is assumed that new arriving user requests one of the video fragments randomly. We consider that request frequency for each video fragment is equal. If the system is able to provide jitter-free video streaming to all users, then the new user admitted to system. For each channel rate we observe the number of jitter-free video streaming service completed in a 1 hour period and monitor the user acceptance rate.

Fig. 3.8 and Fig. 3.9 shows the results for channel rate of 1Mb/s and 2Mb/s respectively. Our results indicate that when mean inter-arrival time is small then the channel is more utilized in a sense that total number of video transmission is high. On the other hand, for small mean inter-arrival time we observe that user acceptance rate is low. In addition, increasing the channel rate increase the both total number of video transmission and user acceptance rate. However, this increment is not necessarily linear.

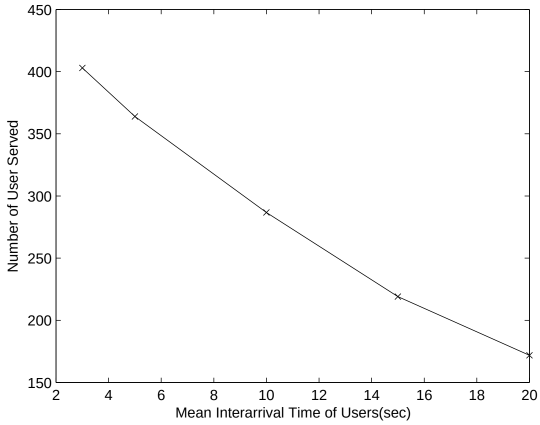


(A) Number of Users Served Vs Mean Interarrival Time λ

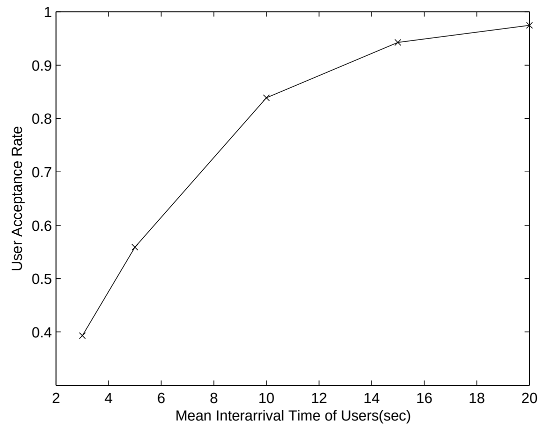


(B) User Acceptance Rate Vs Mean Interarrival Time λ

FIGURE 3.8: Analysis of user acceptance and total video transfer for channel rate of 1Mb/s



(A) Number of Users Served Vs Mean Interarrival Time λ



(B) User Acceptance Rate Vs Mean Interarrival Time λ

FIGURE 3.9: Analysis of user acceptance and total video transfer for channel rate of 2Mb/s

Chapter 4

Multiuser Video Streaming: Interruption Analysis

4.1 Jitter Probability for Single User

To analyze the jitter probability, we assume that time is divided into fixed length time slots and each time slot is denoted by l. In addition, channel experience independent and identically distributed (iid) block fading such that channel gain is constant over a time slot and it varies between the time slots independently. Duration of a time slot is equal to fixed packet length and ρ denotes the average probability of successful packet transmission. Since, the time interval is considered as a sequence of time slots l we use the notation of $R(l)$ and $p(l)$ for receiver curve and playback curve respectively. We remark that notation $p(l)$ is used for playback curve and playout curve interchangeably for ease of simplification in the notation. Let H denote the total number of GOPs in a given video file. Hence, there are K number of increment points in the playback curve of the corresponding video file and each increment point is denoted by l_h. Recall that for jitter-free video streaming, the total number of packets received by the user up to time slot l should be equal to or larger than $p(l)$ for any time slot $l > 0$ i.e.,

$$R(l) \geq p(l) \tag{4.1}$$

Hence, our scope of examination is to find a closed form expression for the probability $Pr(R(l) \geq p(l))$ to provision the jitter probability before server starts video streaming process. First we discuss previous works related to quantifying the jitter probability. Then, we demonstrate how to calculate the exact jitter probability for given channel statistic ρ, playback curve p(t) and initial playback delay Δ. Thereafter, we proposed a closed form expression for the approximate jitter probability. Finally, we prepare an experiment setup to compare these methods for varying video lengths and different channel statistics.

4.1.1 Upper Bound for Jitter Probability

In [38], authors proposed an upper bound for jitter probability using the notion of $R_{best}(l)$ curve. $R_{best}(l)$ curve describes the receiver curve in the best case such that at

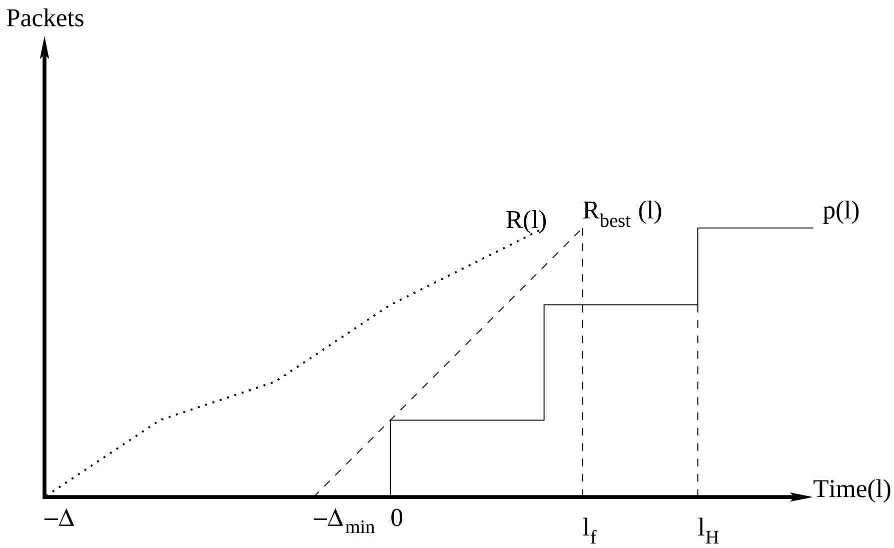


FIGURE 4.1: R_{best} Curve

any time slot l packet transmission is successful. For a given playback curve $p(l)$, Δ_{min} is the minimum playback delay for jitter free streaming that can only be achieved with $R_{best}(l)$. Notice that l_f denotes the first time instant where $R_{best}(l) \geq p(l_H)$. From Fig. 4.1 observe that if $R(l) \geq R_{best}(l)$ holds for all $l \in [-\Delta_{min}, l_f]$, then we ensure jitter-free streaming such that $R(l) \geq p(l)$ holds for any l. Therefore, $Pr(R(l) \geq R_{best}(l))$ provides a lower bound for the probability of jitter-free streaming and equivalently

$$\pi_u = 1 - Pr(R(l) \geq R_{best}(l)) \tag{4.2}$$

is a upper bound for jitter probability. Note that if $R_{best}(l) > R(l)$ for some $l \in [-\Delta_{min}, l_f]$ then $R_{best}(l_f) > R(l_f)$ holds certainly since $R_{best}(l)$ has the maximum achievable slope. On the other hand, if $R(l_f) \geq R_{best}(l_f) = p(l_H)$ holds then it means that $R(l) \geq R_{best}(l)$ should hold for all $l \in [-\Delta_{min}, l_f]$. Therefore, we use $Pr(R(l_f) \geq p(l_H))$ instead of $Pr(R(l) \geq R_{best}(l))$ because that it is easier to find a closed form expression for $Pr(R(l_f) \geq p(l_H))$.

Recall that Δ_{min} is given as,

$$\Delta_{min} = \max_{l} \left\{ R_{best}^{-1}(p(l)) - l \right\}. \tag{4.3}$$

Further, due to structure of the $R_{best}(l)$ curve, the time interval $[-\Delta_{min}, l_f]$ corresponds to transmission of $p(l_H)$ packets. Hence, l_f is given as $-\Delta_{\text{min}} + p(l_H)$. Let M denote the total number of time slots up to l_f such that $M = \Delta + l_f$. Then, upper bound for jitter probability π_u given as,

$$\pi_u = \sum_{i=p(l_H)}^{M} \binom{M}{i} \rho^i (1-\rho)^{M-i}. \tag{4.4}$$

4.1.2 Buffer Starvation

In [42] buffer level of user is modeled as a $M/M/1$ queue with Poisson arrival process with a rate of λ and Poisson service process with a rate of μ. Therefore, arrival of a packet occurs with probability $\rho = \frac{\lambda}{\lambda+\mu}$ and departure of a packet occurs with probability $1-\rho$. For the initial queue length of x_1 and the total size N of a file, buffer starvation probability [42] is given as

$$\pi = \sum_{k=x_1}^{N-1} \frac{x_1}{2k-x_1} \binom{2k-x_1}{k-x_1} \rho^{k-x_1} (1-\rho)^k. \tag{4.5}$$

Since playback curve information $p(l)$ is available, it is easy to determine the departure rate μ. Furthermore, we assume that channel statistics are known, thus it is also possible to assign a proper value to λ.

4.1.3 Exact Calculation of Jitter Probability

In this subsection we demonstrate a recursive method to calculate the exact jitter probability π for given playback curve $p(l)$, initial delay Δ and channel statistic ρ. First of all let us define the probability vector $\boldsymbol{pv_h}$ and assume that a GoP duration corresponds to D time slots.

Definition 3. $\boldsymbol{pv_h}$ is a vector of probabilities such that $\boldsymbol{pv_h}(i)$ indicates the sum of the probabilities of the jitter-free realizations of receiver curve $R(l)$ up to l_h with a value of $R(l_h) = p(l_h) + (i-1)$ i.e.,

$$\boldsymbol{pv_h}(i) = Pr\left(R(l_h) = (p(l_h) + (i-1)) \mid R(l_{h-1}) \geq p(l_{h-1}), \ldots, R(l_1) \geq p(l_1)\right). \tag{4.6}$$

where $1 \leq i \leq mind_h$, and $mind_h$ defines the maximum index for $\boldsymbol{pv_h}$. Remark that total number of packet transmission up to time slot l_h is $\Delta + (h-1)D$ and $\boldsymbol{pv_h}$ contains the probabilities for jitter-free realizations up to time slot l_h, thus for the maximum index we have the following inequality $mind_h \leq \Delta + (h-1)D - p(l_h) + 1$. Further, notice that $\sum_{i=1}^{mind_h} \boldsymbol{pv_h}(i)$ corresponds to probability that no jitter event occurs up to time slot l_h

The aim of this method is to obtain $\boldsymbol{pv_H}$, since jitter probability π is equal to $\sum_{i=1}^{mind_H} \boldsymbol{pv_H}(i)$ from definition of the $\boldsymbol{pv_h}$. Now we want to show how to procure $\boldsymbol{pv_H}$ recursively. Let $\boldsymbol{pv_0}$ contain the probabilities of the all outcomes (number of successful packet transmission) from D trials (total number of packet transmission in a GoP duration) such that

$$\boldsymbol{pv_0}(i) = \binom{D}{i-1} \rho^{i-1}(1-\rho)^{D-i+1}, \tag{4.7}$$

In addition, let $\boldsymbol{\acute{pv}_h}$ denote the convolution of the $\boldsymbol{pv_h}$ and $\boldsymbol{pv_0}$. Then, the ith index of the $\boldsymbol{\acute{pv}_h}$ defines probability that receiver curve $R(l)$ achieves the value $R(l_{h+1}) = p(l_h) + (i-1)$ at time slot l and up to time slot l_h realization of the receiver curve $R(l)$ is jitter-free i.e.,

$$Pr(R(l_{h+1}) = p(l_h) + (i-1)|\ R(l_h) \geq p(l_h), \ldots, R(l_1) \geq p(l_1)), \tag{4.8}$$

where $1 \leq i \leq mind_h + D$. Observe that for $1 \leq i \leq p(l_{h+1}) - p(l_h)$, we have the inequality $R(l_{h+1}) < p(l_{h+1})$ which means for those given probabilities jitter event occurs at time slot l_{h+1}. Therefore, if we remove the first $p(l_{h+1}) - p(l_{h+1})$ elements of the $\boldsymbol{\acute{pv}_h}$ and denote the remaining vector with $\boldsymbol{\check{pv}_h}$, then $\boldsymbol{\check{pv}_h}(i)$ defines the sum of probabilities of jitter-free realizations of receiver curve $R(l)$ up to l_{h+1} with a value of $R(l_{h+1}) = p(l_{h+1}) + (i-1)$ i.e.,

$$Pr(R(l_{h+1}) = p(l_{h+1}) + (i-1)|\ R(l_h) \geq p(l_h), \ldots, R(l_h) \geq p(l_h)). \tag{4.9}$$

We monitor that $\boldsymbol{\check{pv}_h}$ is identical to $\boldsymbol{pv_{h+1}}$. Then, it is possible obtain the probability vector $\boldsymbol{pv_H}$ recursively and find the exact probability of jitter.

4.1.4 Approximation for the Jitter Probability

In this subsection, we search for a closed form expression for the approximate jitter-probability. First we define the required minimum average rate $\gamma_{\min}$ such that $\gamma_{\min}$ is the minimum slope that makes the line ,from time instant 0 to t_H, to stand over the

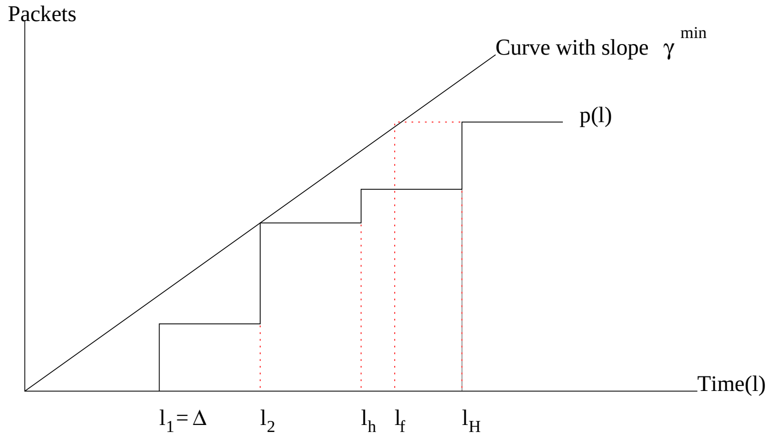


FIGURE 4.2: Minimum slope $\gamma_{\min}$

given playback curve. Hence, $\gamma_{\min}$ is given by

$$\gamma_{\min} = \max_{h=1,\ldots,H} \frac{p(l_h)}{l_h} \tag{4.10}$$

Let l_f denote the first time slot that the line with slope $\gamma_{\min}$ reach the value of $p(l_H)$ such that $l_f = \left\lceil \frac{p(l_H)}{\gamma_{\min}} \right\rceil$. Then, we claim that

$$\pi = Pr(R(l) \geq p(l)) \approx Pr(R(l_f) \geq p(l_H)). \tag{4.11}$$

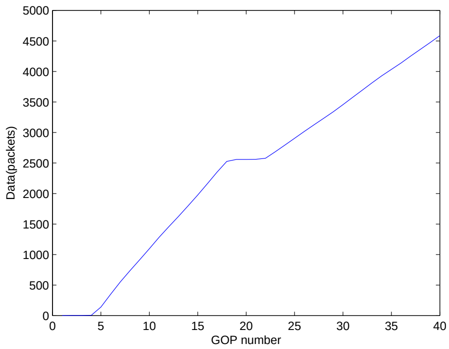
Hence, approximate jitter probability is given as

$$\pi = \sum_{i=p(l_H)}^{M} \binom{l_f}{i} \rho^i (1-\rho)^{l_f - i}. \tag{4.12}$$

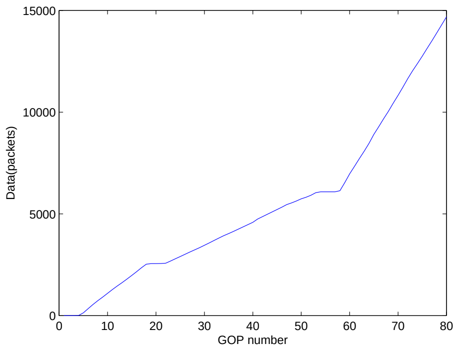
4.1.5 Numerical Results

In this section, we compare different methods that are used to measure jitter probability. Eventually, we will show that proposed algorithm to find approximate jitter probability performs the best result among the other methods. The video file used in the experiments are encoded according to single layer H.264/AVC standard [69]. We consider the encoded video file of Silence of the Lambs movie with CIF resolution (352x288) at 30 frames per second. A GOP size of 16 and a frame configuration of 3 B frames in between I/P key pictures is used [69]. In addition, in the experiment we use the video file with fixed quantization parameter $QP = 16$. In the simulation set up, we consider the first 20,40,100 seconds of the same video file.

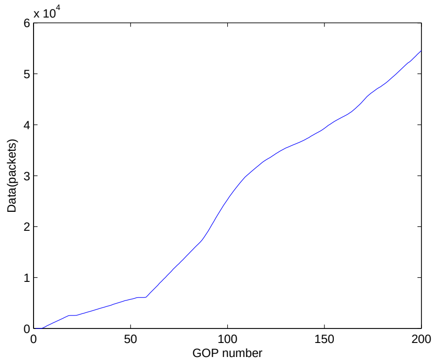
Fig. 4.3 monitors the playback structure of video fragments with different video lengths. We set Ip packet size to 1500 bytes [71] and initially we assume maximum transmission



(A) Playback Curve for Video Duration of 20Seconds



(B) Playback Curve for Video Duration of 40 Seconds

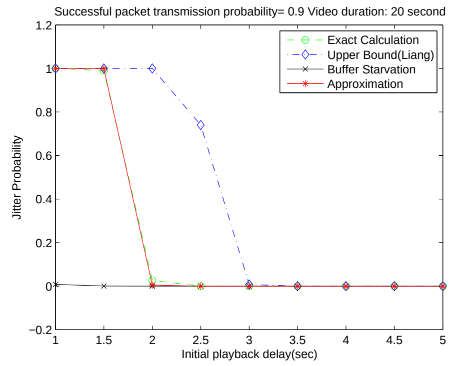


(C) Playback Curve for Video Duration of 100 Seconds

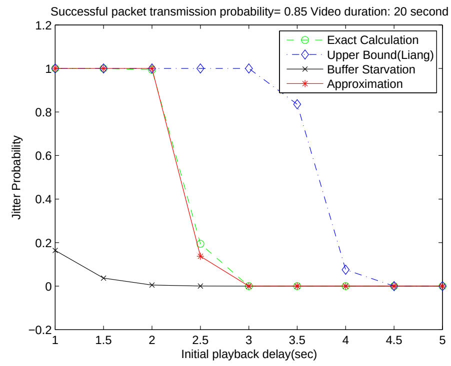
FIGURE 4.3: Playback Curves for Different Video Durations

bit rate as 1Mb/s. Notice that ratio of the packet size to maximum transmission bit rate gives the fixed packet length. Further, ratio of GOP duration to fixed packet length is the total number of trial in a GOP duration. In the experiment for each video length we consider different channel statistics, in other words we change the successful packet transmission probability ρ. 1500 bytes [71] and initially we assume maximum transmission bit rate of 1Mb/s. Notice that ratio of the packet size to maximum transmission bit rate gives the fixed packet length. Further, ratio of GOP duration to fixed packet length is the total number of trial in a GOP duration. In the experiment for each video length we consider different channel statistics, in other words we change the successful packet transmission probability ρ.

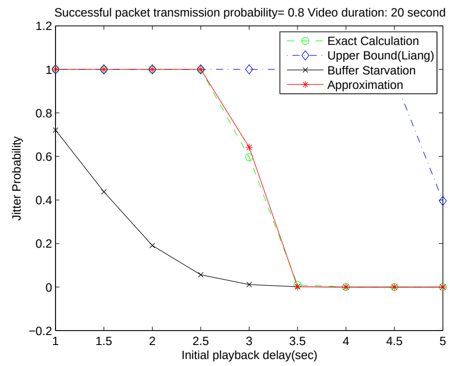
Fig. 4.4 and Fig. 4.5 show the results for video duration of 20 seconds and 40 seconds respectively. Note that when we use the video fragment with length of 40 seconds, maximum transmission bit rate is set to 1.4Mb/s. However we also investigate the cases where maximum transmission bit rate is 1Mb/s and 1.2Mb/s. Remark that there are 2 different ways to avoid the video interruption. These are increasing the playback delay and increasing the maximum transmission bit rate which is corresponds to the total



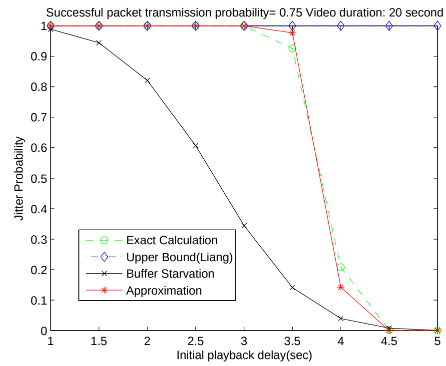
(A) Jitter Probability Vs Playback Delay for $\rho = 0.9$



(B) Jitter Probability Vs Playback Delay for $\rho = 0.85$



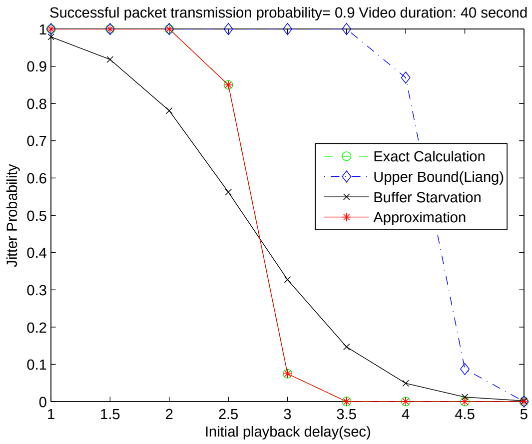
(C) Jitter Probability Vs Playback Delay for $\rho = 0.8$



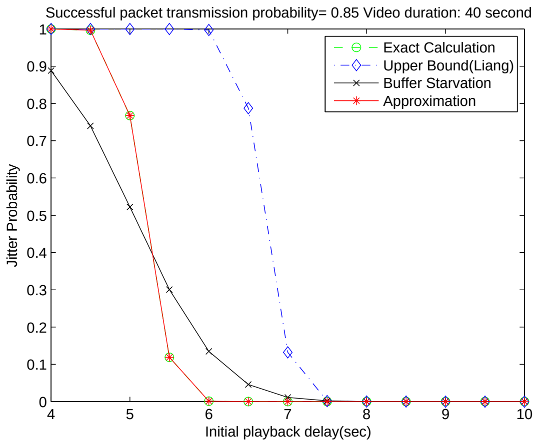
(D) Jitter Probability Vs Playback Delay for $\rho = 0.75$

FIGURE 4.4: Jitter Probability Analysis For Video Duration of 20 Seconds

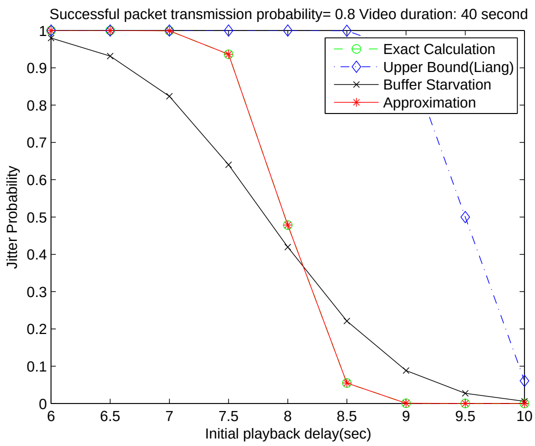
number of packet transmission in a GOP duration. Fig. 4.6 shows the effect of these jitter preventive methods on the strictness of jitter upper bound introduced in [38]. Fig. 4.6 verifies that, if the playback delay is increased instead of increasing the maximum transmission bit rate then upper bound for jitter probability performs better and for long playback delays it almost fit to exact jitter probability calculation. On the other hand, for short playback delay it provide a loss bound. This is because that in the jitter upper bound analysis we utilize the R_{best} curve. If the maximum transmission bit rate is high compared the average source rate of the video file then it implies that R_{best} curve is not close to playout curve of the user, thus it provides a loos upper-bound. Conversely, if the maximum transmission bit rate and the average source rate of the video file is close the each other then we have strict bound for jitter probability. However, when the maximum transmission bit rate and the average source rate of the video file is close the each other we need longer playback delay to prevent jitter. Fig. 4.7 show the results for video duration of 100 seconds. The results indicates that proposed approximate jitter probability method provides the closest results to exact jitter probability compared buffer starvation and jitter probability upper bound. In addition, it almost fit to exact jitter probability for different video durations. Remark that buffer starvation method



(A) Jitter Probability Vs Playback Delay for $\rho = 0.9$

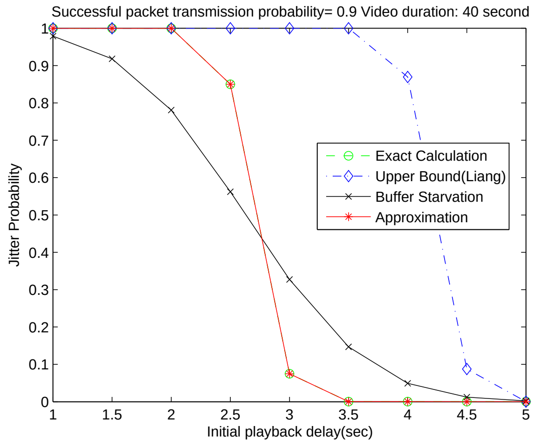


(B) Jitter Probability Vs Playback Delay for $\rho = 0.85$

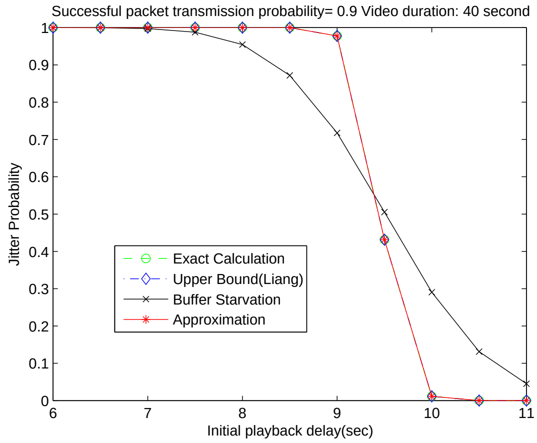


(C) Jitter Probability Vs Playback Delay for $\rho = 0.8$

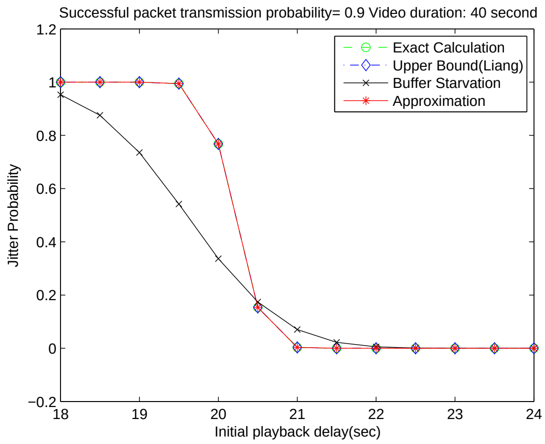
FIGURE 4.5: Jitter Probability Analysis For Video Duration of 40 Seconds



(A) Jitter Probability Vs Playback Delay (Maximum transmission bit rate = 1.4Mb/s

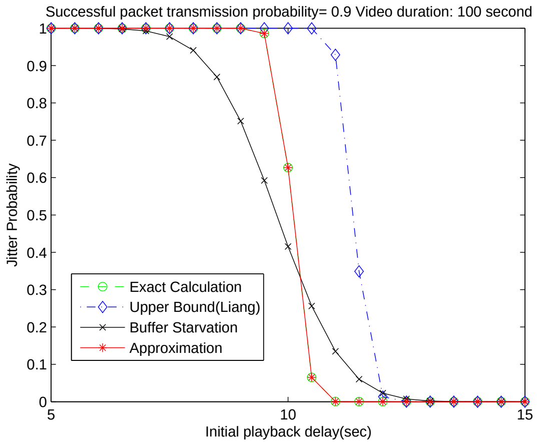


(B) Jitter Probability Vs Playback Delay (Maximum transmission bit rate = 1.2Mb/s

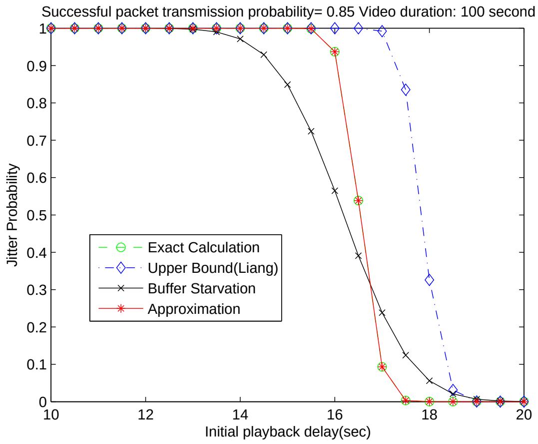


(C) Jitter Probability Vs Playback Delay (Maximum transmission bit rate = 1Mb/s

FIGURE 4.6: Jitter Probability Analysis For Video Duration of 40 Seconds for fixed ρ



(A) Jitter Probability Vs Playback Delay for $\rho = 0.9$



(B) Jitter Probability Vs Playback Delay for $\rho = 0.85$

FIGURE 4.7: Jitter Probability Analysis For Video Duration of 100 Seconds

does not fully utilize the playback information but use the average source rate statistic to compute jitter probability. Hence, it is not surprising that it performs better for longer vide files since video statistics are more reliable for longer video files in a sense that it depends on more GOP samples. Further, Fig. 4.7 monitors that we have a smooth transition for buffer starvation method as expected since we use the video statistics for jitter probability calculation. On the other hand, we observe more dramatic change in the case of exact jitter probability and approximation method. This is mainly because of the structure of the playback curve. In the playback curve structure some certain GOPs may be dominant over the others be the determinant factor for jitter probability. Hence, we conclude that for longer video files buffer starvation method provides close results to exact jitter probability, nevertheless ,due to its smoothly changing structure, does not fit to exact jitter probability.

4.2 Video Transmission to Multi-end Users Over Heterogeneous Channels

4.2.1 Introduction

In this case, ρ_n denotes the average probability of successful packet transmission for channel belong to user n. In this section, for given playout curves $\boldsymbol{p} = \{p_n(l)\}_{n=1}^N$ and channel statistics $\{\rho_n\}_{n=1}^N$, we aim to find the optimal scheduling policy $\omega^\star$ where the minimum system jitter probability is achieved i.e.,

$$\omega^\star = \underset{\omega}{\operatorname{argmin}}\, 1 - Pr(R_1(l) \geq p_1(l), \ldots, R_N(l) \geq p_N(l) |\ \omega). \tag{4.13}$$

Then, given minimization problem in (4.13) is solved by using Probabilistic Dynamic Programing in the following way. First of all $\boldsymbol{p}$ is considered as the state of the dynamic programing. We note that end of the each time slot current state is updated according to the previous state $\boldsymbol{p}$, scheduled user n and result of the action (success or fail). Define $\pi_l(\boldsymbol{p})$ to be the probability that a jitter event does not occur after time slot l for recent $\boldsymbol{p}$. In the dynamic programing $\pi_l(\boldsymbol{p})$ equal to the maximum expected reward that can be earned during next periods. Hence, $\pi_l(\boldsymbol{p})$ is obtained from the following recursion

$$\pi_l(\boldsymbol{p}) = \max_n \left(\rho_n \pi_{l+1}(\boldsymbol{p}|\ (n,1)) + (1-\rho_n)\pi_{l+1}(\boldsymbol{p}|\ (n,0))\right) \tag{4.14}$$

where $(n, 1)$ and $(n, 0)$ indicates successful and failed transmission for scheduled user n respectively. In addition, user n maximizing the right hand side of 4.14 is the optimal scheduling decision for time slot l. In the preliminaries, we explain nearest deadline first (NDF) algorithm in a details, however shortly NDF algorithm schedules the user i if there is a packet request of user i in the nearest deadline.

4.2.2 Preliminaries

4.2.2.1 Binomial Identities

let $g(k, n, \rho)$ denote the probability that kth success from consecutive independent Bernoulli trials with a parameter ρ, occurs at the nth trial such that

$$g(k, n, \rho) = f(k-1;\ n-1, \rho)\rho \tag{4.15}$$

where f is the probability mass function of binomial distribution with parameter ρ.

Lemma 9. Following identity holds

$$\sum_{n=k}^{N+1} g(k,n,\rho)F(n-1) = \rho \sum_{n=k-1}^{N} g(k-1,n,\rho)F(n) + (1-\rho)\sum_{n=k}^{N} g(k,n,\rho)F(n) \quad (4.16)$$

where $F(n)$ is an increasing function of n.

Proof: First, consider the following term $g(k-1,n,\rho)\rho + g(k,n,\rho)(1-\rho)$ which is written as

$$f(k-2;\ n-1,\rho)\rho^2 + f(k-1;\ n-1,\rho)(1-\rho)\rho =$$
$$\left(\binom{n-1}{k-2}\rho^k(1-\rho)^{n-1-(k-2)}\right) + \left(\binom{n-1}{k-1}\rho^k(1-\rho)^{n-1-(k-2)}\right). \quad (4.17)$$

Then, the expression in the above is simplified into

$$\left(\binom{n}{k-1}p^{k-1}(1-p)^{n-(k-1)}\right)p = g(k,n+1,p). \quad (4.18)$$

Hence, we show that

$$g(k,n+1,p) = g(k-1,n,p)p + (1-p)g(k,n,p). \quad (4.19)$$

Now observe the following steps for the remaining part of the proof,

$$p\sum_{n=k-1}^{N} g(k-1,n,p)F(n) + (1-p)\sum_{n=k}^{N} g(k,n,p)F(n) \quad (4.20a)$$
$$= \sum_{n=k-1}^{N} g(k-1,n,p)pF(n) + \sum_{n=k}^{N} g(k,n,p)(1-p)F(n) \quad (4.20b)$$
$$= \sum_{n=k}^{N} \left(g(k-1,n,p)p + g(k,n,p)(1-p)\right)F(n) + g(k-1,k-1,p)pF(k-1) \quad (4.20c)$$
$$= \sum_{n=k}^{N} g(k,n+1,p)F(n) + \underbrace{g(k-1,k-1,p)p}_{g(k,k,p)}F(k-1) \quad (4.20d)$$
$$= \sum_{n=k-1}^{N} g(k,n+1,p)F(n) \quad (4.20e)$$
$$= \sum_{n=k}^{N+1} g(k,n,p)F(n-1). \quad (4.20f)$$

4.2.2.2 Nearest Deadline First Algorithm

We first explain the nearest deadline first algorithm then show how to find corresponding jitter probability recursively. Let consider the case where the current time slot is l_c and there are M deadline at $l_c + l_1, \ldots, l_c + l_M$ respectively. Further, let $q_{n,m}$ denote the number of additional packet requested by user n between the deadline $m-1$ and m. We

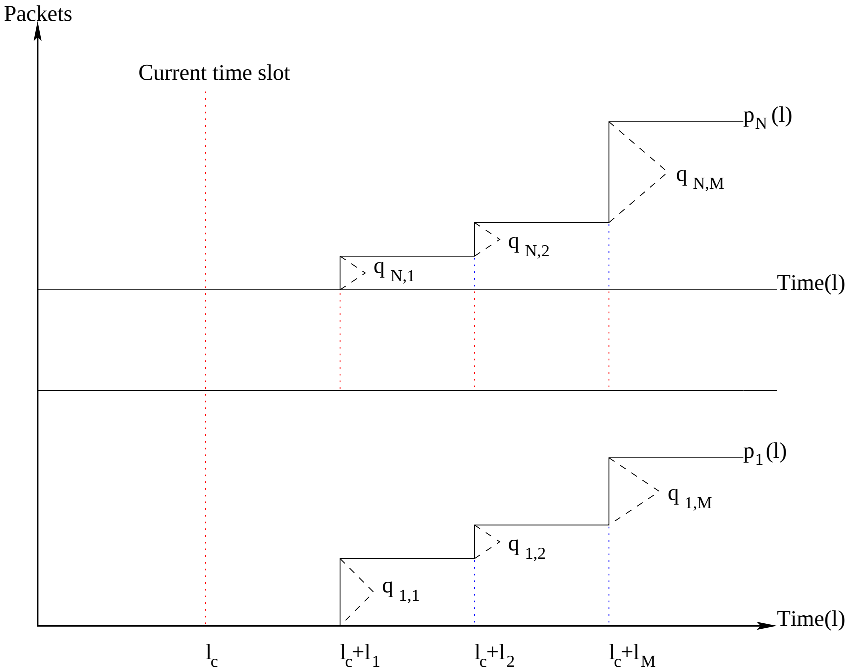


FIGURE 4.8: Playout curves at time slot l_c

note that playout curves in 4.8 defines the remaining packet demands for each user after time slot l. For this given case, user n is scheduled at time slot l_c if $q_{n,1} \neq 0$ according to the NDF algorithm. However, there might be more than one user satisfying the property $q_{n,1} \neq 0$. In such case any of them is a possible decision for NDF algorithm, nevertheless without loss generality we assume that user with minimum index is scheduled. Now we monitor how to derive $\pi_l(\boldsymbol{p})$ recursively. Let us define $k_{n,m}$ as the number of time slots allocated to user n to transmit $q_{n,m}$ packets. Further, K *is* used to define distance of a time slot to current time slot l_c. Notice that under the NDF algorithm, user with minimum index (let m be the smallest) among the users in the nearest deadline is served until it receives $q_{1,m}$ number of packets. In other words until the $q_{1,m}$th successful transmission user m is scheduled ceaselessly. Therefore, $\pi_l(\boldsymbol{p})$ given as

$$\sum_{k_{1,1}=q_{1,1}}^{l_1} g(q_{1,1}, k_{1,1}, \rho_1) F_{1,1}(k_{1,1}, q_{2,1}). \tag{4.21}$$

The term $F_{n,m}(K, q_{n,m})$ in the 4.21 is obtained by the following recursion

$$F_{n,m}(K, q_{n,m}) = \begin{cases} \sum_{k_{n,m}=q_{n,m}}^{l_m - K} g(q_{n,m}, k_{n,m}, \rho_n) F_{n+1,m}(K + k_{n,m}, q_{n+1,m}) & \text{if } n \neq N \\ \sum_{k_{1,m+1}=q_{1,m+1}}^{l_{m+1} - K} g(q_{1,m+1}, k_{1,m+1}, \rho_1) F_{1,m+1}(K + k_{1,m+1}, q_{1,m+1}) & \text{if } n = N \end{cases}$$

Now observe the following properties related to $F_{n,m}(K, q_{n,m})$. First of all, multiplying $F_{n,m}(K, q_{n,m})$ with some constant is equivalent to multiplying the term $F_{n+1,m}(K, q_{n+1,m})$ with the same constant . Secondly, decreasing the K increases the value of $F_{n,m}(K, q_{n,m})$ and finally if K is reduced for $F_{n,m}(K, q_{n,m})$ than K values for all other F functions ,that are take place after $F_{n,m}(K, q_{n,m})$ in the recursion (i.e., $F_{n,m+d}(K, q_{n,m+d})$, where $d \geq 0$), is also reduced by the same amount.

4.2.3 Optimal Scheduling Policy

Theorem 3. For given playout curves $\boldsymbol{p} = \{p_n(l)\}_{n=1}^N$ and channel statistics $\{\rho_n\}_{n=1}^N$ NDF algorithm provides the optimal scheduling policy ω such that ω is a solution to following optimization problem.

$$\underset{\omega}{\operatorname{argmin}}\, 1 - Pr(R_1(l) \geq p_1(l), \ldots, R_N(l) \geq p_N(l) |\ \omega). \tag{4.22}$$

Proof: Optimality of the NDF algorithm is proved by using the induction method. Let maximum video streaming duration is L. In the inductive step of the proof, we show that if NDF algorithm is the optimal scheduling policy between $[l+1, L]$ then at time slot l optimal scheduling decision should be serving to user with nearest deadline. In other words, NDF algorithm is also the optimal scheduling policy between $[l, L]$. The base case of the induction method is trivial since in the base case there is a single deadline.

Recall that scheduling the user n at time slot l is optimal decision if n maximizes the following expression $(\rho_n \pi_{l+1}(\boldsymbol{p}|\ (n,1))) + ((1-\rho_n)\pi_{l+1}(\boldsymbol{p}|\ (n,0)))$. Hence, we first show that user i with earliest deadline m_i is preferred against the user j with earliest deadline m_j if $m_i < m_j$. Then, we show that scheduling any user in the nearest deadline is optimal decision. To prove the first part, consider two different scheduling decision at time slot l where user i is scheduled in one case and user j is scheduled in other case. Now we want to show that $\pi_l(\boldsymbol{p}|\ i) \geq \pi_l(\boldsymbol{p}|\ j)$. Recall, $\pi_l(\boldsymbol{p}|\ n)$ is given as

$$\pi_l(\boldsymbol{p}|\ n) = (\rho_n \pi_{l+1}(\boldsymbol{p}|\ (n,1))) + ((1-\rho_n)\pi_{l+1}(\boldsymbol{p}|\ (n,0))) \tag{4.23}$$

Remark that recursion process for $\pi_{l+1}(\boldsymbol{p}|\ (n,1))$ and $\pi_{l+1}(\boldsymbol{p}|\ (n,0))$ identical up to $F_{n,m}$ where m is the earliest deadline for user n. In addition, utilizing the property of the F function we move ρ_n and $(1-\rho_n)$ to right before the terms $F_{n,m}(K, q_{n,m}-1)$ and

$F_{n,m}(K, q_{n,m})$. Hence, we consider a single recursion process with a new $F_{n,m}$ such that it is equal to $\rho_n F_{n,m}(K, q_{n,m} - 1) + (1 - \rho_n) F_{n,m}(K, q_{n,m})$. Then using the Lemma 2 we observe that

$$\rho_n F_{n,m}(K, q_{n,m} - 1) + (1 - \rho_n) F_{n,m}(K, q_{n,m}) = F_{n,m}(K - 1, q_{n,m}). \tag{4.24}$$

Now remark that difference in the recursion process for $\pi_l\left(\boldsymbol{p} \mid i\right)$ and $\pi_l\left(\boldsymbol{p} \mid j\right)$ occurs in the functions F_{i,m_i} and F_{j,m_j}. Recall the property of the F function which state that if K is reduced for $F_{n,m}(K, q_{n,m})$ than K values for all other F functions ,that are take place after $F_{n,m}(K, q_{n,m})$ in the recursion, also reduced by the same amount. Hence, we have $F_{j,m_j}(K - 1, q_{j,m_j})$ also in the recursion of the $\pi_l\left(\boldsymbol{p} \mid i\right)$ since $m_i < m_j$. However, same argument is not valid for $\pi_l\left(\boldsymbol{p} \mid j\right)$. Therefore, we conclude that $\pi_l\left(\boldsymbol{p} \mid i\right) \geq \pi_l\left(\boldsymbol{p} \mid j\right)$.

For an interruption-free video streaming, all the packets assigned to a deadline should be transmitted to corresponding users before the deadline. Notice that if we consider the packets in the same deadline, order of these packets in the transmission process does not have an impact on the interruption probability since reordering does not change the probability.

Now, we concentrate on a special case where transmission channels are identical. For this special case, it is possible to identify the system jitter probability since we know that NDF algorithm is the optimal scheduling policy. When the transmission channels are identical, then in terms of system jitter probability system can be considered as if there is a single user whose playback curve is aggregate of playback curves of the users.

4.2.4 Numerical Results

The video file used in the experiments are encoded according to single layer H.264/AVC standard [69]. We consider the encoded vide files with CIF resolution (352x288) at 30 frames per second. A GOP size of 16 and a frame configuration of 3 B frames in between I/P key pictures is used [69]. In the simulation scenario, we assume that there are three users are in the system an each of them request a different video stream. We further assume that transmission channels are identical with maximum transmission bit rate of 1 Mb/s and probability of successful packet transmission $\rho = 0.85$. Users request the following video files ; 100 seconds of Tokyo Olympics with quantization parameter $QP = 24$, 60 seconds of Silence of Lambs with $QP = 24$ and 80 seconds of Stars Wars with $QP = 34$ respectively. In the simulation we use NDF algorithm record the number of jitter event in 10^6 trials. Then, we utilize method for exact jitter probability calculation. Fig. 4.9 verifies our argument.

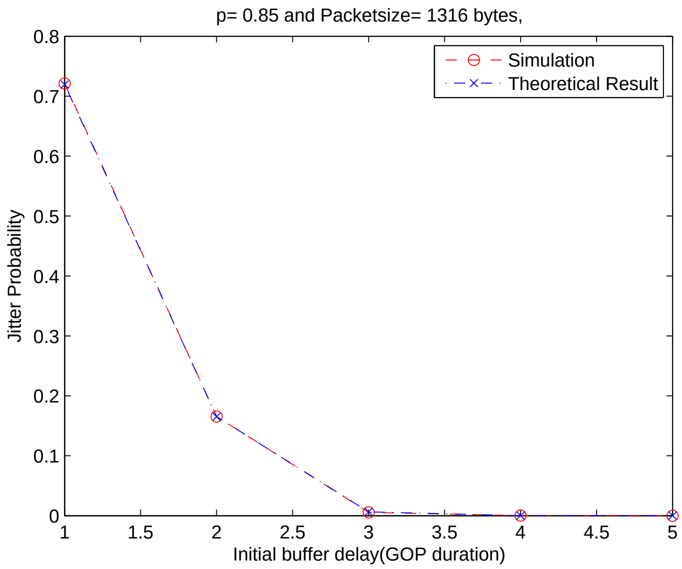


FIGURE 4.9: Simulation Result Vs Theoretical Result

Chapter 5

Conclusion and Future Works

5.1 Conclusions

In the first chapter, we have focused on discovering the optimum resource allocation strategies for providing jitter-free video streaming service to a multitude of end users. In the proposed multiuser video streaming setup, there are N different users requesting N different video files from a central video streaming server over a common communication channel characterized by the cumulative data rate curve $R(t)$. In this setup, the playback curve $p_i(t)$ of the video file requested by user i, $i = 1, \ldots, N$ characterizes the data rate demands placed by users on the server. For given $R(t)$ and $p_i(t)$, $i = 1, \ldots, N$ an important design parameter to avoid interruption, i.e., jitter, during the video streaming process is the playback delays that allow users buffering bits before displaying the requested video files. These playback delays need to be minimized to maximize the end user experience. Our main finding in this work is the derivation of the optimum rate splitting algorithm that establishes a robust procedure indicating how to divide $R(t)$ among N users in such a way that all users receive a jitter-free video streaming service and the maximum of playback delays, which we call the system delay, is minimized. More specifically, the proposed algorithm provides a family of rate splitting policies for any given $R(t)$ and $p_i(t)$, $i = 1, \ldots, N$ and any such rate splitting policy belonging to this family minimizes the system delay. In the first section, we assume that video requests of the users are synchronized and server has the perfect channel state information. Then we show that it is possible to obtain an optimal rate splitting policy by customizing Algorithm 1 for the case of asynchronised video requests. We further show that customization of Algorithm 1 to case of asynchronised video requests, minimize the residual cumulative play out curve which means that channel is fully utilized. A more comprehensive study can be done by increasing the variety of the video fragments and using Zipf function to model their request frequency.

In the second chapter, we first examine different methods to measure jitter probability when there is a single user in the video streaming system. Further, we show that NDF algorithm ,which schedules the user whose packets are in the nearest deadline, is the optimal scheduling policy in a sense that it minimizes the system jitter probability. Finally, we argue that if the transmission channels are identical, then in terms of system jitter probability system can be considered as if there is a single user whose playback curve is aggregate of the playback curves of the users. Hence, it is possible to determine system jitter probability. Our argument is also verified by the simulation results.

5.2 Future Works

In the future works, we will investigate the multiuser video streaming environment with a feed-back mechanism, so that beginning of the each time slot channel state information is known for each user. We aim to provide an optimal scheduling algorithm to minimize the system jitter-probability. Further, we will explore how much will the feedback system improve the quality of experience in terms of the system jitter probability.

Appendix A

NDF Algorithm

A.1 Theorem 3

Proof: Recall, $F_{n,m}$ is obtained by recursion process in the following way

$$F_{n,m}(K, q_{n,m}) = \sum_{k_{n,m}=q_{n,m}}^{l_m-K} g(q_{n,m}, k_{n,m}, \rho_n) F_{n+1,m}(K + k_{n,m}, q_{n+1,m}).$$

In the recursion equation $l_m - K$ defines the distance between current time slot an the deadline l_m, at the same time $l_m - K$ indicates the maximum number of time slots that can be allocated to user n before the deadline l_m. However, we know that if there are other users in the deadline l_m, then allocating all $l_m - K$ time slots to user n result in jitter certainly. Hence, we have a constraint for number of time slots that can be allocated to user n before the deadline l_m and let we call it $C_{n,m}$. $C_{n,m}$ is interpreted as the minimum number of time slot that should be allocated to users other than user n before the deadline l_m. Therefore, we have the following recursion instead of the previous one

$$F_{n,m}(K, q_{n,m}, c_{n,m}) = \sum_{k_{n,m}=q_{n,m}}^{l_m-K-C_{n,m}} g(q_{n,m}, k_{n,m}, \rho_n) F_{n+1,m}(K + k_{n,m}, q_{n+1,m}, c_{n+1,m}). \tag{A.1}$$

Recall, we want to show that for the optimal scheduling, we are indifferent between the users that are in the nearest deadline. If this claim is correct, then scheduling any user in the nearest deadline should give the same jitter probability. Let assume user n is scheduled, where $q_{n,1} \neq 0$ then corresponding jitter probability is given as

$$\pi_l\left(\boldsymbol{p}|\ n\right) = \left(\rho_n \pi_{l+1}\left(\boldsymbol{p}|\ (n,1)\right)\right) + \left(\left(1-\rho_n\right) \pi_{l+1}\left(\boldsymbol{p}|\ (n,0)\right)\right).$$

Note that when user n is scheduled and corresponding packet transmission is successful, then constraints $C_{i,1}$ for users $i < n$ will be decreased by one. Hence, we have

$$\rho_n \sum_{k_{1,1}=q_{1,1}}^{l_1-(C_{1,1}-1)} g(q_{1,1}, k_{1,1}, \rho_1) F_{1,1}(k_{2,1}, q_{2,}, C_{2,1} - 1)+$$

$$(1 - \rho_n) \sum_{k_{1,1}=q_{1,1}}^{l_1-C_{1,1}} g(q_{1,1}, k_{1,1}, \rho_1) F_{1,1}(k_{2,1}, q_{2,1}, C_{2,1}). \tag{A.2}$$

However, for F functions that take place after $F_{n,1}$ in the recursion, constraints $C_{i,j}$ are same for $\pi_{l+1}(\boldsymbol{p}|\ (n,1))$ and $\pi_{l+1}(\boldsymbol{p}|\ (n,0))$. Then, using the following identity

$$\begin{aligned}
&\rho_i g\left(q_{i,1}, l_1 - K - (C_{i,1} - 1), \rho_i\right) F_{i,1}\left(K + k_{i,1}, q_{i+1,1}, C_{i+1,1} - 1\right) + \\
&\sum_{k_{i,1}=q_{i,1}}^{l_1-K-C_{i,1}} g(q_{i,1}, k_{i,1}, \rho_i) F_{i,1}(K - 1 + k_{i,1}, q_{i+1,1}, C_{i+1,1}) \\
&= F_{i-1,1}(K - 1, q_{i,1}, C_{i,1})
\end{aligned} \tag{A.3}$$

we observe that (A.2) is equivalent to

$$\sum_{k_{1,1}=q_{1,1}}^{l_1-(C_{1,1}-1)} g(q_{1,1}, k_{1,1}, \rho_1) F_{1,1}(k_{1,1} - 1, q_{2,1}, C_{2,1}), \tag{A.4}$$

and this is true for all n with $q_{n,1} \neq 0$.

Bibliography

[1] Cisco. Cisco visual networking : Forecast and methodology, 2013-2018. Jun 2014.

[2] Sandvine. The global internet phenomena report: 2h 2014. Jun 2014.

[3] Point Topic. Iptv statistics market analysis. Jun 2013.

[4] Dapeng Wu and R. Negi. Effective capacity: a wireless link model for support of quality of service. *IEEE Transactions on Wireless Communications*, 2(4):630–643, July 2003.

[5] Dapeng Wu and R. Negi. Downlink scheduling in a cellular network for quality-of-service assurance. *IEEE Transactions on Vehicular Technology*, 53(5):1547–1557, Sept 2004.

[6] Dapeng Wu and R. Negi. Utilizing multiuser diversity for efficient support of quality of service over a fading channel. *IEEE Transactions on Vehicular Technology*, 54 (3):1198–1206, May 2005.

[7] Xi Zhang, Jia Tang, Hsiao-Hwa Chen, Song Ci, and M. Guizani. Cross-layer-based modeling for quality of service guarantees in mobile wireless networks. *IEEE Communications Magazine*, 44(1):100–106, Jan 2006.

[8] Jia Tang and Xi Zhang. Cross-layer modeling for quality of service guarantees over wireless links. *IEEE Transactions on Wireless Communications*, 6(12):4504–4512, December 2007.

[9] Xi Zhang, Jia Tang, Hsiao-Hwa Chen, Song Ci, and M. Guizani. Cross-layer-based modeling for quality of service guarantees in mobile wireless networks. *IEEE Communications Magazine*, 44(1):100–106, Jan 2006.

[10] Qinghe Du and Xi Zhang. Statistical qos provisionings for wireless unicast/multicast of multi-layer video streams. *IEEE Journal on Selected Areas in Communications*, 28(3):420–433, April 2010.

[11] A. Khalek, C. Caramanis, and R. Heath. Delay-constrained video transmission: Quality-driven resource allocation and scheduling. *IEEE Journal of Selected Topics in Signal Processing*, PP(99):1–1, 2014.

[12] Deli Qiao, M.C. Gursoy, and S. Velipasalar. The impact of qos constraints on the energy efficiency of fixed-rate wireless transmissions. *IEEE Transactions on Wireless Communications*, 8(12):5957–5969, December 2009.

[13] Lingjia Liu, Yang Yi, J.-F. Chamberland, and Jianzhong Zhang. Energy-efficient power allocation for delay-sensitive multimedia traffic over wireless systems. *IEEE Transactions on Vehicular Technology*, 63(5):2038–2047, Jun 2014.

[14] Song Ci, M. Guizani, and G.B. Brahim. A dynamic resource allocation scheme for delay-constrained multimedia services in cdma 1 times;ev-dv forward link. *IEEE Journal on Selected Areas in Communications*, 24(1):46–53, Jan 2006.

[15] Jia Tang and Xi Zhang. Quality-of-service driven power and rate adaptation for multichannel communications over wireless links. *Wireless Communications, IEEE Transactions on*, 6(12):4349–4360, December 2007.

[16] A. Abdrabou and Weihua Zhuang. Stochastic delay guarantees and statistical call admission control for ieee 802.11 single-hop ad hoc networks. *IEEE Transactions on Wireless Communications*, 7(10):3972–3981, October 2008.

[17] Jun Xu, Xuemin Shen, Jon W. Mark, and Jun Cai. Adaptive transmission of multi-layered video over wireless fading channels. *IEEE Transactions on Wireless Communications*, 6(6):2305–2314, June 2007.

[18] Jianwei Huang, Zhu Li, Mung Chiang, and A.K. Katsaggelos. Joint source adaptation and resource allocation for multi-user wireless video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(5):582–595, May 2008.

[19] J. Chakareski and P. Frossard. Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources. *IEEE Transactions on Multimedia*, 8(2):207–218, April 2006.

[20] Xiaoqing Zhu, P. Agrawal, J.P. Singh, Tansu Alpcan, and B. Girod. Distributed rate allocation policies for multihomed video streaming over heterogeneous access networks. *IEEE Transactions on Multimedia*, 11(4):752–764, June 2009.

[21] Xiaoqing Zhu and B. Girod. Distributed media-aware rate allocation for wireless video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(11):1462–1474, Nov 2010.

[22] Xiaoqing Zhu, T. Schierl, T. Wiegand, and B. Girod. Distributed media-aware rate allocation for video multicast over wireless networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(9):1181–1192, Sept 2011.

[23] Liang Zhou, Xinbing Wang, Wei Tu, G. Muntean, and Benoit Geller. Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks. *IEEE Journal on Selected Areas in Communications*, 28(3):409–419, April 2010.

[24] Chenglin Li, Hongkai Xiong, Junni Zou, and Zhihai He. Joint source and flow optimization for scalable video multirate multicast over hybrid wired/wireless coded networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 21 (5):550–564, May 2011.

[25] Fangwen Fu and M. van der Schaar. A systematic framework for dynamically optimizing multi-user wireless video transmission. *IEEE Journal on Selected Areas in Communications*, 28(3):308–320, April 2010.

[26] Chao Chen, R.W. Heath, A.C. Bovik, and G. de Veciana. A markov decision model for adaptive scheduling of stored scalable videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(6):1081–1095, June 2013.

[27] N.M. Freris, Cheng-Hsin Hsu, J.P. Singh, and Xiaoqing Zhu. Distortion-aware scalable video streaming to multinetwork clients. *IEEE/ACM Transactions on Networking*, 21(2):469–481, April 2013.

[28] Maodong Li, Zhenzhong Chen, and Yap-Peng Tan. Scalable resource allocation for svc video streaming over multiuser mimo-ofdm networks. *IEEE Transactions on Multimedia*, 15(7):1519–1531, Nov 2013.

[29] Xin Ji, Jianwei Huang, Mung Chiang, G. Lafruit, and F. Catthoor. Scheduling and resource allocation for svc streaming over ofdm downlink systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(10):1549–1555, Oct 2009.

[30] Chen Gong and Xiaodong Wang. Adaptive transmission for delay-constrained wireless video. *IEEE Transactions on Wireless Communications*, 13(1):49–61, January 2014.

[31] Huasen Wu, Xiaojun Lin, Xin Liu, and Youguang Zhang. Application-level scheduling with deadline constraints. In *INFOCOM, 2014 Proceedings IEEE*, pages 2436–2444, April 2014.

[32] Shizhen Zhao and Xiaojun Lin. Rate-control and multi-channel scheduling for wireless live streaming with stringent deadlines. In *INFOCOM, 2014 Proceedings IEEE*, pages 1923–1931, April 2014.

[33] Jie Xu, Y. Andrepoulos, Yuanzhang Xiao, and M. van der Schaar. Non-stationary resource allocation policies for delay-constrained video streaming: Application to video over internet-of-things-enabled networks. *IEEE Journal on Selected Areas in Communications*, 32(4):782–794, April 2014.

[34] V. Joseph and G. de Veciana. Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs. In *INFOCOM, 2012 Proceedings IEEE*, pages 567–575, March 2012.

[35] C. Chen, X. Zhu, G. de Veciana, A. Bovik, and R. Heath. Rate adaptation and admission control for video transmission with subjective quality constraints. *IEEE Journal of Selected Topics in Signal Processing*, PP(99):1–1, 2014.

[36] Hao Hu, Xiaoqing Zhu, Yao Wang, Rong Pan, Jiang Zhu, and F. Bonomi. Proxy-based multi-stream scalable video adaptation over wireless networks using subjective quality and rate models. *IEEE Transactions on Multimedia*, 15(7):1638–1652, Nov 2013.

[37] T. Stockhammer, H. Jenkac, and G. Kuhn. Streaming video over variable bit-rate wireless channels. *IEEE Transactions on Multimedia*, 6(2):268–277, April 2004.

[38] Guanfeng Liang and Guanfeng Liang. Effect of delay and buffering on jitter-free streaming over random vbr channels. *IEEE Transactions on Multimedia*, 10(6): 1128–1141, Oct 2008.

[39] T.H. Luan, L.X. Cai, and Xuemin Shen. Impact of network dynamics on user's video quality: Analytical framework and qos provision. *IEEE Transactions on Multimedia*, 12(1):64–78, Jan 2010.

[40] A. Anttonen and A. Mammela. Interruption probability of wireless video streaming with limited video lengths. *IEEE Transactions on Multimedia*, 16(4):1176–1180, June 2014.

[41] A. ParandehGheibi, M. Medard, A. Ozdaglar, and S. Shakkottai. Avoiding interruptions - a qoe reliability function for streaming media applications. *IEEE Journal on Selected Areas in Communications*, 29(5):1064–1074, May 2011.

[42] Yuedong Xu, E. Altman, R. El-Azouzi, M. Haddad, S. Elayoubi, and T. Jimenez. Analysis of buffer starvation with application to objective qoe optimization of streaming services. *IEEE Transactions on Multimedia*, 16(3):813–827, April 2014.

[43] Farid Molazem Tabrizi, Joseph Peters, and Mohamed Hefeeda. Dynamic control of receiver buffers in mobile video streaming systems. *IEEE Transactions on Mobile Computing*, 12(5):995–1008, May 2013.

[44] J.-P. Wagner and P. Frossard. Playback delay optimization in scalable video streaming. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 860–863, July 2005.

[45] Cheng-Shang Chang and J.A. Thomas. Effective bandwidth in high-speed digital networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1091–1100, Aug 1995.

[46] K. Stuhlmuller, N. Farber, M. Link, and B. Girod. Analysis of video transmission over lossy channels. *IEEE Journal on Selected Areas in Communications*, 18(6): 1012–1032, June 2000.

[47] L Kleinrock. *Queueing Systems, Volume 2: Computer Applications.* Wiley, New York, 1976.

[48] Hisashi Kobayashi. Application of the diffusion approximation to queueing networks i: Equilibrium queue distributions. *J. ACM*, 21(2):316–328, April 1974. ISSN 0004-5411.

[49] Jian Yang, Han Hu, Hongsheng Xi, and L. Hanzo. Online buffer fullness estimation aided adaptive media playout for video streaming. *IEEE Transactions on Multimedia*, 13(5):1141–1153, Oct 2011.

[50] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 187–198, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2836-4.

[51] A. Begen, T. Akgul, and M. Baugher. Watching video over the web: Part 2: Applications, standardization, and open issues. *IEEE Internet Computing*, 15(3): 59–63, May 2011.

[52] I. Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, 18(4):62–67, April 2011.

[53] Thomas Stockhammer. Dynamic adaptive streaming over http –: Standards and design principles. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, pages 133–144, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0518-1.

[54] Ya-Fan Su, Yi-Hsuan Yang, Meng-Ting Lu, and H.H. Chen. Smooth control of adaptive media playout for video streaming. *IEEE Transactions on Multimedia*, 11 (7):1331–1339, Nov 2009.

[55] Truong Cong Thang, Quang-Dung Ho, Jung Won Kang, and A.T. Pham. Adaptive streaming of audiovisual content using mpeg dash. *IEEE Transactions on Consumer Electronics*, 58(1):78–85, February 2012.

[56] Saamer Akhshabi, Ali C. Begen, and Constantine Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, pages 157–168, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0518-1.

[57] Truong Cong Thang, H.T. Le, A.T. Pham, and Yong Man Ro. An evaluation of bitrate adaptation methods for http live streaming. *IEEE Journal on Selected Areas in Communications*, 32(4):693–705, April 2014.

[58] Min Xing, Siyuan Xiang, and Lin Cai. A real-time adaptive algorithm for video streaming over multiple wireless access networks. *IEEE Journal on Selected Areas in Communications*, 32(4):795–805, April 2014.

[59] Junchen Jiang, V. Sekar, and Hui Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking*, 22(1):326–340, Feb 2014.

[60] Jiayi Liu, C. Rosenberg, G. Simon, and G. Texier. Optimal delivery of rate-adaptive streams in underprovisioned networks. *IEEE Journal on Selected Areas in Communications*, 32(4):706–718, April 2014.

[61] Yingsong Huang and Shiwen Mao. Downlink power control for multi-user vbr video streaming in cellular networks. *IEEE Transactions on Multimedia*, 15(8):2137–2148, Dec 2013.

[62] E. Danish, V. Silva, A. Fernando, C. Alwis, and A. Kondoz. Content-aware resource allocation in ofdm systems for energy-efficient video transmission. *IEEE Transactions on Consumer Electronics*, 60(3):320–328, Aug 2014.

[63] H. Abou-zeid, H.S. Hassanein, and S. Valentin. Energy-efficient adaptive video transmission: Exploiting rate predictions in wireless networks. *IEEE Transactions on Vehicular Technology*, 63(5):2013–2026, Jun 2014.

[64] Seong-Ping Chuah, Zhenzhong Chen, and Yap-Peng Tan. Energy-efficient resource allocation and scheduling for multicast of scalable video over wireless networks. *IEEE Transactions on Multimedia*, 14(4):1324–1336, Aug 2012.

[65] M. Ismail, Weihua Zhuang, and S. Elhedhli. Energy and content aware multi-homing video transmission in heterogeneous networks. *IEEE Transactions on Wireless Communications*, 12(7):3600–3610, July 2013.

[66] C. Singhal, S. De, R. Trestian, and G.-M. Muntean. Joint optimization of user-experience and energy-efficiency in wireless multimedia broadcast. *IEEE Transactions on Mobile Computing*, 13(7):1522–1535, July 2014.

[67] György Dán and Viktória Fodor. Quality differentiation with source shaping and forward error correction. In Giorgio Ventre and Roberto Canonico, editors, *MIPS*, volume 2899 of *Lecture Notes in Computer Science*, pages 222–233. Springer, 2003.

[68] D. Bethanabhotla, G. Caire, and M.J. Neely. Utility optimal scheduling and admission control for adaptive video streaming in small cell networks. In *2013 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 1944–1948, July 2013.

[69] P. Seeling, M. Reisslein, and B. Kulapala. Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial. *IEEE Communications Surveys Tutorials*, 6(3):58–78, Third 2004.

[70] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, July 2003.

[71] Cisco. Fundamentals of digital video. Jun 2014.