EVALUATION OF ADAPTIVE VIDEO OPTIMIZATION FOR STALL MINIMIZATION IN WIRELESS NETWORKS

by

KARISHMA KATARA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2015

To my family, friends and colleagues

Acknowledgements

I would like to express my sincere gratitude to Professor Dr.Qilian Liang, for supervising me and continuous support for the thesis and helping me in every step of my work. Thank you for giving me all these opportunities and a chance to work on something that I truly enjoy. I can truly say that working in your lab has been an once-in-a-lifetime experience.

I would like to thank my thesis committee Dr. Alan Davis and Dr. William Dillon for supporting me in this work.

I would also like to thank Lisa Parkin Director QA and Nitin Bhandari Senior Wireless Quality Engineer for guiding me during my Internship at Jasper Wireless U.S which helped in collecting real-world data used for this piece of work. Scott Staples Director-Policy Architect, thank you for offering advice on my project, laying the foundation as well as helping in my presentation. Mark Montz Senior Wireless Network Architect, thank you for everything from basic concepts, mathematical analysis and advice on how to present.

I would also like to thank Venugopal Velcheru for his valuable suggestions and guiding me during my course work at the University of Texas at Arlington. Thank you to all the lab mates of Wireless Communication and Networking for the immense help.

Last but not least I would like to thank my family and friends for always supporting and motivating me. I would not be where I am today without the love from my family. My parents and brother have always stood behind me no matter what I wanted to do.

July 2, 2015

Abstract

EVALUATION OF ADAPTIVE VIDEO OPTIMIZATION FOR STALL MINIMIZATION IN WIRELESS NETWORKS

Karishma Katara, M.S.

The University of Texas at Arlington, 2015

Supervising Professor: Qilian Liang

Mobile devices such as smart phones and tablets have become an integral part of peoples' daily lives. Users are consuming more content over wireless networks than ever before, and the largest portion of that traffic by volume is delivered as video. Even with the rollout of high speed LTE networks, subscriber demand for content is growing at an overall faster rate than network capacity is being added in many urban areas. This presents a challenge to the network operators since users expect a stall-free playback experience, on devices that support high resolution and high quality content.

To help maintain the subscriber experience, operators may deploy a solution to adjust the video transmission rate requirements depending on varying network capacity. Mobile users would otherwise experience sudden stalls during the video playback in a congested network.

The main objective of this thesis is to analyze a system or solution to optimize video content such that finite network resources are more fairly shared, and the subscriber experience is protected during times of network congestion by reducing the number of stalls. The goal is provide smooth video playback on the mobile device with the best video quality possible by applying adaptive video optimization. Adaptive optimization allows a change in

resolution, bit rate or quantization of the video in accordance with the available bandwidth. The user's experience for the video played on these device depends mainly on the stalls and the perception of quality.

Compared to the conventional compression schemes where optimization is applied universally regardless of network congestion, the adaptive optimization algorithm described in this paper adjusts the compression level applied based on the fluctuating channel throughput

In this paper, real world data collected from a live network is studied and analyzed how the number of stalls vary for dynamic compression. By simulating this scenario I can say that the highest overall playback quality possible is achieved by selecting the best video bit rate depending on the channel condition and the amount of video stream buffered in the client.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

1.1 Motivation

If a picture is worth a thousand words, a video is worth a million. A majority of internet traffic is video today. Mobile video services are very popular. Video-based social network such as YouTube [1] and Vimeo [2] are popular websites for people to share information. Cisco's traffic forecast states that video content over wireless networks is increasing tremendously and the mobile video data is expected to increase at a compound growth rate of 75% to around exobytes bytes by 2017 [3] [4]

With the increasing capacity in wireless communication systems and users growing need for video contents, a huge number of wireless video services are finding their way into our everyday lives and the growth will continue to accelerate as stated above.

It is very important for the wireless service provider to predict and deploy the capacity needed to support all of their subscribers. As the telecommunications infrastructure and device capabilities are evolving rapidly, the demand for mobile network services continues to grow. Clearly, the wireless network traffic is becoming more dynamic and complex. Hence, optimizing the video to provide a good subscriber experience for the video users is very important.

The capacity of a wireless network has a finite limit depending on a number of factors. These include: the radio technology used (LTE, WiMAX, Wi-Fi, UMTS, etc), the cell site density, frequency allocation, the number of subscribers, the content consumed by those subscribers, the time of day, backhaul bandwidth availability, the weather conditions and so on. This is a very complex set of variables, and the pace of innovation is large (new devices, new content providers, changes to existing content providers, etc)

making capacity planning a very difficult task. Based on the huge capital and operational expense required to deploy additional wireless capacity, modern telecom operators attempt to deploy capacity ahead of demand where possible, but generally not by a large factor to avoid large amounts of idle capacity. Based on this business model, it is common for operators to experience network congestion in regions where demand has grown faster than capacity has been added. In these locations, the quality of the service will suffer as subscribers compete to consume the available resources, and may grow frustrated with the resulting experience.

One method being used by modern operators to lessen this impact for their subscribers, is to intercept and optimize (compress) the video from web pages by examining the HTTP traffic enroute to the subscriber. This reduces payload size, and lessens the overall demand on network resources which results in a better subscriber experience averaged across the entire user base.

Depending on the network conditions and variations during the day, videos played from mobile devices are prone to stalling (buffering) during playback. Even 4G/LTE networks can get congested and experience stalling. Stalling can last from seconds to minutes which leads some subscribers to abandon their playback. Such stalls can causes subscriber frustration, loss of interest and low customer satisfaction with their wireless service provider.

For example, while a child is watching an animated video on a smartphone, each time the video stalls the child becomes upset. Once connectivity is restored, the child smiles and laughs. If the stall is prolonged, the child may cry until the video is restored. This is a clear demonstration of how the quality of experience is perceived by the most basic category of subscriber. While watching a movie on Netflix, the serving system will automatically drop playback quality (from High Definition to Standard

Definition for example) rather than pausing or buffering because stalls are known to be one of the key aspect of frustration.

The goal of this thesis is to optimize the video data in order to share the network resources fairly with reduced number of stalls and acceptable quality. As part of user experience testing in the wireless network the number of stalls and the loss image quality were observed.

1.2 Thesis Organization

Chapter 2 presents background information about the different optimization schemes considered in the thesis. Chapter 3 gives an overview of the Layer 7 system which is a part of the production network . Chapter 4 discusses the optimization formulation which details the analysis of the objective function. Chapter 5 outlines the congestion scheme used. Chapter 6 explains the compression used. Chapter 7 explains convex optimization method for stalls. Chapter 8 describes the rate distortion scheme and provide a metric for image quality. Chapter 9 discusses the network setup up which was investigated and a detailed analysis of the test results.Chapter 10 offers conclusions and recommendations for future research.

Chapter 2

Background

This chapter gives an introduction of the different optimization schemes and compression techniques considered in this paper.

2.1 Optimization Scenarios

Below are the three scenarios used in this paper for analysis of the benefit and impact of deploying a content optimization platform in a wireless operator's network:

1. **No optimization:** All the data is transferred between the mobile device and the content provider without modification (compression, transcoding, etc). The data is sent with the original quality available from the content provider regardless of whether the network is under congestion or not. As the data is not getting optimize when there is congestion, the video will stall and the quality of the picture is degraded too. It should be noted that original mobile screens were low resolution and small, so original video bandwidth requirements were lower that most home video systems. (etc)

2. **Traditional/Static optimization:** Optimize all content at a pre-defined setting. The optimization platform is not aware of network congestion, and hence the optimization is applied whether the subscriber is in a congested network region or not. The aim of the scheme is to save network resources such as RAN bandwidth based on offline analysis of the system performance. This saves bandwidth, which in turns leads to financial savings (either 3rdparty payment for wholesale RAN bandwidth consumed, or offsetting the cost of deploying more owned RAN resources).

The optimization techniques applied may be lossless (i.e. JIT) or lossy

One disadvantage of this approach is that in a non-congested network the data is compressed unnecessarily. Another disadvantage is that ,the quality will be reduced even if there was bandwidth for the original content.

Compression reduces the number of stalls gets but also the image quality in a congested network.Compression settings are applied globally and such global settings may not be aggressive enough to avoid stalls in ALL regions. The busiest locations will still experience playback stalls as well (estimated as 10% in this model).

3. **Dynamic Adaptive Optimization**: Optimization takes place dynamically depending upon the amount of congestion and other factors affecting bandwidth received by the handset present in the network.

- The types of optimization techniques used are both lossless and lossy compression
- In a non-congested network, no compression takes place. The data is sent with its original quality and without stalls
- In a congested network, the content may be compressed more aggressively depending on the severity of the congestion. This may lower quality even further than the static settings used in traditional optimization, but with the goal of eliminating all stalls even in busy locations.
- Lossy optimization is applied only to the subscribers receiving a poor experience due to network congestion.
- Lossless optimization may still be applied to all subscribers if network conditions allow ,providing the highest quality to the subscriber.
- The main aim of this dynamic adaptive optimization is to predict when to apply Lossy or Lossless video to a subscriber session. Similar techniques can be applied to web (text) and image content as well, but this model will focus on video content.

- Adaptive video optimization: Monitors media playback performance for each subscriber to determine how much optimization is required to maintain a high quality of experience. It uses a user experience index/quality of index for each subscriber.

The following items apply to either traditional or dynamic optimization of content:

- The level of compression is adjustable.
- The user experience level is determined by the proxy based on the amount of bandwidth available to support a stall-free transfer. The algorithm can fine-tune the compression to be more aggressive (optimize videos if there is even a small chance of stalling) or less aggressive (optimize videos only if system is very confident that the video will not stall).
- Sessions that have a good user experience with few delays have a high confidence level, and may not require lossy optimization. This doesn't impact the users unless it's absolutely necessary.
- Sessions that have a poor experience with many delays have a low confidence level, may require lossy optimization applied more aggressively.
- There may be levels in between these extremes.
- When the network bandwidth is sufficient to download the video without stalling, the original video is served to the subscriber.
- When the network bandwidth is insufficient, lossy optimization optimizes the video dynamically to improve the user experience by preventing stalling/buffering when network congestion increases and network

throughput decreases. This may temporarily reduce the visual quality of the video, but prevent it from stalling.

- Both the traditional and dynamic scheme uses “just in time” (JIT) techniques to spread the video delivery over the entire video play time. The browser buffering is limited to ensure unwatched content is not transferred if the subscriber abandons the video midway through playback.
- When we use dynamic adaptive video optimization, lossless optimization has no inherent impact to the subscriber experience, and hence it is applied in all scenarios. It is only important to selectively apply lossy optimization given there is an experience tradeoff in doing so.

Chapter 3

Layer 7 System Overview

3.1 Introduction

This chapter gives an introduction of the layer 7 content optimization platform. The production network is based on the 3rd Generation Partnership Project (3gpp) specifications [39]. The Layer 7 box is located on the Gi data bearer path between the GGSN/PGW and the external Firewall. Gi is a standard TCP/IP connection link. The ideal place is on the 3GPP defined "Gi" link between the GGSN/PGW, and the Internet via the firewall. This link is TCP/IP based as the GGSN/PGW is the IP mobility anchor for the session, and encapsulates all GTP(GPRS tunneling protocol-which carries GPRS with GSM,UMTS and LTE n/w) messaging from the network/device.

Layer 7 acts as a transparent proxy, intercepting all ingress and egress HTTP flows for the purpose of applying various optimization techniques. The Figure 3.1 is a physical representation of Layer 7 in the network:

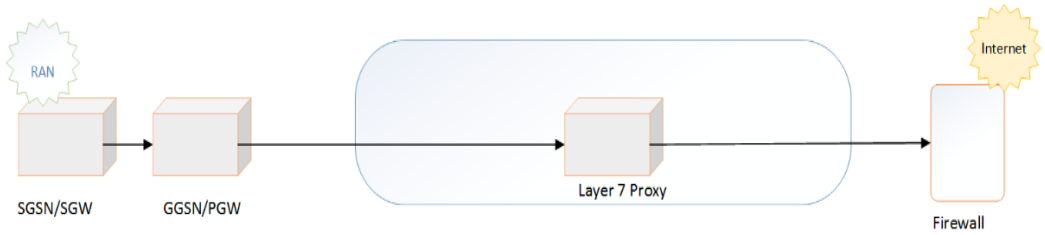


Figure 3.1 High level Physical Architecture

From a logical perspective, all the traffic that is subjected for optimization is routed across the platform and then to the firewall.

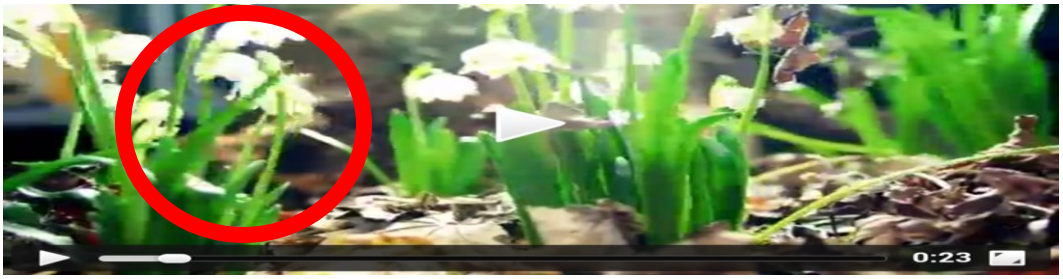
The primary purpose of the Layer 7 content optimization platform is basically to reduce the total data volume of data generated by the subscribers on the radio access network. This is achieved by intercepting the traffic from various content providers, and applying various optimization techniques before relaying back to the subscriber. The

subscriber consuming the content is not aware of the optimization technique that is being performed. It can be lossless or lossy compression based on the available bandwidth.

The solution will be in-line on the Gi but only ports 80/1935 will be routed to ATM. ATM will send TCP 80/1935 traffic to the application layer via HTTP classification rules[25]

Different techniques can be applied to achieve a balance between the compression ratio (%) and the impact to the Quality of Experience (QoE). The compromise in quality may appear as loss of resolution, pixelization (or 'blockiness') artifacts in fast moving scenes, loss of color depth.

Un-optimized content



Optimized content

Figure 3.2 Snapshot of different resolution video (plants)

Un-optimized content

Optimized content

Figure 3.3 Snapshot of different resolution video(human)

Observations:

- By applying 0% of compression, no savings will be obtained in the network. If the number of videos are increased from 50, note (at least) the video rate is assumed to be 1Mb/s per subscriber session, and the network will share the resources equally with a new subscriber. Users will see stalls as the available bandwidth is only 50Mbps
- Blocky lines, loss of detail, difference in color are seen from the images in Figure 3.2 and 3.3

- As the percentage of compression increases the number of videos streamed increases with a lowered quality but without stalling as explained earlier.
- This translates directly into the service provider being able to support a greater number of subscribers on infrastructure with a finite capacity. The service provider must decide how to configure the system for the tradeoff between quality of experience, and increase in effective capacity based on their business goals.
- As seen in the snapshot very minute differences can be seen, the quality of both look good in terms of customer experience. The amount throughput reduces based on the amount of compression.

The bandwidth allocation algorithm deals well with dynamically changing network conditions. Depending upon the congestion, the content is optimized before delivering it to the device. Content optimization in turn leads to reduced usage of the bandwidth.

The percentage of optimization is the bandwidth savings, which is very important for the operator. The more amount of savings, the more number of users can stream with good quality.

Chapter 4

Optimization Formulation

4.1 Objective function formulation

In real system development, the software architect normally prefers solving a series of snapshot problems rather than an unstructured dynamic evolution of the underlined system. In this paper, we specifically consider how to minimize the number of stalling events by compressing the data to the maximum threshold level and lowering the quality of the video to acceptable level. The system itself is proposed to use this information to determine the correct compression settings for each individual.

The problem can be formulated mathematically by using geometrical programming and convex optimization techniques. Geometric programming is a class of nonlinear optimization as stated in [26].Though the standard form of geometric programming is a non-convex optimization, it can be converted to a convex optimization problem. By doing so the local optimum is also a global optimum[28]. The scope of linear programming is an efficient mechanism that can be used in a real network.

The importance of convex optimization and formulations have increased drastically in the last decade as stated in [27] due to emergence of the new theory for rank minimization and structured sparsity, and successful statistical learning models.

The convex optimization algorithms stated in [11] can solve problems that include large data sets as nowadays the data sizes vary from terabytes to exabytes. One of the common problems faced by big data is the solution to a class of problems which are solved by composite formulations. The fundamentals of big data from [11] are given by the following composite formula:

$$F=\min_x \{F(x):=f(x)+g(x):x\varepsilon R^+\} \quad (4.1)$$

where f and g are convex functions. I use the numerical approaches to obtain the optimal solution x of (4.1) along with the assumption z and x are approximately linear practically and x is always an integer. x represents the quantization parameter and z represents the compression rateThe composite convex optimization of (4.1) $x\varepsilon R^{p}$ from the data $y\varepsilon R^{n}$ is:

$$\eta = \max_{x,z\in R^{+}} \{F(x,z):h(x)+g(z) : z=mx\} \quad (4.2)$$

The functions h and g are both efficient

The main optimization function in this thesis is formed using equation (4.2) subject to various constraints which will be explained in the chapter as shown below:

$$\max_{i=1,2,..m} \{\text{Picture quality}+\text{Non stall factor}\} \quad (4.3)$$

h(x) is the picture quality function. It can be varied by changing the quantization parameter value(x)

g(z) is the non stall factor function. It can be reduced by increasing the compression rate (z) or Com_{p}.

The value x represents the quantization parameter and z represents the compression rate. Compression rate and quantization parameter are roughly proportional to each other. H.264 compression is used.The level of compression used is noted as a "QP-level". The QP-levels have more compression at a higher levels, and it is approximately linear (although not exactly so). Since the QP levels are integers, an integer value of the compression is chosen that is equal to or greater than needed.

Though they follow a linear relationship theoretically in practical scenarios they don't.' m' represents the slope of the graph which looks to be -1 from Figure 12.It is an approximate linear fit ,a line is used to approximate so it doesn't choose the wrong QP value. QP results are taken to next highest integer.

Combining h(x) and g(z) subject to the various constraints like the bandwidth, bit rate, etc., the quality of the video should be maximized.If congestion is detected in the network for the same quality of the video and bandwidth we will experience stalling. In order to reduce the stalling, the quality of the video must be decreased to an acceptable level.

Quantization in an H.264 encoder is controlled by a quantization parameter, QP that ranges from 0 to 51. QP is an index used to derive a scaling matrix. It is possible to calculate the equivalent quantizer step size (Qstep) for each value of QP. As QP increases, Qstep increases; in fact, Qstep doubles for every increase of 6 in QP. The logarithmic relationship can be seen in this graph of QP (x-axis, linear) vs. Qstep (y-axis, logarithmic).

The objective function (4.2) maximize the subscriber experience by reducing the number of stalls and reducing the quality to an acceptable level. Stall time and video quality are directly dependent on each other which is shown in equation (4.2)

The constraints are both hard which set conditions for the variables that are required to be satisfied and soft constraints which have some variable values that are penalized in the objective function in case the conditions on the variables are not satisfied.

The constraints for equation (4.2) are:

Jointly subject to

$$f_c = 1/t_c \neq 0 \quad \text{(4.3a)}$$

Constraints (4.3a) state that the stall frequency is never equal to zero as stalling time is never allowed to be infinite

$$f_o = f_2 - f_1 \neq 0 \text{ (4.3b)}$$

Constraint (4.3b) states that bandwidth is never equal to zero

$$P_o > 0 \qquad (4.3c)$$

Constraint (4.3c) states that the probability of the buffer to be zero is greater than or equal to zero

$$C_p \neq 0 \qquad (4.3d)$$

Constraint (4.3d) states that congestion is always present in the network.

Constraint (4.3 e) states traditional compression which is always there in the network

$$\text{Variables } x,z\ ,\ T_r \qquad (4.3f)$$

The variables in (4.3f) represent the quantizer parameter(x), the percentage of compression(z) and reciprocal of video replay rate(T_r).

h(x) can be varied by changing the quantization parameter value

g(z) can be reduced by reducing the compression rate Com_p

Chapter 5

Congestion Control

5.1 Introduction

In terms of networks, congestion refers to the network state where a node/link carries more data than its capacity allows. Congestion deteriorates the service quality which results in queuing delay, packet loss, frame loss and blocking new connections. The response times slows down with reduced network throughput in a congested network. Congestion occurs when the bandwidth is insufficient and the network data traffic exceeds its capacity at least locally.

The Transmission Control Protocol (TCP) is the most widely used transport protocol in the Internet architecture. A large majority of traffic carried in the Internet is controlled by TCP.

TCP provides a connection oriented and reliable byte stream service meaning any two applications using TCP must establish a connection with each other before they can exchange data in a reliable way.

Figure 5.1 shows the position of TCP in the internet architecture. TCP provides a reliable service whereas User Datagram Protocol (UDP) does not give any guarantee of delivery. Both UDP and TCP use the functionalities offered by the Internet Protocol (IP) layer of the International Standards Organization (ISO) protocol stack

TCP offers to several services to applications including data segmentation, reassembly services ,implements flow and congestion control mechanism as well as error recovery techniques. The design of TCP is like a black-box vision of the network. The network shouldn’t provide any explicit information to TCP. For this reason TCP defines

and updates several state variables like the congestion window, round trip time that represent the actual view of the network state.

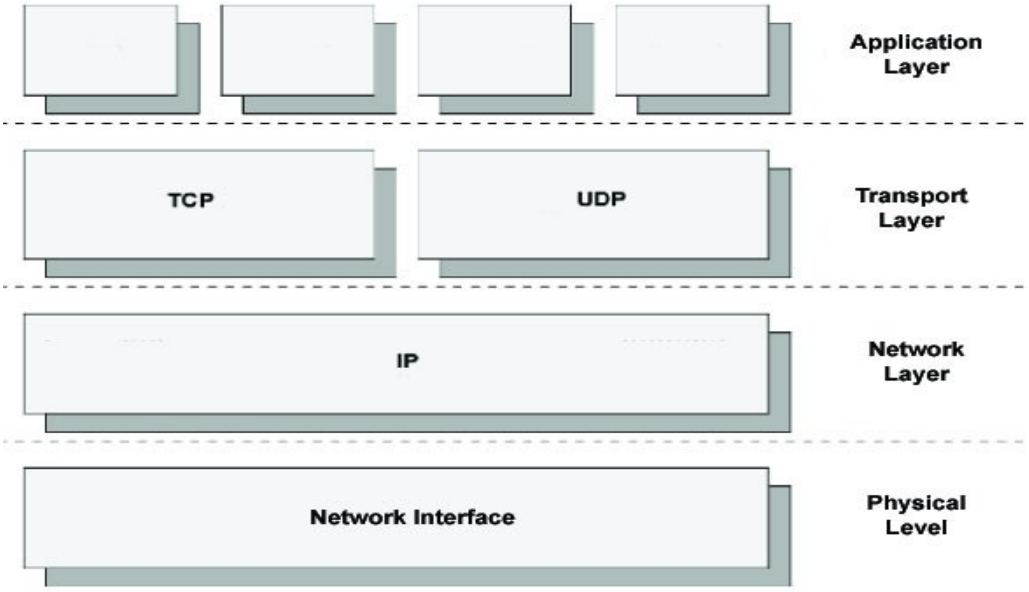


Figure 5.1 Internet Protocol Architecture

TCP implements a control on its transmission rate using a sliding window algorithm based on the receipt of the acknowledgements (ACK) and retransmits the lost segments to achieve an errorless file transfer.

TCP numbers every byte with a sequence number and the number of first byte contained in the segment is reported in the sequence number field of the segment itself.

5.2 TCP westwood overview

The traditional optimization always occurs regardless if the network is congested or not. In order to reduce the bandwidth efficiently an adaptive optimization is used with help of the TCP Westwood (TCPW) [10][24] algorithm.

- The TCPW algorithm is a sender side only modification of the stack. It implements the classic slow-start and congestion avoidance phases to probe the network but once congestion is detected, it makes use of the estimate of the

best-effort available bandwidth ‘B’ to adaptively set the congestion window and the slow-start threshold. The increase after congestion is additive but decrease Adaptive (AIAD) as compared to AIMD (Additive Increase Multiplicative decrease) of Reno.

When n duplicate acknowledgments(DUPACKs) are received it means the network capacity is reached. TCPW sets both the slow start threshold and the congestion window equal to the ideal window w=BWE*RTTmin. BWE represents the estimated bandwidth and RTT min is the minimum round trip time.

Algorithm after n duplicate ACKs(aka DUPACKS):

if (n DUPACKS are received)

ssthresh=(BWE*RTTmin)/seg_size;’’ssthresh-slow start threshold,BWE-bandwidth,RTTmin-rount trip time)

if(cwin>ssthresh) //congestion avoidance//

cwin=ssthresh;

endif

endif

The seg_size refers to the length of the payload in bits. During the congestion avoidance phase we are probing for extra available bandwidth.

When n duplicate acknowledgments(DUPACKs) are received it means the network capacity is reached. TCPW sets both the slow start threshold and the congestion window equal to the ideal window w=BWE*RTTmin. BWE represents the estimated bandwidth and RTT min is the minimum round trip time.The value of RTTmin is set to the smallest RTT sample observed over the duration of the connection. This setting allows the queue to be drained after a congestion episode. Also, after the ssthresh has been set, the congestion window is set equal to the slow start threshold only if cwin>ssthresh.

The main reason for using the end-to-end estimate of the bandwidth BWE is to set the slow start threshold is that the TCP exploits it the slow start phase to probe for available bandwidth. Seg_size represents the length of a TCP segment in bits

Algorithm after coarse timeout expires:

if (coarse timeout expires)

ssthresh=(BWE*RTTmin)/seg_size;

if(sshthresh<2)

ssthresh=2;

endif

cwin=1;

endif

After a timeout, cwin and ssthresh are set equal to 1 and the BWE respectively.

Note: When the ACKs are successfully received, TCPW increases cwin.

TCPW sender monitors the acknowledgment reception rate and from it estimates the data packet rate currently achieved by the connection. Whenever a sender perceives a packet loss (i.e. a timeout occurs or 3 duplicate ACKs are received), the sender uses the bandwidth estimate to properly set the congestion window (cwin) and the slow start threshold (ssthresh).

By backing off to cwin and ssthresh values that are based on the estimated available bandwidth (rather than simply halving the current values as Reno[10][28] does), TCP Westwood avoids overly conservative reductions of cwin and ssthresh; and thus it ensures a faster recovery.

Most importantly, TCPW is very effective in handling wireless loss. This is because TCPW uses the current estimated rate as [24] reference for resetting the

congestion window. The current rate is only marginally impacted by loss. In TCP Reno and the previous schemes the cwin is reduced to half blindly.

During the congestion avoidance phase we are probing for extra available bandwidth. Therefore, when n DUPACKs are received, it means that we have hit the network capacity (or that, in the case of wireless links, one or more segments were dropped due to sporadic losses). Thus, the slow start threshold is set equal to the available pipe size when the bottleneck buffer is empty, which is BWE*RTTmin/seg_size, the congestion window is set equal to the ssthresh and the congestion avoidance phase is entered again to gently probe for new available bandwidth. The value RTTmin is set as the smallest RTT sample observed over the duration of the connection.

This setting allows the queue be drained after a congestion episode. During the slow-start phase we are still probing for the available bandwidth. Therefore the BWE we obtain after n duplicate ACKs is used to set the slow start threshold. After ssthresh has been set, the congestion window is set equal to the slow start threshold only if cwin>ssthresh. In other words, during slow start, cwin still features an exponential increase as in the current implementation of TCP Reno.

The rationale of the algorithm is again simple. After a timeout cwin and ssthresh are set equal to 1 and BWE, respectively, so that the basic Reno behavior is still captured, while a speedy recovery is granted by the ssthresh being set to the bandwidth estimation at the time of timeout expiration.

Bandwidth Estimate Algorithm:

TCPW estimates the bandwidth by means of the flow of returning ACKs. Due to delays and ACKs compression, the flow of the returning ACKs must be low-pass filtered in an appropriate way [30,31]

In particular, when an ACK is received at time t_k ,it means that certain amount of data d_k has been received by the TCP receiver. On ACK reception, a sample of the available bandwidth can be computed:

$$b_k = \frac{d_k}{t_k - t_{k-1}} = \frac{d_k}{\Delta_k} \quad (5.1)$$

where t_{k-1} the time is the previous ACK was received and Δ_k is the last inter arrival time.

The amount of d_k data acknowledged by an ACK is determined by a proper counting procedure and b_k samples are low-pass filtered using a time-varying filter:

$$\hat{b}_k = \frac{2\tau_f - \Delta_\kappa}{2\tau_f + \Delta_\kappa} \hat{b}_{k-1} + \Delta_\kappa \frac{b_k + b_{k-1}}{2\tau_f + \Delta_\kappa} \quad (5.2)$$

where τ_f is the filter time constant .A modified version of the filter is used to compensate for the ACK compression effects along the ACK path, one bandwidth sample is computed every RTT. In more clear terms we count all the data d_k acknowledged during the last RTT and compute the bandwidth sample $b_k = \frac{d_k}{\Delta_k}$,where Δ_k is the last RTT.

In order to avoid aliasing effects due to the filter described in (5.2) Nyquist theorem can be used. It states that , Δ_k ,the sampling time needs to be smaller than τ_f /2

to obtain a properly working discrete time low pass filter with time constant τ_f .We assume $\Delta_k < \tau_f/4$ and when $\Delta_k > \tau_f/4$, interpolate and re-sample using N=int(4.RTT/τ_f) virtual samples b_k arriving with the inter arrival time $\Delta_k = \tau_f/4$

By the flow conservation principle, the estimator provides a value if the bandwidth ,B ,isclose to the input rate cwin/RTT. Therefore B=cwin/RTT

The value for DUPACKs counts towards the bandwidth estimation since their arrival indicates a successfully received segment in the wrong order.

The two important aspects of the bandwidth estimation process are:

(i)The source should keep track of the number of DUPACKs it has received before the new data is acknowledged.

(ii)The source should be able to detect delayed ACKs and act accordingly.

Equation for TCPW Throughput:

$$r^{West} = \min[\frac{W\max}{RTT}, \frac{1}{\sqrt{RTT.T_q}}\sqrt{(1-p)/p}]$$

where:

p=drop probability

RTT=round trip time

T_q =Mean queuing time.RTT-RTTmin

$W\max$ -Maximum value for the congestion window cwin

r^{West} -Steady state mean throughput of the flow

By further derivation we find that the throughput of Westwood depends on the round trip time $\frac{1}{\sqrt{RTT}}$ and increases fair sharing of the network capacity among flows experiencing different RTTs.

Simulation:

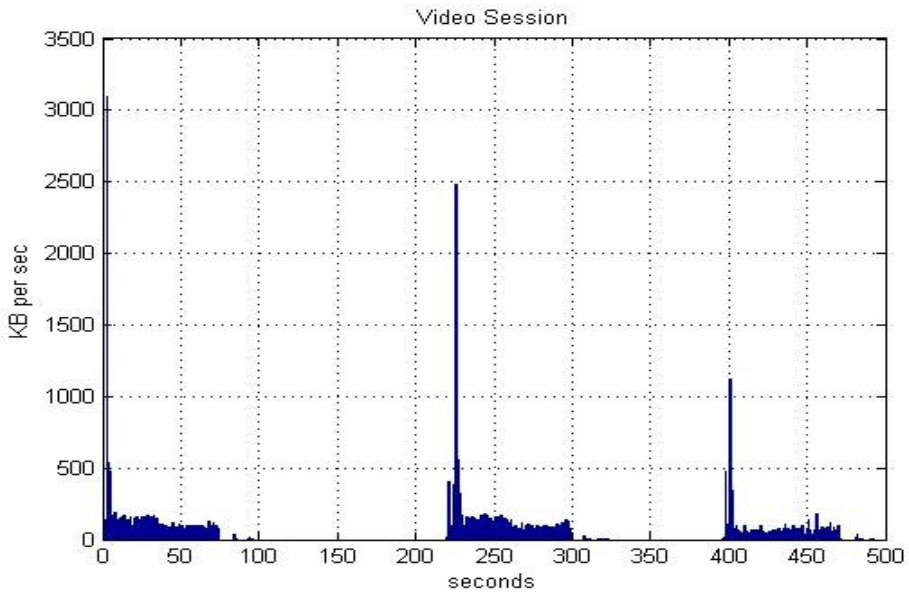


Figure 5.2 Throughput of the video session

Figure 5.2 shows how the throughput of a particular video session varies for a time interval of 500 seconds in a real environment.

The bandwidth is shared fairly using TCPW until the a normal TCP flow is detected. To provide a single numerical measure reflecting fair share distribution across various connections ,use the Jain Fairness Index defined in [29] :

$$\text{FairnessIndex} = \frac{(\sum_{i=1}^{N} b_i^2)}{N\sum_{i=1}^{N} b_i^2}$$

where: b_i is the throughput of the i^{th} connection

N is the number of connections

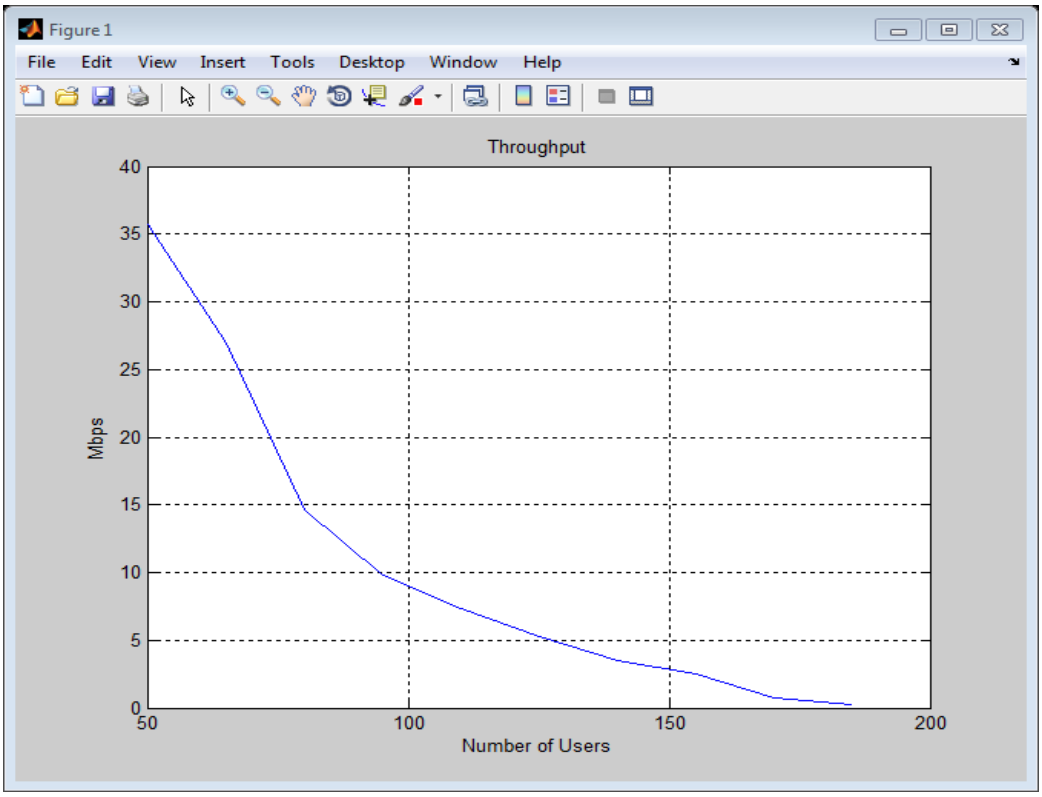


Figure 5.3 Fairness Evaluation

Figure 5.3 shows how the number of users/subscribers increases once the throughput of the video session decreases.

Simulation:

The fairness index is 0.91

Throughput (Mbps) =35.74 26.98 14.64 9.85 7.32 5.28 3.48 2.48 0.75 0.27

Number of user=50:10:200

Observation:

For various average effective decreases in the throughputs an increase in the number of users is observed,that are being served for a particular fairness index.

The graph shows the average throughput received per user. The network bandwidth is constant. If fewer users are present in the network the average throughput per user is more. Similarly when a large number of users are present in the network for the same bandwidth the average throughput per user decreases.

Depending upon the amount congestion, a network will decide which optimization technique needs to be used. When the network is heavily congested and a bottleneck scenario is encountered dynamic optimization is applied.

The bandwidth allocation algorithm deals well with dynamically changing network conditions. Depending upon the congestion, the content is optimized before delivering it to the receiver. Content optimization in turn leads to reduced usage of the bandwidth.

Chapter 6

Compression

6.1 Evolution of video coding standards [32]:

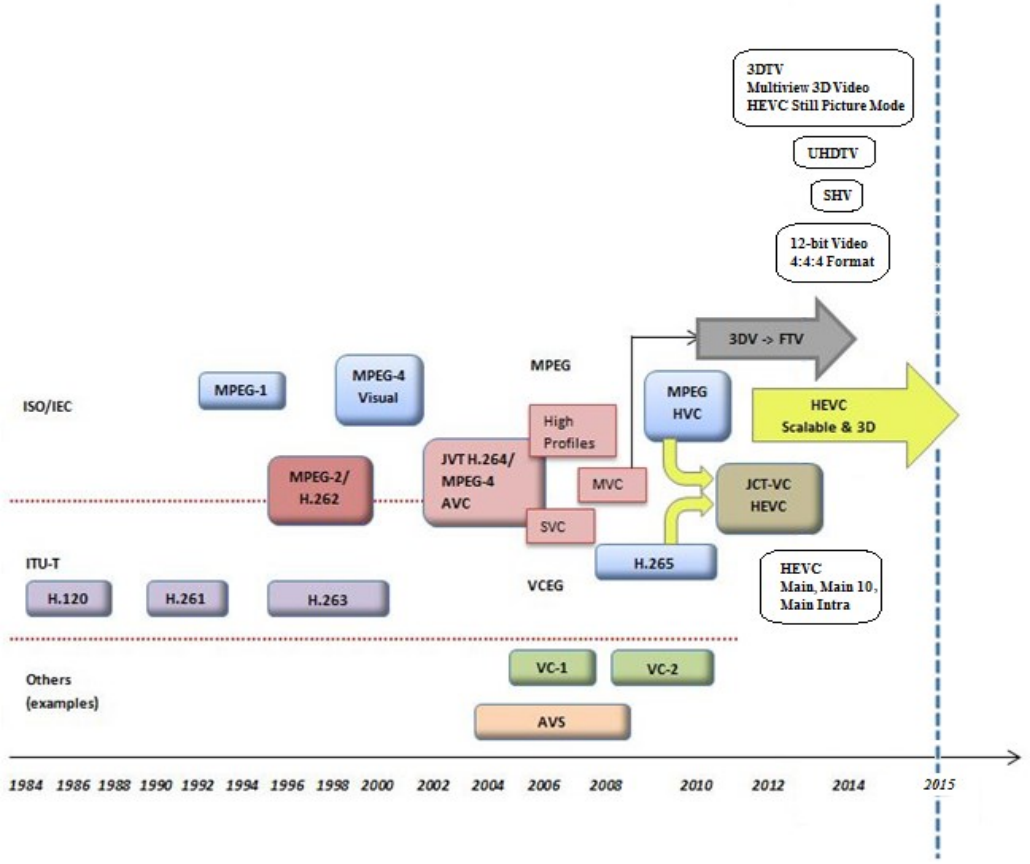


Figure 6.1 Evolution of video coding standards [32]

Major video coding standards have been developed by the International Standardization Organization / International Electro technical Commission (ISO/IEC) and the International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) [32]. Figure 6.1 depicts a historical perspective for video coding standards development since the very first ITU-T H.120. The evolution of H.264/AVC doubled the coding efficiency from that of the MPEG-4 simple profile and has therefore gained wide industrial

acceptance recently [32]. Further extensions of H.264/AVC include high profiles , scalable video coding (SVC) extension , and multi view video coding (MVC) extension [32] .

6.2 Need for video compression- Growing demand for video [30]

Video traffic is growing exponentially as shown in figure 6.2. The mobile users and network operators face a challenge due to the rapid increase of bandwidth usage.

.

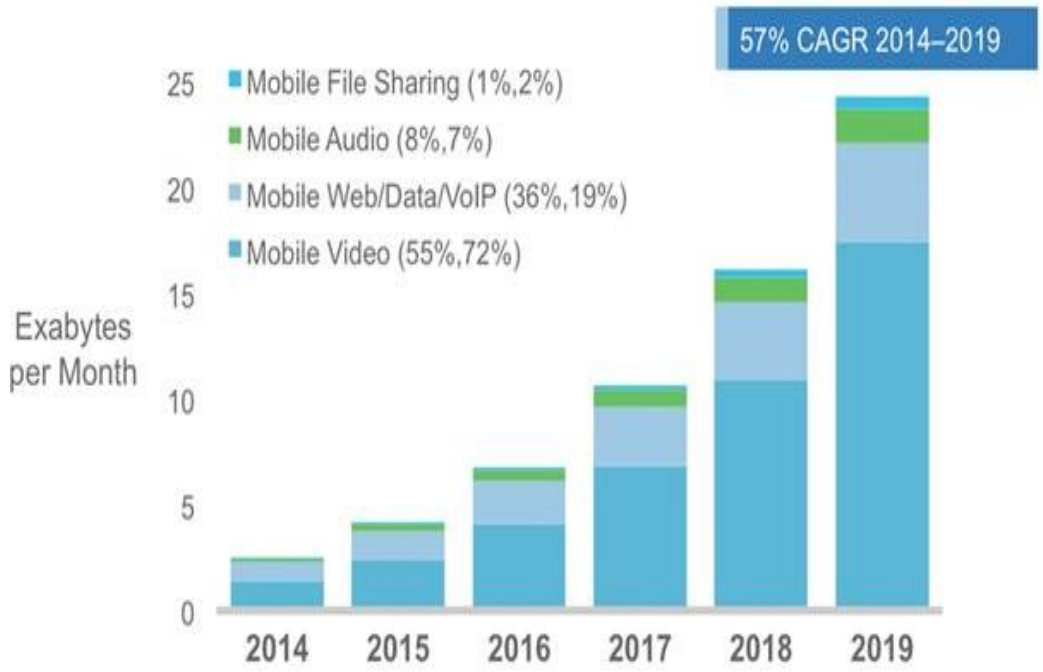
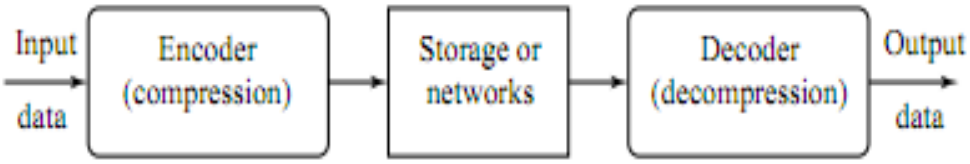


Figure 6.2 Figures in parentheses refer to 2014, 2019 traffic share. Source: Cisco VNI Mobile, 2015

The goal of video compression is to massively reduce the amount of data required to store the digital video file, while retaining the quality of the original video



A General Data Compression Scheme

Figure 6.3 A General Data Compression Scheme

Video and Audio files are very large. Unless we develop and maintain very high bandwidth networks (Gigabytes per second or more) we have to compress the data. In other words, it is better and economically to watch a compressed video rather than an uncompressed video.

For example, a TV picture resolution of 640 ×480 and a frame rate of 15 fps If represented by 3 bytes per pixel ,1 sec of video=13.82 MB and 1 hr of video=49.76GB.In order to reduce the amount of data compression is needed.

The user would expect the video to be played with no stalls. A stall is defined as a pause/freeze while watching the video. It is generally considered unpleasant and reduces the perception of quality of the service

Different techniques can be applied to achieve a balance between the compression ratio (%) and the impact to the Quality of Experience (QoE). The compromise in quality may appear as loss of resolution, pixelization (or 'blockiness') artifacts in fast moving scenes, loss of color depth.

A stall is specifically seen when the buffer at the player/device is not filled. It is experienced by the user as a pause in video playback and "from our surveys and experience" people generally would prefer to watch content without a stall at a slightly

lower quality provided they aren't aware they could get better quality. Or perhaps if better quality is not available.

The subscriber can't really tell the quality has been compromised as they are fine with it and even if they can slightly, they are more likely to get frustrated and give up trying if it stalled frequently

Therefore, the subscriber prefers to lower the quality with less number of pauses.It is reasonable assumption that it is better to have lower quality than pauses.

6.3 Fundamental concepts in video coding

6.3.1 Color Spaces

The common color spaces for digital image and video representation are stated:

RGB Color space – Each pixel is represented by three numbers symbolize the relative proportions of red, green and blue colors.

Y C_r C_b Color space–Y symbolizes the luminance component, a monochrome version of color image. Y is a weighted average of R, G and B:

$Y=k_rR+k_gG+k_bB$, where k are the weighting factors.

The color information is represented as color differences or chrominance components, where each chrominance component is difference between R, G or B and the luminance Y.

As the human visual system is less sensitive to color than the luminance component, the color space has advantages over RGB space. The amount of data required to represent the chrominance component reduces without impairing the visual quality [34].

The different patterns of sub-sampling [34] are:

- 4:4:4 – The three components Y C_r C_b have the same resolution, which is for every 4 luminance samples there are 4 C_r and 4 C_b samples.

- 4:2:2 – For every 4 luminance samples in the horizontal direction, there are 2 C_r and 2 C_b samples. This pattern is used for high quality video color reproduction.
- 4:2:0 – C_r and C_b each have half the horizontal and vertical resolution of Y. This pattern is used in applications such as video conferencing, digital television and DVD storage.

Figure 6.4 4:2:0 sampling [34]

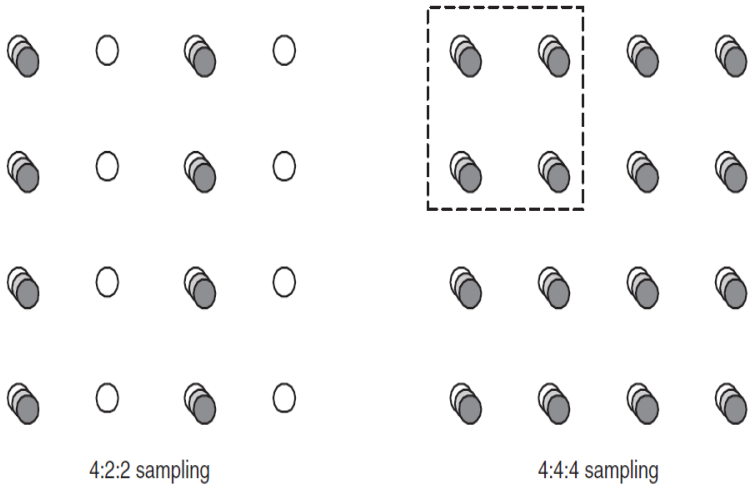


Figure 6.5 4:2:2 and 4:4:4 sub-sampling patterns [34]

6.4 H.264/Advanced video coding introduction

The codec studied in this paper is referred to as H.264. H.264/Advanced Video Coding (AVC) [6] is a video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. It is widely used on in several domains and emerging coding standard.The main goals of the H.264/AVC standard are to enhance compression performance and make provision for a network-friendly video representation addressing conversational (video telephony) and non-conversational (storage, broadcast, or streaming) applications [6].

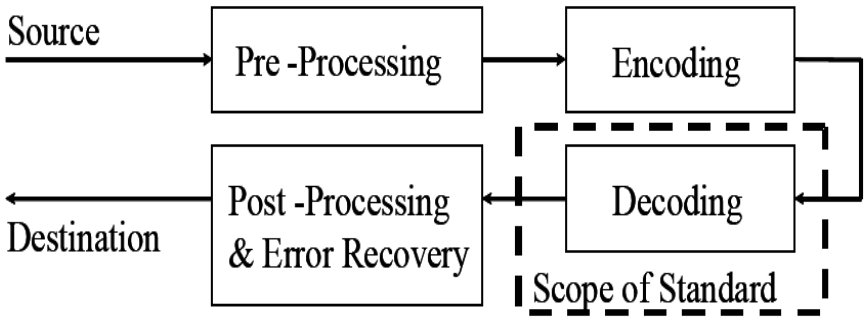


Figure 6.6 Scope of video coding standardization

In regard to Figure 6.6, only the syntax and decoder are standardized. This approach permits optimization beyond the obvious and complexity reduction for implement ability.

The standardization of the first version of H.264/AVC was completed in May 2003. In the first project to extend the original standard, the JVT then developed what was called the Fidelity Range Extensions (FRExt). These extensions enabled higher quality video coding by supporting increased sample bit depth precision and higher-resolution color information, including sampling structures known as YC_bC_r 4:2:2 and YC_bC_r 4:4:4. Several other features were also included in the Fidelity Range Extensions project, such as adaptive switching between 4×4 and 8×8 integer transforms, encoder-specified perceptual-based quantization weighting matrices, efficient inter-picture lossless coding, and support of additional color spaces. The design work on the Fidelity Range Extensions was completed in July 2004, and the drafting work on them was completed in September 2004 [33]

Further recent extensions of the standard then included adding five other new profiles intended primarily for professional applications, adding extended-gamut color space support, defining additional aspect ratio indicators, defining two additional types of "supplemental enhancement information" (post-filter hint and tone mapping), and

deprecating one of the prior FRExt profiles that industry feedback indicated should have been designed differently.

6.5 Encoder and decoder in H.264

An H.264 video encoder carries out prediction, transform and encoding processes (Figure 6.7) to produce a compressed H.264 bit stream. An H.264 video decoder carries out complementary processes of decoding, inverse transform and reconstruction to produce a decoded video sequence [7].

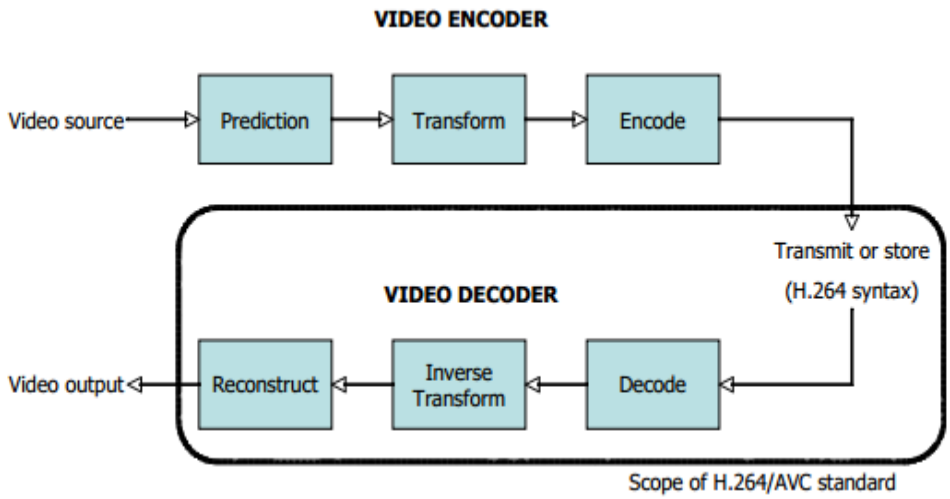


Figure 6.7 Block Diagram of H.264 Codec [7]

The H.264 consists of two conceptual layers, the Video Coding Layer (VCL) that represents the efficient representation of video, and Network Abstraction Layer (NAL) that converts VCL representation into a format suitable for transport layers such as packet stream transport, bit stream format, etc.

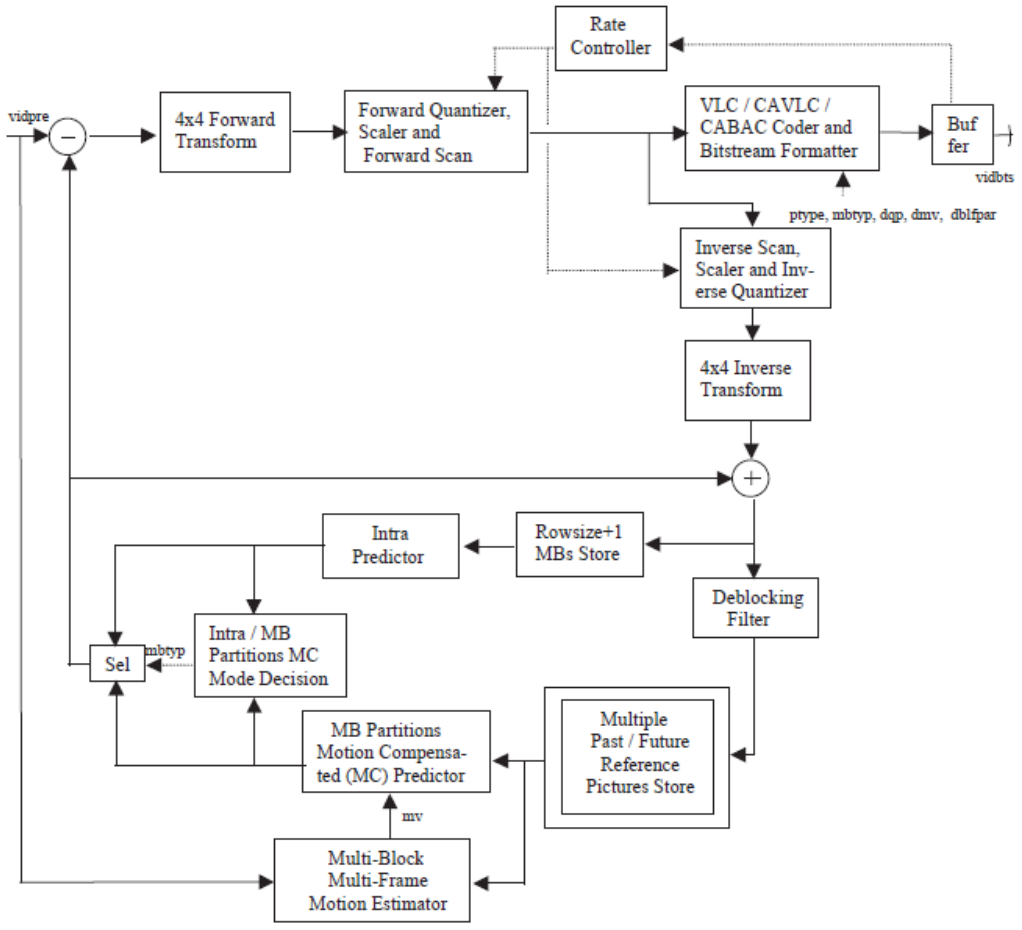


Figure 6.8 Detailed Block Diagram of H.264 CODEC [7]

In Figure 6.8, the picture is split into macroblocks. The first picture of the raw yuv sequence is intra-coded. This means it uses the information incorporated within the picture itself. Spatially neighboring samples of the previously coded blocks are used to predict each sample of a block with an intra-frame.

The image is split into one or more slices, which are sequences of micro blocks. The slices are independent because their syntax can be transmitted from the bit stream if the original sequence and parameter sets are given. Each slice group is further split into multiple slices, which consists sequences of macroblocks.

The input F_n shown in figure 6.7 in the forward path is used for encoding. By making use of the reconstruction frame in different modes like intra and inter for prediction P a prediction macroblock is formed.

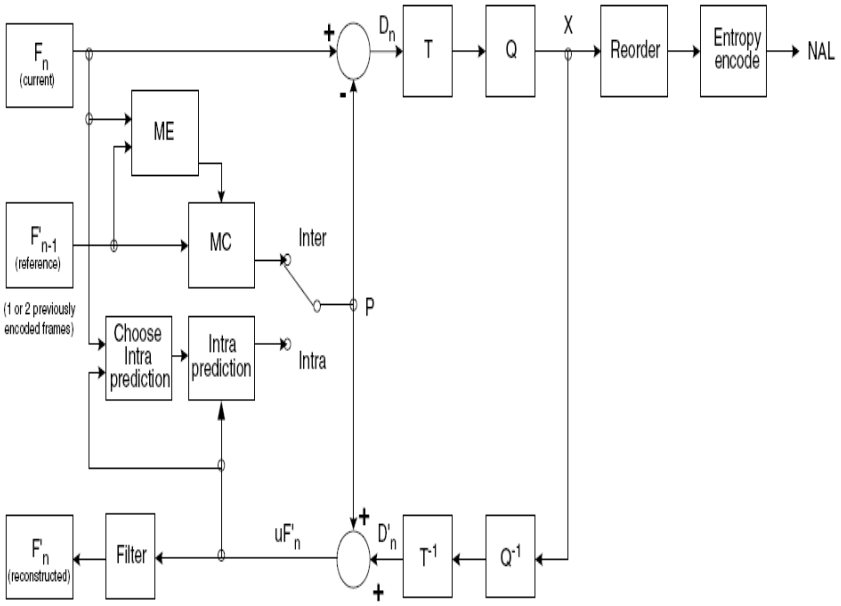


Figure 6.9 Block Diagram of H.264 Encoder [35]

The encoder processes a frame of video in units of a macro-block (16x16 displayed pixels) [7] as shown in Figure 6.9 .

It forms a prediction of the macro-block based on previously-coded data, either from the current frame (intra prediction) or from other frames that have already been coded and transmitted (inter prediction)

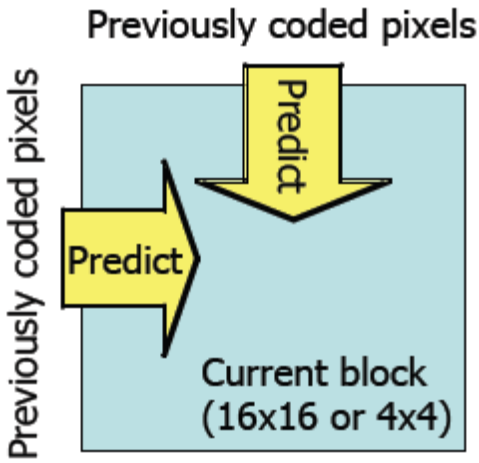


Figure 6.10 Intra prediction in H.264[7]

A range of block sizes (from 16x16 down to 4x4) to predict pixels in the current frame from similar regions in previously coded frames is used by inter prediction (Figure 6.11)

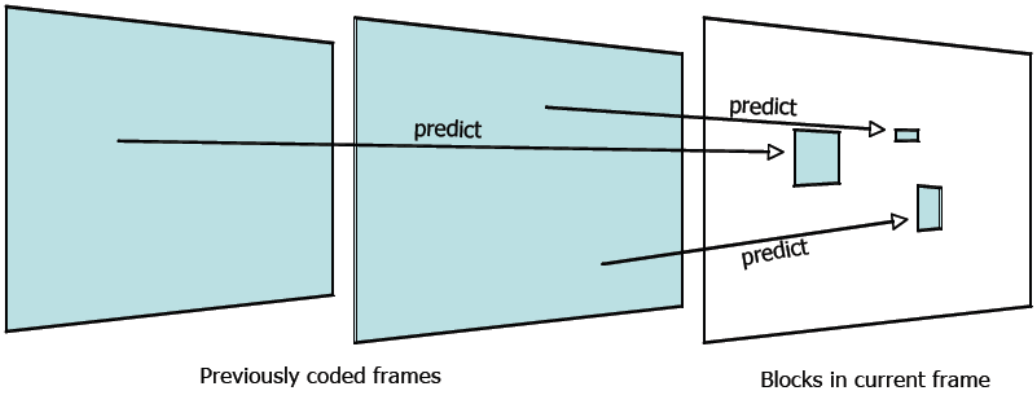


Figure 6.11 Inter prediction in H.264 [7]

Motion estimation is elucidated by calculating the suitable inter prediction. Motion compensation is described by subtracting an inter prediction from the current macro-block.

The next step is transformation and quantization of D_n , which causes the a set of X quantized transforms to be generated. Furthermore, the formulation of the bit stream is done which consists of the macroblock along with the side information like the quantizer size, prediction mode, etc.

The transformation makes use of a block of residual samples. The residual samples are transformed with help of a 4x4 or 8x8 integer transform an approximate form of the Discrete Cosine Transform (DCT) [36]

The output of the transform, is quantized, i.e., each co-efficient is divided by an integer value. Quantization reduces the data for the coefficients according to a quantization parameter (QP).The transmission occurs at the Network Abstraction Layer (NAL) as stated in [16] which is used for storage or transmission.

The compressed bit stream is sent to the decoder from the NAL as shown in Figure 6.12.

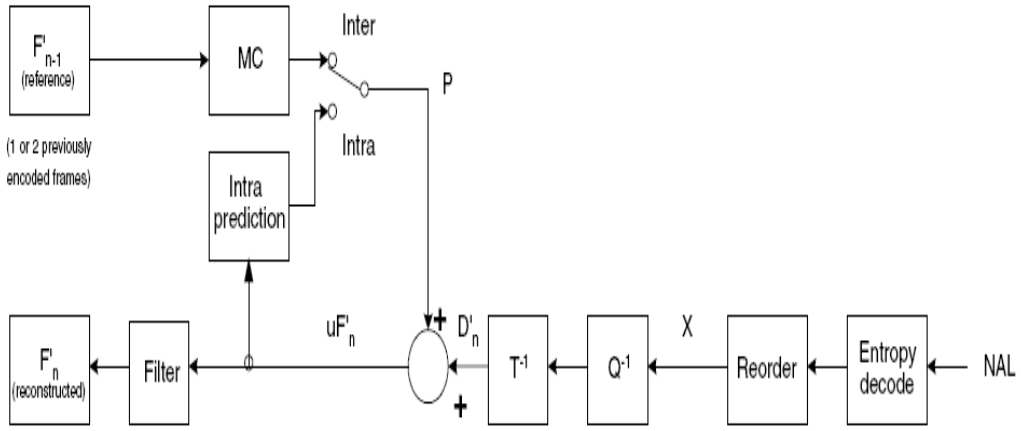


Figure 6.12 Block Diagram of H.264 Decoder [35]

The quantized coefficients are produced by entropy decoding of the data elements and re-ordering. The inverse transformation and rescaling D_n leads to the identical encoder D_n .

Compression is applied to reduce the redundancy and irrelevancy. The redundancy can be due to two main reasons as stated:

- Spatial: The nearby pixels are correlated
- Temporal: The adjacent frames are highly correlated

Irrelevancy arises due to perceptually unimportant information

The algorithm used in this thesis is a quantitative model which includes the relationship between the Quantization parameter(QP) and the available bandwidth over the network. QP is initialized manually upon the start of the video sequence. Normally, a small initial QPo is chosen if the available bandwidth is wide and the packet loss is less and a big QPo should be used if it is narrow and packet loss is more.

The QP taken into consideration influences the detail of information carried in the transformed Group of Pictures. As the bit rate is determined the encoder encodes the GOP using the target bit rate. The determination of the bit rate for every GOP makes the codec change the resolution.

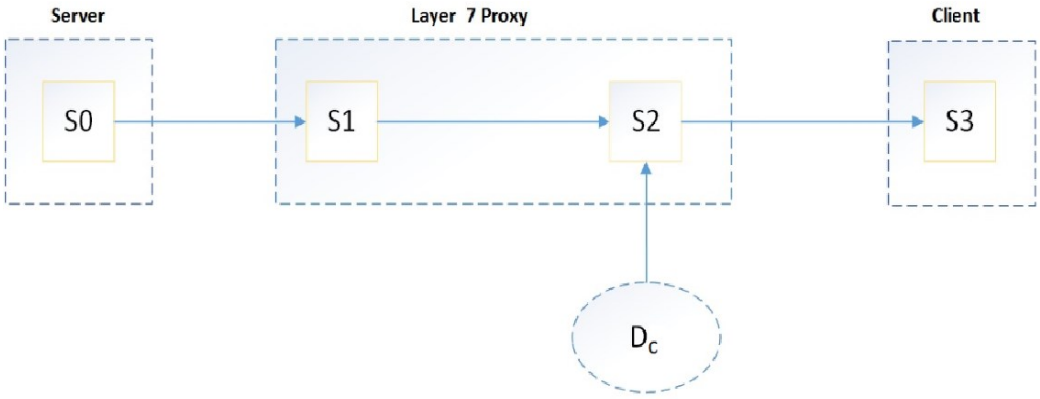


Figure 6.13 Generic model for selecting the optimal compression rate D_c

As seen in Figure 6.13, the video proxy(Layer 7 proxy) acts as an intelligent ,content aware intermediary between the wireless network and the internet. The proxy receives the video frames, denoted as S0, from a remote server. It decompresses the video frames to extract the raw video denoted as S1. Then the proxy compresses the raw video again to S2 with an optimal compression rate. The video client eventually receives frames S3 with distortion due to packet loss/congestion in the path from the proxy to the client.

The streaming rate as in [22] after the compression in the video proxy, Dc, is the key parameter in the whole system. Two different aspects determines the optimal value of Dc. Firstly, the ratio Dc /D, where D is the raw data transmission rate, indicates the information loss due to the compression in the proxy if lossy compression is applied. If lossless compression takes place there is no information loss. If this ratio is too low, the received video quality at the client is inevitably poor even if there are no losses in the network. Second, since the transmission inside the wireless network cannot be error free, the current network condition, such as any congestion in the path, will influence Dc. Under poor network condition, a smaller value of Dc is preferred. On the other hand,

when the network condition is good, the video can be streamed at a higher rate. Therefore, there is a tradeoff in selecting the value of Dc.

The adaptive optimization algorithm adjusts the compression level applied based on the fluctuating channel throughput. Basically when traffic grows past the point where 'no optimization' would start to result in stalls, but no longer fixes anything once traffic exceeds the point when full optimization also fails to prevent stalls. In between those upper/lower bandwidth requirements the L7 solution can help to improve overall customer experience.

6.6 Compression algorithm

The optimal encoding parameters set are defined such that it maximizes the overall user satisfaction, subject to several constraints imposed by both the delivery system and the service provider from [13]

Let V denote be the set of videos. Each video $v \in V$ can be encoded in different representations, each one characterized by the encoding rate $r \in R$ and the spatial resolution $s \in S$, .R and S are the sets of bit rates and spatial resolutions used to generate the representations. In the model then the triple (v, r, s) corresponds to the representation of a video $v \in V$ encoded at a resolution $s \in S$ and at a bit rate $r \in R$. Each resolution s admits encoding rates within the range $[b_{vs}^{min}, b_{vs}^{max}]$ for video v. More precisely, the user as an index in the set of rates and use br for the actual value (in bits per second) of the encoding rate.

Let U be the number of users that the CDN network can serve, where each user $u \in U$ requests a video channel $v_u \in V$ at a given resolution $s_u \in S$ by means of an Internet connection with a capacity of c_u bits per second. It is can also be assumed that each user is associated with one single video resolution at a time. For example a user

streaming a 640x480 resolution and over time one subscriber may do a small screen on a web page (e.g176x144) and later in the day watch a movie (352x288).

An arbitrary user watching video v at resolution s experiences a satisfaction level of $f_{vs}(r)$, which is an increasing function of the bit rate r, ranging from 0 to 1(i.e. as a percentage of satisfaction ,100% being perfectly satisfied).The satisfaction function is different for every resolution. For example, for a user watching a video v at resolution s, $f_{vs}(r) = 1$ if $b_r = b_{vs}^{max}$, but the same rate might lead to a satisfaction less than 1 for the same video content but displayed at a higher resolution. The satisfaction level is denoted by f_{vrs} rather than $f_{vs}(r)$.

$f_{vrs} \in R^+$ =Satisfaction level for the representation encoded at rate r and resolution s of the video v

$b_r \in R^+$ =Value in kbps of the encoding rate r

$b_{vs}^{min} \in R^+$ = Value in kbps of the minimum encoding rate that the video v at resolution s can admit.

$b_{vs}^{max} \in R^+$ =Value in kbps of the maximum encoding rate that the video v at resolution s can admit.

$c_u \in R^+$ =Maximum Internet connection capacity in kbps of user u

$v_u \in V$ =Video channel requested by user u

$s_u \in S$ =Spatial resolution requested by user u

$C \in$ R+= Total network capacity in kbps

$$\max_{\{\alpha,\beta\}} \sum_{u\varepsilon U} \sum_{v\varepsilon V} \sum_{r(x)\varepsilon R} \sum_{s\varepsilon S} f_{vrs}\alpha_{uvrs} \qquad (3a)$$

The objective function (8a) maximizes the overall user satisfaction. It is taken as the objective function because it considers customer satisfaction for a particular video resolution. Basically trying to sum the user satisfaction for all the users at the particular video level.

s.t

$$\alpha_{uvrs} \leq \beta_{vrs}, u\varepsilon U, v\varepsilon V, r\varepsilon R, s\varepsilon S \qquad (3b)$$

$$\beta_{vrs} \leq \sum_{u\varepsilon U} \alpha_{uvrs}, v\varepsilon V, r\varepsilon R, s\varepsilon S \qquad (3c)$$

The constraints (3b) and (3c) set up a consistent relation between the decision variables α and β . α is the set of all users watching a video and α_{uvrs} represents the set of the users at a particular bandwidth level The number of users watching have to be using bandwidth).No user can have all the bandwidth. β represents the bandwidth.β_{vrs} represents total bandwidth used which is less than the sum of all users/subscribers bandwidth (The amount of bandwidth used might include "shared videos).The total bandwidth is less than the sum of the bandwidth to the users. In conclusion, the number of subscribers watching that video "fit" into that bandwidth (e.g. the number of watchers "alpha" are less or equal to the bandwidth available to those same viewers)

$$(b^{\min}_{vs} - b_r).\beta_{vrs} \leq 0, v\varepsilon V, r\varepsilon R, s\varepsilon S \quad (3d)$$

$$(b^{\max}_{vs} - b_r).\beta_{vrs} \leq 0, v\varepsilon V, r\varepsilon R, s\varepsilon S \quad (3e)$$

The constraints (3d) and (3e) force to zero some variables. They ensure that each video v at resolution s is encoded only at the bit rates in the range between the minimal and maximal admissible rates for the video v at resolution s depending on the congestion. Also (3d) states the number of subscribers that are watching the MINIMUM rate video as

compared to the bandwidth available is "less than or equal to 0“ (which forces it basically to zero, since it is in zero and R+ mostly).(3e) states the number of subscribers that are watching above the MAXIMUM rate video as compared to the bandwidth available is "less than or equal to 0“ (which forces it basically to zero, since we are in R+ mostly)

$$\sum_{r(x)\varepsilon R} \alpha_{uvrs} \leq \{1, \text{if } v=v_u \ \& \ s=s_u, u\varepsilon U, v\varepsilon V, s\varepsilon S; 0, \text{otherwise} \quad (3f)$$

The constraint (8f) ensures that, user u is served and receives the requested video stream v with the correct resolution s_u

The sum of "alpha" is the number of subscribers at that resolution . α_{uvrs} is either 0 or 1 (strictly) as "beta' is also 0 or 1.So these are constraints on the number of subscribers in each "situation.(If you are watching at the given resolution, you get counted, and not otherwise)(Zero indicates not used and one indicates used)

$$\sum_{v\varepsilon V} \sum_{r(x)\varepsilon R} \sum_{s\varepsilon S} b_r \alpha_{uvrs} \leq c_u, u\varepsilon U \qquad (3g)$$

$$\sum_{u\varepsilon U} \sum_{v\varepsilon V} \sum_{r(x)\varepsilon R} \sum_{s\varepsilon S} b_r . \alpha_{uvrs} \leq C \qquad (3h)$$

The constraints (3g) and (3h) respectively limit the user link capacity and the overall network (CDN) capacity.

The number of users showing videos at a given bandwidth should be less than the capacity for such users. (Total users at a given bandwidth usage watching a show can't exceed the capacity allocated at that level) as shown in (3g)

Total capacity is the sum of all such users.cu represents the capacity for users at a given bandwidth and C represents the total capacity.(total usage can't exceed total capacity) as shown in (3h)

$$\alpha_{uvrs}[0,1], u\varepsilon U, v\varepsilon V, r\varepsilon R, s\varepsilon S \quad (3i)$$

$$\beta_{vrs}[0,1], v\varepsilon V, r\varepsilon R, s\varepsilon S \quad (3j)$$

The constraints (3i) and (3j) indicate the values of α and β range between 0 and 1.

The amount of compression can also be calculate by the buffer size as below:

$$\phi_1 = | D_i | \quad (4)$$

$$\phi_{i+1} = \phi_i + | D_i | \quad (5)$$

Where D_i is equal to the difference between the received data and the displayed data.

The buffer size increases when the received data is more than the displayed data. The buffer size of the next interval is the buffer size of the current interval plus the received data difference

The sum of the absolute value of the buffer changes to provide a metric of how "tight" the compression is to the displayed data. The closer the received data (based on the compression) to the data displayed it's better. If the displayed data is lot less than the received, then the compression ratio is too high.

Every compression algorithm has its limit, it cannot get more benefits when the video stream rate is higher than a certain value, and it's a waste of network bandwidth if the video compression ratio is higher than the threshold.

$$D_c \leq D*H \quad (6)$$

H represents the threshold as in [22]

The key here is that it is not a "waste of network bandwidth when compression is applied. It is a chance to "maximize" the subscriber satisfaction since "higher quality video" is shown at the point where the network bandwidth allows it. The amount of compression

shouldn’t exceed the threshold limit. The Peak –Signal-To-Noise Ratio (PSNR) depends on the video content along with the compression ratio. When the compression ratio is higher than a particular value i.e., the threshold then the PSNR doesn’t improve.

Chapter 7

Convex Optimization

7.1 Stall optimization algorithm

The number and the duration of the playout stalls are used as the primary metrics for measuring the QOE of a video. A stall is specifically seen when the buffer at the player/device is not filled. It is experienced by the user as a pause in video playback and "from our surveys and experience" people generally would prefer to watch content without a stall at a slightly lower quality provided they aren't aware they could get better quality. The QOE is a subjective experience measures.

The playout stall is impacted strongly by the buffer management. The buffer size of a client is defined as the amount of data that is received by the client but not yet played out.

If the buffer is not having data which is worth to be streamed or to be played for one second a stall occurs. Depending on the resolution of the video the amount of the data which is required to stream a video for a duration of 1sec is the data worth of 1sec as shown in Figure 7.1 in the later part of the chapter.

Consider maximizing the weighted proportional fairness given [10] as

$g(r,z)=\sum_{i=1}^{n}\phi_i(z_i)\log(r_i)$ (7.1) where r_i is the TCP (receiver advertised) window size for flow i. The objective function (7.1) uses a logarithmic function (the law of diminishing utility returns). It is used as a "fairness" function that is designed to keep the display from stalling by checking the size of the buffer and keeping it above zero. This is constrained by the available bandwidth for the subscriber.

The weight $\phi_i(z_i)$ is computed as stated. At any time t, let D_i be the difference between the amount of the bytes streamed and the amount of bytes that needs to be sent

by the time t for an uninterrupted playout for a video flow 'i'. $\phi_i(z_i)$ is calculated in an alternative way by multiplying the original data p_i times the compression rate z_i. If there are n flows in the progress the inverse of the normalized buffer side for flow I is given by:

$$\phi_i(z_i) = \frac{\sum_{i=1}^{n} p_i z_i}{p_i z_i} \quad (7.2)$$

The optimal vector $r*$ is found which maximizes the QOE objective function with respect to the capacity constraints.

Window size variables: $r_i, \forall i$ (7.3a)

Capacity constraints: b, ϕ_i, a_i, $\forall i$ (known). b_i, $\forall i$ (Unknown) (7.3b)

The constraint (7.3b) states some parts of the bandwidth are "known" (in this case the total) and some that are not (in this case, the amount of bandwidth each subscriber is taking) a_i

represents the inverse of the round trip time. The instantaneous rate of the TCP flow is window size (in bytes) divided by the round trip time (in seconds)

Shared link constraint: $\sum_{i=1}^{n} a_i r_i \leq b$ (7.3c)

The constraint (7.3c) adds up all the bandwidth used, and makes sure it is less than the total bandwidth. It constraints the bandwidth to the user. D_i<b

Di be the difference between the amount of bytes streamed and the amount of bytes that need to be sent by time t for an uninterrupted playout of a video flow i, as given by the playback curve.

If Di < 1, we set Di to a minimum value of 1.

Normally D_i would be a large number (e.g. hundreds or thousands of bytes), but to prevent divide by zero problems they limit it down to "1" as the smallest value)

If there are n flows as stated above, the inverse of normalized buffer size for flow i is given by !i = !n

i=1 Di in equation 7.2 represents the inverse of the normalized buffer size for flow i. The percent of the buffer is “Di / (total D_i) “which is then inverted to "total D_i / D_i.Hence, it represents the percentage of the total buffer space. The smaller the value of D_i, that it needs more data. the bigger the $\phi_i(z_i)$ function.

Bottleneck constraint: $a_i r_i \leq b_i \quad , \forall i$ (7.3d)

The constraint (7.3d) limits the total bandwidth. It checks to make sure the single subscriber doesn’t exceed its bandwidth limits (or the amount of bandwidth it could receive given noise, etc). Another way to state is equation (7.3c) is the limit for the system and 7.3d) is the limit for the individual.

The optimization problem cannot be solved directly as stated in [10],the values of b_i s are unknown. In order to solve this problem a window vector $\bar{r}$ is needed, and through a series of window vector modifications arrive at an optimal window vector $\bar{r}^*$ where each element r_i corresponds to the desired TCP receiver window size for i.

The video session time is divided in terms of epochs and an epoch is further divided into sub epochs. In order to estimate the buffer sizes of the video clients , measure the amount of data streamed to the user in an epoch and compare it with the playback curve if the corresponding video. The difference between the two quantities gives an estimate of the buffer size at the video client and thus the weight ϕ_i for flow I in the subsequent epoch.

By analysis it is observed in order for the network to be stable it takes 2 to 4 RTTs for a TCP flow to show the effect of a window modification and converge to a rate.

7.2 Membership queries

At a theoretical level [10],the shared and bottleneck link constraints in the problem formulation are half spaces in the n dimensional space R^{n} of window vector values. The goal is to find an optimal point given by a window vector $\overline{p}^{*} = \{p_1^{*}, p_2^{*}, ..., p_n^{*})$

One of the half spaces of the convex polytope are known (since b is known).However, the remaining half spaces are unknown (since all b_{i}s are not known).In order to find the optimal point $\overline{p}^{*}$, the convex optimization algorithm needs to run 'X' iterations and in every iteration a window vector $\overline{p}$ is proposed and get $\overline{p^{sys}}$ as a feedback from the system as to whether the point is inside or outside the polytope.

The algorithm works for single and multiple flows. For a single flow 1,the window size value is set to p_1 as the upper limit of $\mathrm{a}_1 p_1$ on the allowed TCP rate of flow 1.The rate of a TCP flow is expressed as window in bytes divided by the round trip time in seconds as stated previously..Each time we set p_1 ,we check whether $\mathrm{a}_1 p_1 > \mathrm{b}_1$.The window size that corresponds to the actual amount of data transmitted is selected as the system feedback $\mathrm{x}_1^{\mathrm{sys}}$ .If $\mathrm{a}_1 p_1 > \mathrm{b}_1$,i.e., a higher flow rate compared to the capacity of its bottleneck link is allowed. The data experiences packet loss or substantial delay.

Due to this number of acknowledgments are lower than expected since the network is under congestion. If less or equal amount of data to what the bottleneck link can handle is sent, acknowledgments are received for almost all the sent packets. This

would allow p_i to be increased further. Now consider two flows 1 and 2 simultaneously. The window size is set for the flow 1 to p_1 and flow 2 to p_2, to test whether $a_1 p_1 > b_1$ and $a_2 p_2 > b_2$, also the allowed flow rates shouldn't exceed the capacity of the shared link b, $a_1 p_1 + a_2 p_2 \leq b$. If $a_1 p_1 + a_2 p_2 > b$ then the system feedback might get dominated by the shared link constraint.

Hence, the shared bottleneck constraint is never violated when the window vector is set to the window values for all the flows. But it may still violate the constraints; in this case the binary search algorithm is used to converge to an optimal window vector.

Consider N which denotes the number of flows {1,2…….,n}

The optimal solution to the problem with the objective function $\sum_{i=1}^{n} \phi_i(z_i) \log(r_i)$ with the constraint $\sum_{i=1}^{n} a_i p_i \leq b$ is solved below:

Maximizing $\sum_{i=1}^{n} \phi_i(z_i) \log(r_i)$ is equivalent to minimizing $\sum_{i=1}^{n} -\phi_i(z_i) \log(r_i)$ .The optimization problem is convex since the constraint mentioned is a linear inequality and the objective function is also a sum of convex functions, therefore it is convex.

In mathematical optimization, the Karush–Kuhn–Tucker (KKT) conditions (also known as the Kuhn–Tucker conditions) are first order necessary conditions for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied

By solving using the Karush–Kuhn–Tucker(KKT) conditions as stated in [10] the global optimal solutions are found shown below:

$$g(r_1, r_2, ..., r_n) = \sum_{i=1}^{n} -\phi_i(z_i) \log(r_i) + \lambda \sum_{i=1}^{n} a_i r_i - b \quad (7.4)$$

$$\frac{\partial g}{\partial \mathrm{x_i}} = -\frac{\Phi_{\mathrm{i}}(z_i)}{\mathrm{x_i}} + \lambda \mathrm{a_i} = 0 \forall i \tag{7.5}$$

$$\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{a_i} r_{\mathrm{i}} = \mathrm{b} \tag{7.6}$$

Solving the above equations we get the point that gives the optimal solution:

$$\begin{aligned} &\frac{\Phi_{\mathrm{i}}(z_i)}{\frac{r_{\mathrm{i}}}{\sum_{\mathrm{i}} \Phi_{\mathrm{i}}(z_i)} * \mathrm{b}}, \forall i \\ &\lambda = \frac{\sum_{\mathrm{i}} \Phi_{\mathrm{i}}(z_i)}{\mathrm{b}} \end{aligned} \tag{7.7}$$

Equation (7.7) is the end solution.The main idea is that a practical system needs to implement these equations to determine the action.

Chapter 8

Rate Distortion

8.1 Introduction

Rate control plays a significant role in video coding [37]. In video communications, rate control must ensure the coded bit stream is transmitted successfully to make complete usage of the available limited bandwidth.

A rate control algorithm dynamically adjusts encoder parameters to achieve a target bitrate. The algorithm also selects whether to use a fixed or variable rate for the output stream. Different frames will have significantly different sizes depending on the transmission rate. The bits of the consumption of different frame will be significantly different if the coding parameters remain unchanged during the compression process.

The rate control algorithm allocates a budget of bits to each group of pictures, individual picture and/or sub-picture in a video sequence. Rate control is not a part of the H.264 standard, but the standards group has issued non-normative guidance to aid in implementation. The H.264 families of the encoding schemes are inherently lossy processes, which means small quality compromises are made in addition with the reduction of data as part of the compression process.

The key parameter affecting the spatial detail in H.264 is the quantization parameter. When the value of QP is small , all the detail of the video is retained and when it is increased some of the detail gets reduced. This, in turn results in a reduction of the bit rate. Varying the QP, therefore varies the bit rate. Hence, developing a direct relationship between QP and the bit rate is useful in making the codec adaptable. By reducing the bit rate distortion increases and some there is some loss of quality. On the other side, an increase in the bit rate decreases the distortion with a better quality video.

Figure 8.1 suggests the relationship for a picture, lower bit rate is achieved by lowering QP at a cost of increased distortion.

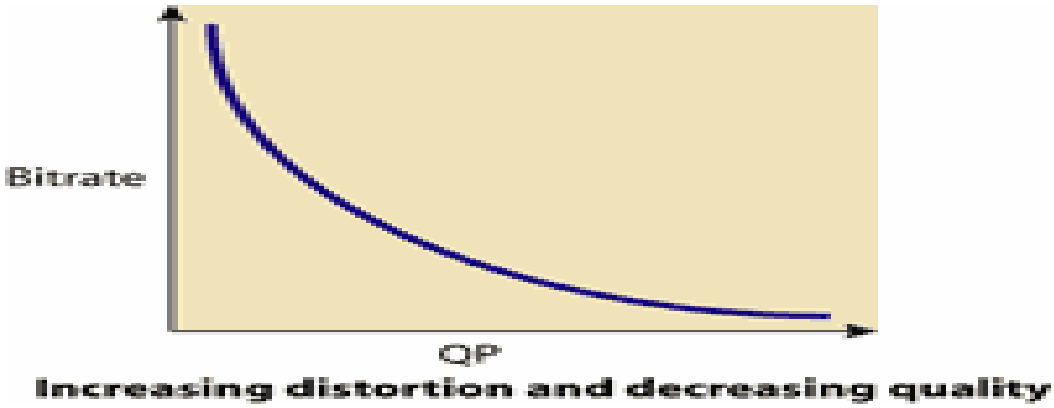


Figure 8.1 For a particular source frame [38]

. The network bandwidth is unpredictable and changes during the session. In general, the distortion D is minimized for a particular video session, expressed as:

$$\min D: R<R_c \qquad (8.1)$$

Also R (D) should be equal to the minimum mutual information between the source T and the reconstruction T^ given by:

$$R(D)=\min_{P_{T^/T:E[d(T,T^)]\leq D}} I(T;T^) \qquad (8.2)$$

The traditional approach for rate adaptation of the codec have been bit stream switching, which requires storing different versions of the video that are pre-coded to have coding efficiency, complexity penalties and are not well suited to the varying network conditions over which the video is being carried. In order to be efficient in varying network conditions I am considering the quantitative algorithm in [16] which includes a relationship between Quantization Parameter [QP] and the available bandwidth. QP influences the detail of information carried in the transformed Group of Pictures. The QP value ranges between 0 to 51.

QP is initialized manually on the start of the video sequence. Generally, a small initial QP is chosen if the available channel bandwidth is wide and the packet loss is small and a big QP when its narrow and packet loss is higher.

Every time a new I(intra)-frame is being coded, the target bit rate and the bits-per-pixel (bpp) indicator is calculated according to the frame size and frame rate. The initial QP stated previously is calculated in a way where its value is close to the bpp indicator. Also, the bpp value is stored each time a group of picture terminates, right before starting with the following I frame of the new group of picture. Logically distinct from the image compression, the distortion of the reconstructed image should be as minimum as possible. Mainly, it is the issue of the relationship between the encoding rate and distortion.

The bit rate is determined when the encoder encodes the GOP using the target bit rate. The determination of the bit rate for every GOP makes the codec change the resolution of the video. At a constant bit rate the packet losses and the change in bandwidth causes the video to stall. In order to reduce the stalls which is the main goal of this thesis, the encoder parameters need to change dynamically to achieve the target bit rate.

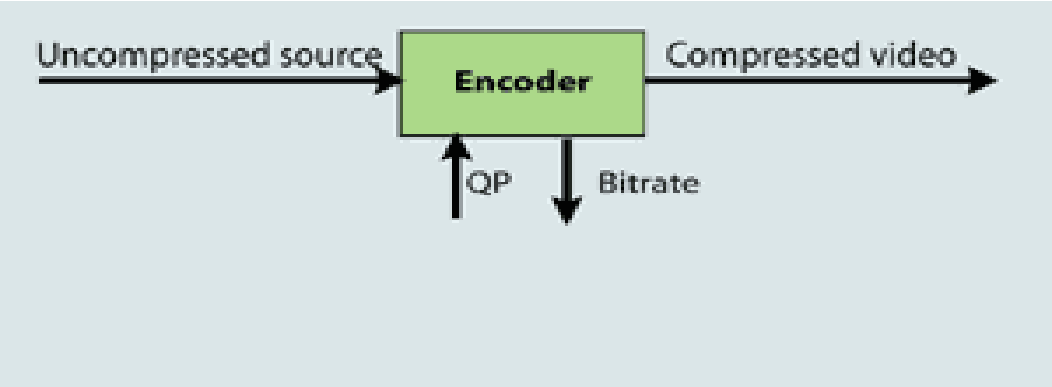


Figure 8.2 Open Loop Encoding (VBR) [38]

Figure 8.2 illustrates open loop (or Variable Bit Rate) operation of a video encoder. The user provides two key inputs – the uncompressed video source and a value for QP. As the source sequence progresses, compressed video of fairly constant quality is achieved, but the bitrate may vary dramatically. Because the complexity of pictures continuously changes in a real video sequence, the value of QP is not easily decided. If the value of QP is fixed for an easy part of the sequence having slow motion and uniform areas, then the bit rate will go up dramatically when the hard(i.e. more complex) parts.

In reality, constraints imposed by decoder buffer size and network bandwidth force us to encode video at a more nearly constant bitrate. To do this, Figure 8.3 illustrates closed loop (or Constant Bit Rate) suggests that the value of QP is varied dynamically based upon estimates of the source complexity, so that each picture (or group of pictures) gets an appropriate allocation of bits to work with. Instead of specifying QP as input, the user specifies demanded bitrate instead.

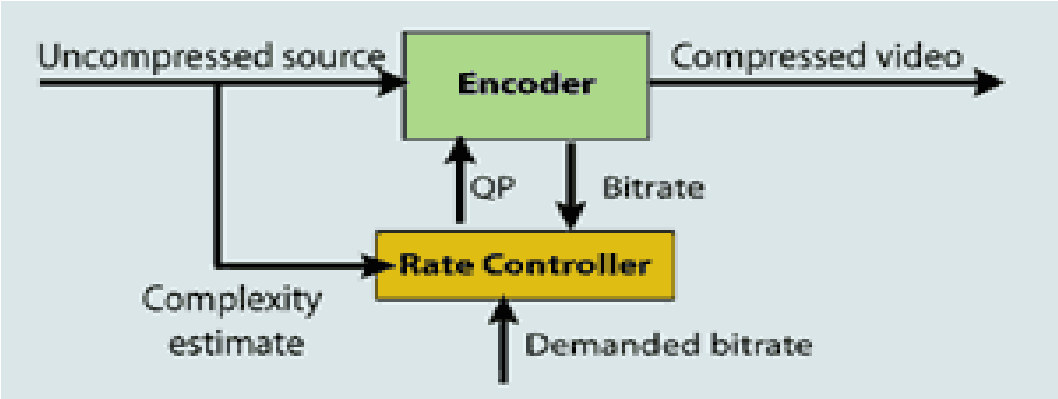


Figure 8.3 Closed Loop Rate Control (CBR) [38]

In the thesis I am making use of the H.264 rate control algorithm to achieve the target bit rate that will reduce the number stalls. Comparison is done on how the distortion level varies for both constant bit rate (fixed QP) and variable bit rate (variable QP) cases.

The heart of the algorithm is a quantitative model as stated previously. The quantization parameter influences only the detail of information carried in the transformed residuals. The bit rate is calculated by:

Bit rate=average bits per pixel/frames per second

8.2 Comparison metrics

8.2.1 Peak signal to noise ratio:

Peak signal-to-noise ratio (PSNR) [12] [13] is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. Because many signals have a very wide dynamic range (ratio between the largest and smallest possible values of a changeable quantity), the PSNR is usually expressed in terms of the logarithmic decibel scale.

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs. The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not. One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.

PSNR is defined via the mean squared error (MSE). Given a noise-free m x n monochrome image, I, and its noisy approximation, K, MSE is defined as:

$$\mathrm{MSE}=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2 \qquad (8.3)$$

The PSNR is defined as:

$$\mathrm{PSNR}=10\log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}}\right) \qquad (8.4)$$

$$=20\log 10\left(\frac{\mathrm{MAXi}}{\sqrt{\mathrm{MSE}}}\right) \qquad (8.5)$$

$$=20\log_{10}(\mathrm{MAX}_I)-10\log_{10}(\mathrm{MSE}) \qquad (8.6)$$

I represents the matrix data of the original image

K represents the matrix data of the degraded image

m represents the number of rows of pixels of images and i represents the index of that row

n represents the number of columns of pixels of images and j represents the index of that row

MAX_I is the maximum possible pixel value of the image

For test sequences in 4:2:0 color format, PSNR is computed as a weighted average of luminance ($PSNR_Y$) and chrominance ($PSNR_U$, $PSNR_V$) components [14] as given below

$$PSNR = \frac{6 * PSNR_Y + PSNR_U + PSNR_V}{8} \quad (8.7)$$

8.2.2 Structural similarity index:

The structural similarity (SSIM) [15] index is a method for measuring the similarity between two images. SSIM emphasizes that the human visual system is highly adapted to extract structural information from visual scenes. Therefore, structural similarity measurement should provide a good approximation to perceptual image quality.

SSIM considers image degradation as perceived change in structural information. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. SSIM is defined in equation 6.

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (8.8)$$

Where x and y correspond to two different signals that need to be compared for similarity, i.e. two different blocks in two separate images.

μ_x represents the average of x

μ_y represents the average of y

σ_x^2 represents the variance of x

σ_y^2 represents the variance of y

σ_{xy} represents the variance of x and y

$c1 = (k1L)^2$

$c2= (k2L)^2$

c1 and c2 they are two variables to stabilize the division with weak denominator

L represents the dynamic range of the pixel-values.k1=0.01 and k2=0.03 by default

In order to evaluate the image quality this formula is applied only on luma. The resultant SSIM index is a decimal value between -1 and 1, and value 1 is only reachable in the case of two identical sets of data.

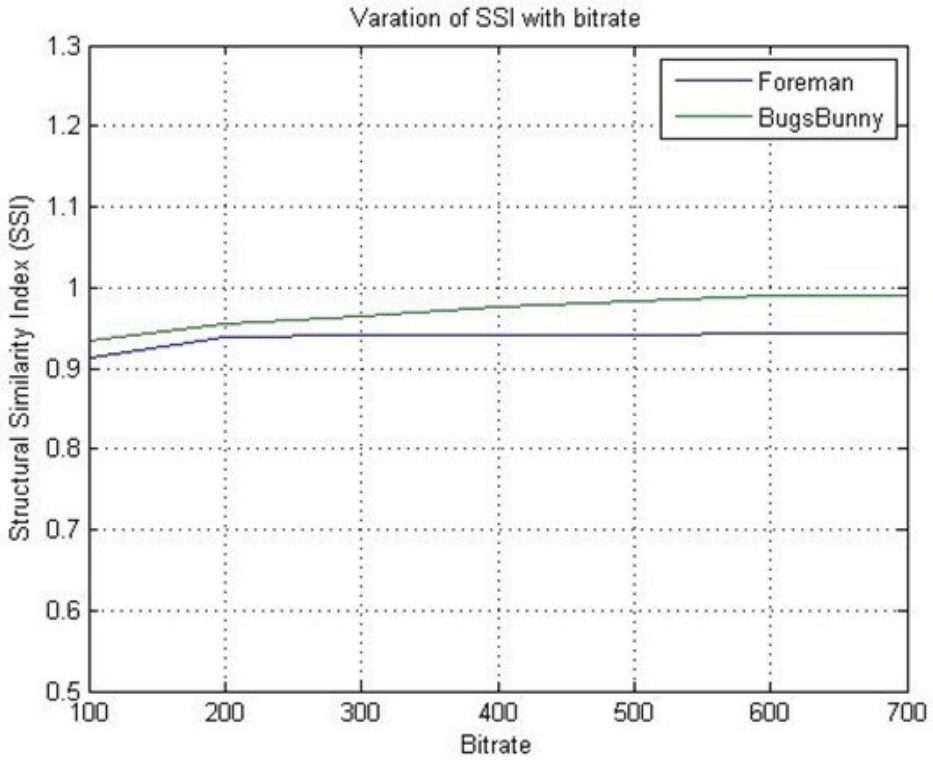


Figure 8.4 Variation of SSI with Bit rate

From the above Figure 8.4 the amount of similarity between the original and compressed images can be seen. The index value reduces when the bit rate decreases. It has a value very close to 1 when there is maximum similarity between the original and compressed images.

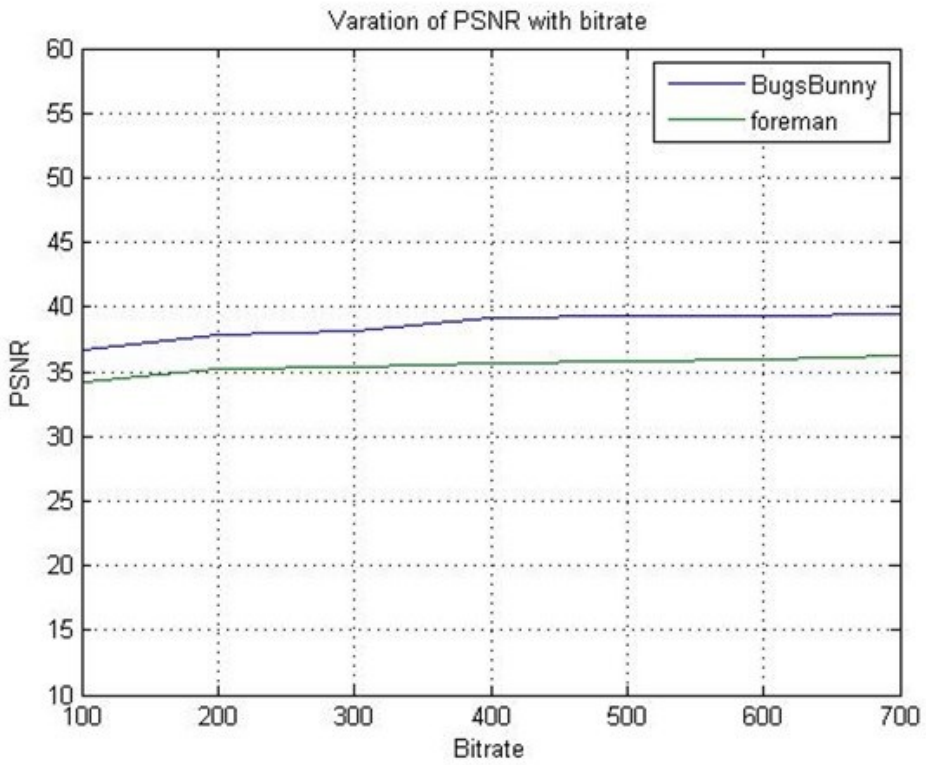


Figure 8.5 Variation of PSNR with Bit rate

From the above Figure 8.5 the amount of distortion between the original and compressed images can be seen. The PSNR value reduces when the bit rate decreases. Also the range in which PSNR is varying is not huge and it is not linear in nature.

8.3 Quality of Experience

This is a subjective experience measure based on an investigation of the impact of PSNR on the actual quality of the picture (or that this measure is explained later). For the purpose of analysis ,consider the equation stated below will be suitable to show the relative benefit of the various techniques.

For the purpose of analysis we consider the equation (4.3) stated previously will be suitable to show the relative benefit of the various techniques:

$\max imize_{i=1,2,..m}$ {Picture Quality +Non stall factor}

where: h(x)=Picture quality= $\max_{\{\alpha,\beta\}} \sum_{u\varepsilon U} \sum_{v\varepsilon V} \sum_{r(x)\varepsilon R} \sum_{s\varepsilon S} f_{vrs}\alpha_{uvrs}$

$r(x) = c_x s / x$, x represents the quantization parameter, r(x) is the bits to encode when the quantization step is QP ,S is the encoding complexity , is the co-efficient of the model.

g(z) =Non stall factor= $\sum_{i=1}^{n} \phi_i(z_i)\log(r_i)$

When x increases h(x) increases and g(z=mx)increases. Similarly decreases when x decreases. Also h(x) and g(z) can be opposite (one case maximal x and other case minimum x)

Consider the case when h(x) is maximized when x is maximized and g(x) is maximized when x is minimized ,the optimal value of x is chosen by applying optimization. Hence we use convex optimization. It helps in choosing the maximum (h(x) +g(z))

h(x) uses the formula mentioned because one sigma covers the bit rate r(x) which is needed to calculate the PSNR, one of them is for video rates. From the PSNR value the relative MOS value is obtained.

The constraints will prevent all the users from having the highest possible resolution since they all are sharing the resources available. Unless, of course, the available resources are sufficient for all subscribers to get video at the highest rates.

g(z) uses the formula mentioned. The g(z) value is an optimization function designed to evaluate the selected compression and see which is better so that least number of stalls are seen.

The PSNR to MOS mapping is done with the help of [12]:

Table 8.1 PSNR TO MOS mapping

PSNR[DB]	MOS
>50	5(Exc
40-50	4(Good)
25-40	3(Fair)
20-25	2(Poor)
<20	1(Bad)

The Table 8.1 shows the range of different PSNR corresponding to the mean opinion score. The mean opinion score represents the quality of the video.

Chapter 9

Experimental Setup, Results And Analysis

The JM 18.6 H.264 high profile reference software encoder is used for encoding different resolution test sequences. The main advantage of this encoder is that it can be executed using a parameter file (encoder.cfg and decoder.cfg)which outlines the list of values such as frame rate ,QP, GOP pattern, start frame to encode and the number of frames. The encoder output consists of the console print of various values such as the bitrate used, number of bits to encode and the average SNR values for the Y, U and V components over the group of frames of the video encoded. In addition, the encoder also produces a reference output of the video after encoding is completed. The takes the encoded bit stream along with the reference output produced by the encoder as the input and uses it to decode the stream and produce the corresponding output video file.

Test sequences in .yuv format are encoded JM 18.6/JM VC 10 [15]. Different resolutions of test sequences are used. The high profile config settings in JM VC10 is used during execution. The following sections give brief description of configuration settings used in the encoding tools.

General Configuration of JM VC 10

FramesToBeEncoded : 15 # Number of frames to be coded

Framerate : 15.0 # Frame Rate per second (0.1-100.0)

Profile DC : 100 # Profile IDC (66=baseline, 77=main, 88=extended; FREXT Profiles: 100=High, 110=High 10, 122=High 4:2:2, 244=High 4:4:4, 44=CAVLC 4:4:4 Intra, 118=Multiview High Profile, 128=Stereo High Profile)

IntraProfile : 1 # Activate Intra Profile for FRExt (0: false, 1: true)

Level DC : 40 # Level IDC (e.g. 20 = level 2.0)

IntraPeriod : 1 # Period of I-pictures (0=only first)

IDRPeriod : 1 # Period of IDR pictures (0=only first)

QPISlice : 22 # Quant. param for I Slices (0-51) (22, 27, 32 or 37 is used at a time)

Rate control:

RateControlEnable = 1 # 0 Disable, 1 Enable

Bitrate = 10000000 # Bitrate (bps)

InitialQP = 0 # Initial Quantization Parameter for the first I frame

InitialQP depends on two values: Bits per Picture and the GOP length

Basic Unit = 0 # Number of MBs in the basic unit and should be a fraction of the total number

# of MBs in a frame ("0" sets a BU equal to a frame)

Channel Type = 0 # type of channel (1=time varying channel; 0=Constant channel)

RCUpdateMode = 2 # Rate Control type. Modes supported:

0 = original JM rate control,

1 = rate control that is applied to all frames regardless of the slice type,

2 = original plus intelligent QP selection for I and B slices (including Hierarchical),

3 = original + hybrid quadratic rate control for I and B slice using bit rate statistics

Command line parameters for using JM VC10 encoder: lencod [-h] [-d defenc.cfg] {[-f curenc1.cfg]...[-f curencN.cfg]} {[-p EncParam1=EncValue1]...[-p EncParamM=EncValueM]}

Options:

-h	Prints parameter usage.
-d	Use <defenc.cfg> as default file for parameter initializations. If not used then file defaults to “encoder.cfg” in local directory.
-f	Read <curencM.cfg> for resetting selected encoder parameters. Multiple files could be used that set different parameters.
-p	Set parameter <EncParamM> to <EncValueM>. The entry for <EncParamM> is case insensitive.

Sample command line parameters for JM VC10 encoder:

C:\ H.264\JM\JM Software\JM 18.6\JM\bin>lencod.exe –f encoder_high.cfg -p InputFile="C:\Users\Karishma\Desktop\Foreman.yuv" –p SourceWidth=176 -p SourceHeight=144

Traditional compression applied in the network is a static value which is 20%.This value was shown by experience in the real network as a good balance of stalls and quality. The videos were stalling when the number of users increased, limiting capacity. Hence a compression scheme that changes the compression rate dynamically

in order to get the least number of stalls is needed in order to avoid the expense of changing rest of the infrastructure.

Rate control approach with variable QP and fixed QP have been applied. It Is found that by using variable QP we can achieve the target bit rate for varying network conditions

Adaptive compression uses the quantization parameter as the main parameter to be controlled.

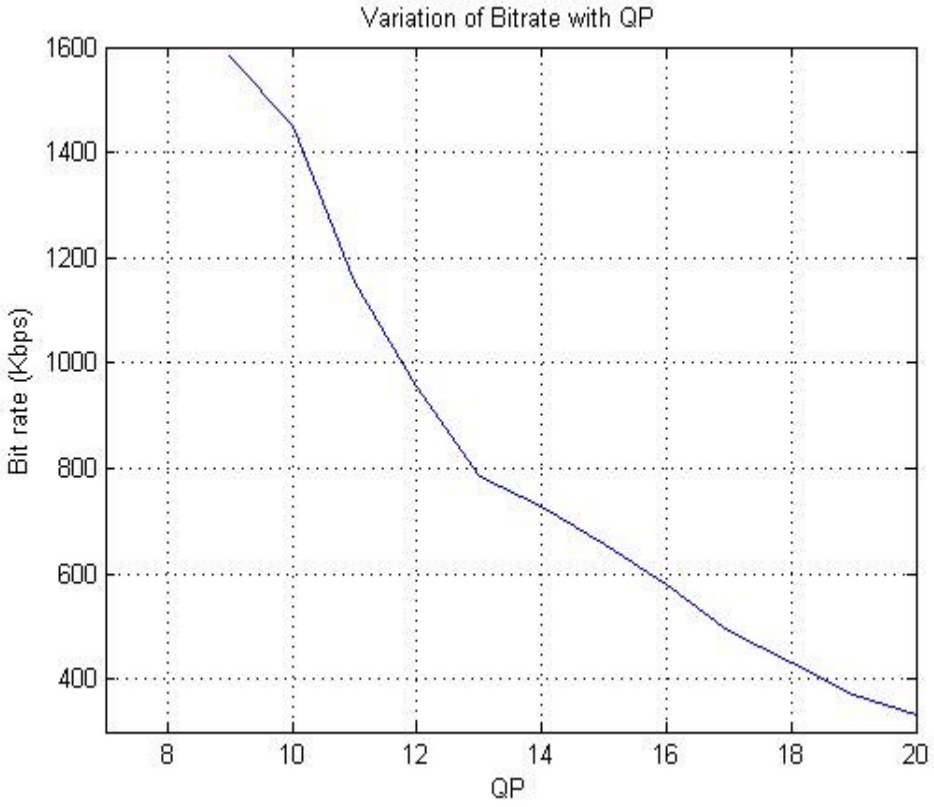


Figure 9.1 Variation of bitrate verses the quantization paramter

Observations:

The bandwidth of the network changes due to a variety of reasons like congestion in the network, packet losses, etc. In order to resolve the problems stated a rate control

algorithm is applied, that dynamically adjust the encoder parameters mainly QP.As quantization parameter increase the bit rate decreases.

Table 9.1 The bitrate values achieved for different QP values

QP	Bit Rate
9	1532.43
10	1470.2
11	1
12	956.34
13	843.12
14	796.34
15	654.67
16	579.14
17	493.32

Table 9.1 - Continued

18	14 32.12
19	13 70.76
20	13 30.43

Observations:

From the table 9.1 it can be seen that the bit rate value decreases with an increase the quantization parameter

I have simulated four different video sequences with rate control enabled and rate control disabled .The approach has been tested for the below set of raw video sequences:

Sequence 1: Foreman

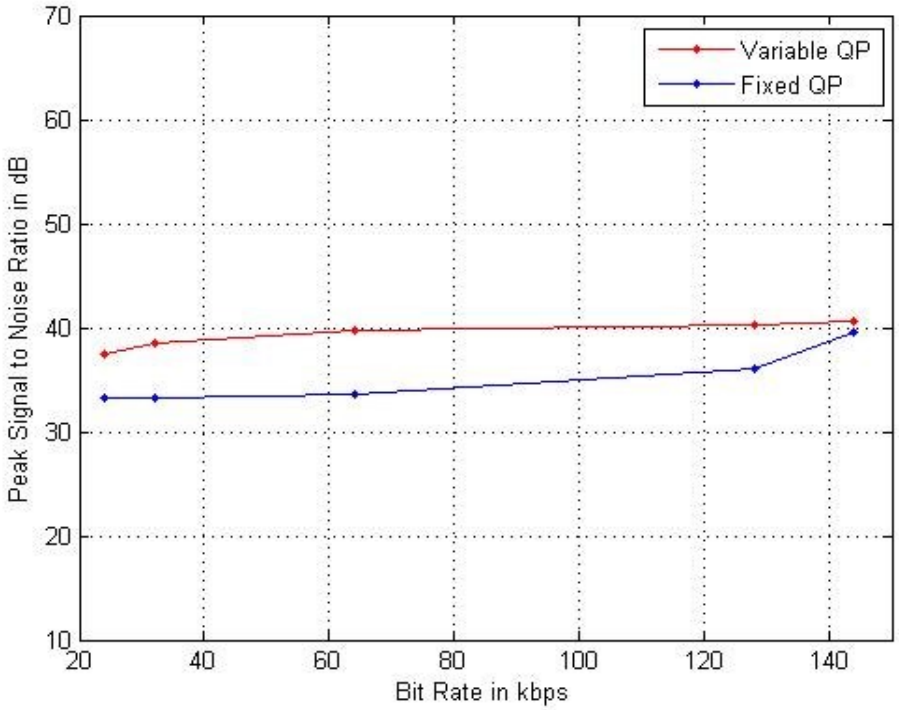


Figure 9.2 Foreman video sequence encoding with rate control enabled and disabled (low amount of movement)

Observations:

From the Figure 9.2 it can be seen that higher the bit rate of the video, higher is the PSNR value.Variable quantization parameter(rate control enabled) results are better compared to fixed quantization paramter(rate control disbaled)

Sequence 2: Bugs Bunny

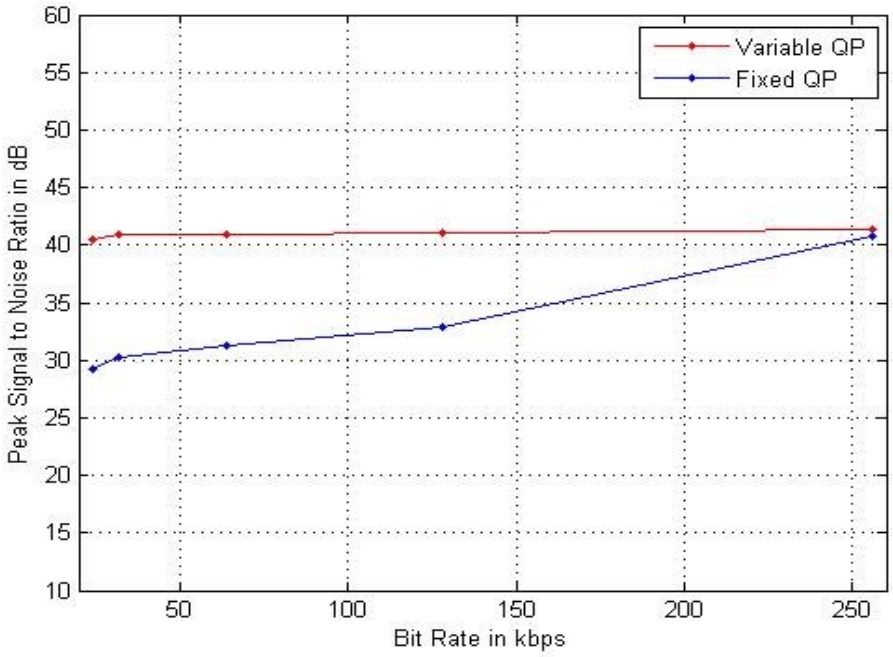


Figure 9.3 Bugs Bunny video sequence encoding with rate control enabled and disabled(animation)

Observations:

From the Figure 9.3 it can be seen that higher the bit rate of the video, higher is the PSNR value.Variable quantization parameter(rate control enabled) results are better compared to fixed quantization paramter(rate control disbaled)

Sequence 3: Akiyo

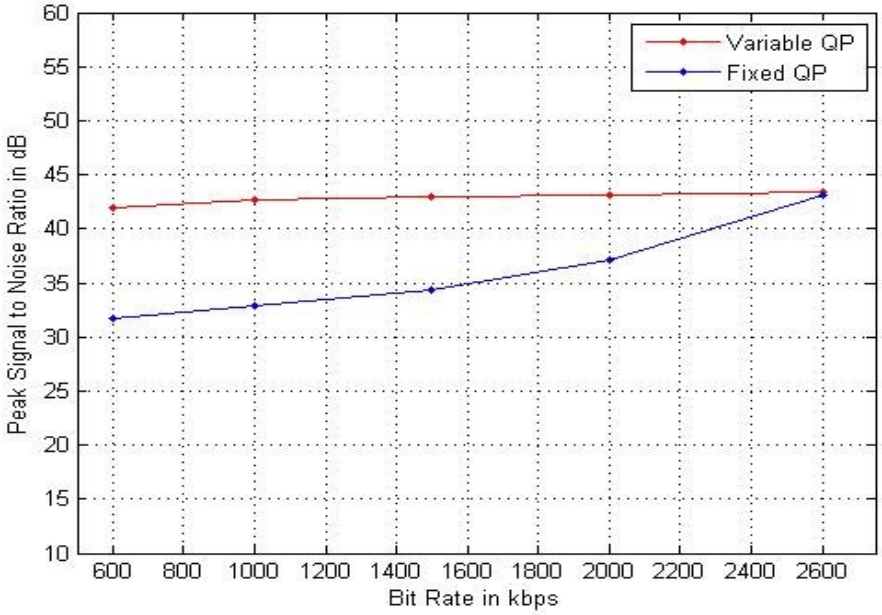


Figure 9.4 Akiyo video sequence encoding with rate control enabled and disabled (low amount of movement)

Observations:

From the Figure 9.4 it can be seen that higher the bit rate of the video, higher is the PSNR value.Variable quantization parameter(rate control enabled) results are better compared to fixed quantization paramter(rate control disbaled)

Sequence 4: Miss America

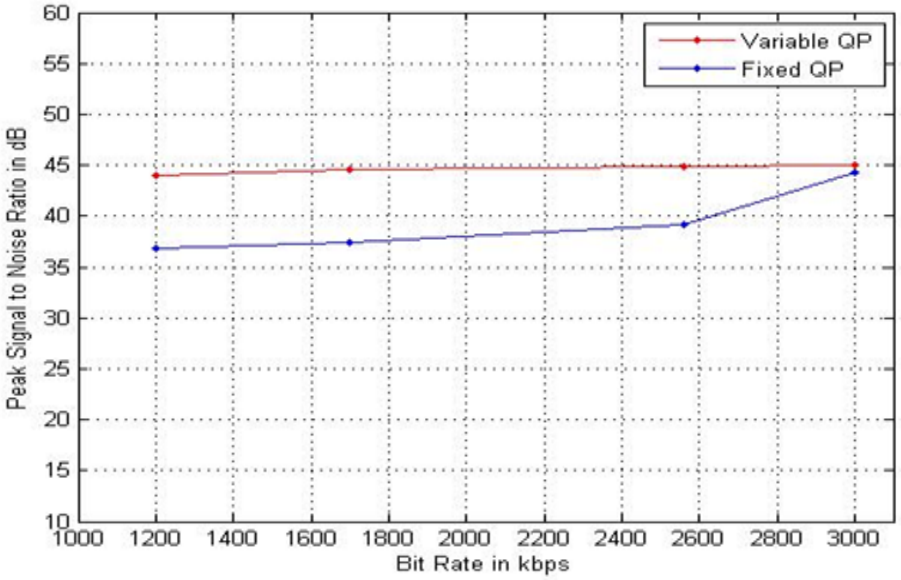


Figure 9.5 Miss America video sequence encoding with rate control enabled and disabled (animation)

Observation:

From the Figure 9.5 it can be seen that higher the bit rate of the video, higher is the PSNR value.Variable quantization parameter(rate control enabled) results are better compared to fixed quantization paramter(rate control disbaled).

Conclusion:

By changing the quantization parameter, reduces the bit rate which makes it adaptable to the changing network conditions. By doing this the number of stalls (pauses) gets reduced. Also by doing this a significant improvement in the PSNR ratio is seen as shown in the figures9.2,9.3,9.4 and 9.5.

Higher PSNR (Peak Signal to Noise Ratio) is achieved using varying the quantization parameter.

In order to reduce the stalls the compression rate changes which is a key parameter between the stalls and quality of the video

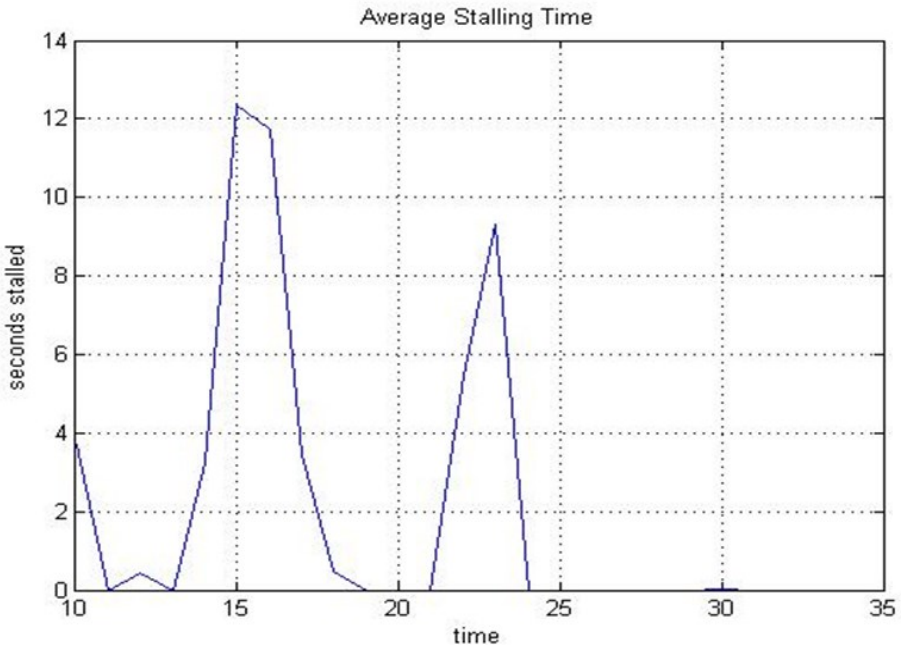


Figure 9.6 Average Stalling Time for a video session

Observations:

The Figure 9.6 shows the average amount of video stalling for all decoded videos, normalized to a 60 second scale.

If a video did not stall, it is counted as 0 seconds of stalling.

If a 60s video took 90s to download, it is counted as 30s of stalling.

Similarly, if a 6s video took 9s to download, the video is first normalized to the 60s video scale and then counted as 30s of stalling.

The average video stalling time is 4.17 seconds

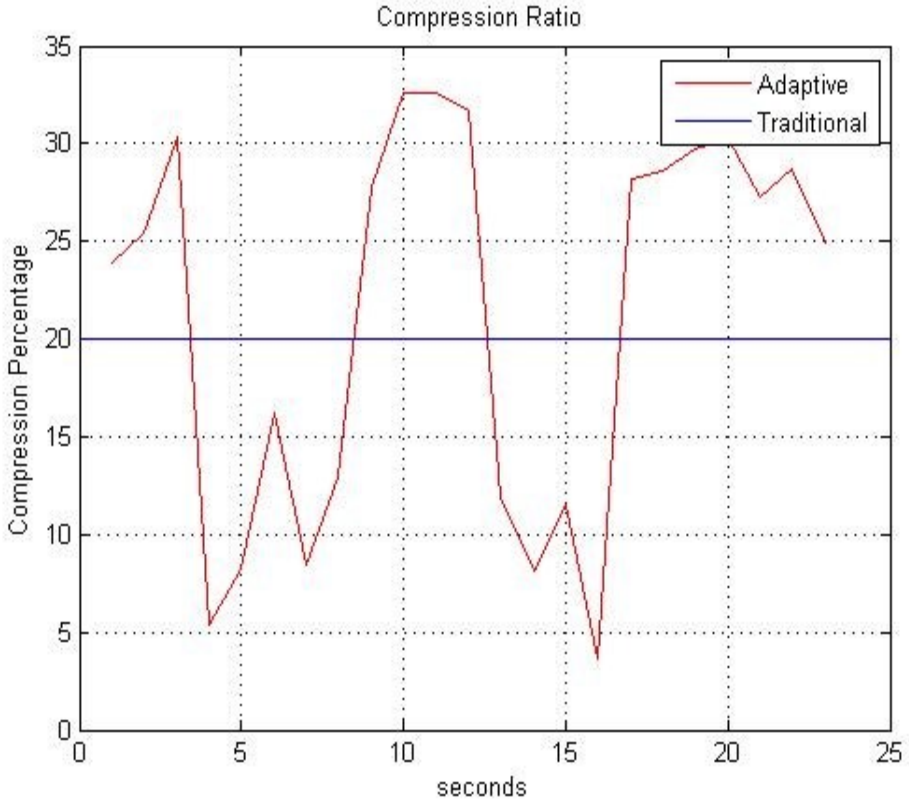


Figure 9.7 Comparision of compression ratio

Observations:

The Figure 9.7 shows the compression percentage trend for traditional and adaptive compression in a real world system.

Traditional compression always has a fixed value of 20%

Adaptive compression rate varies from 0% to 40%.

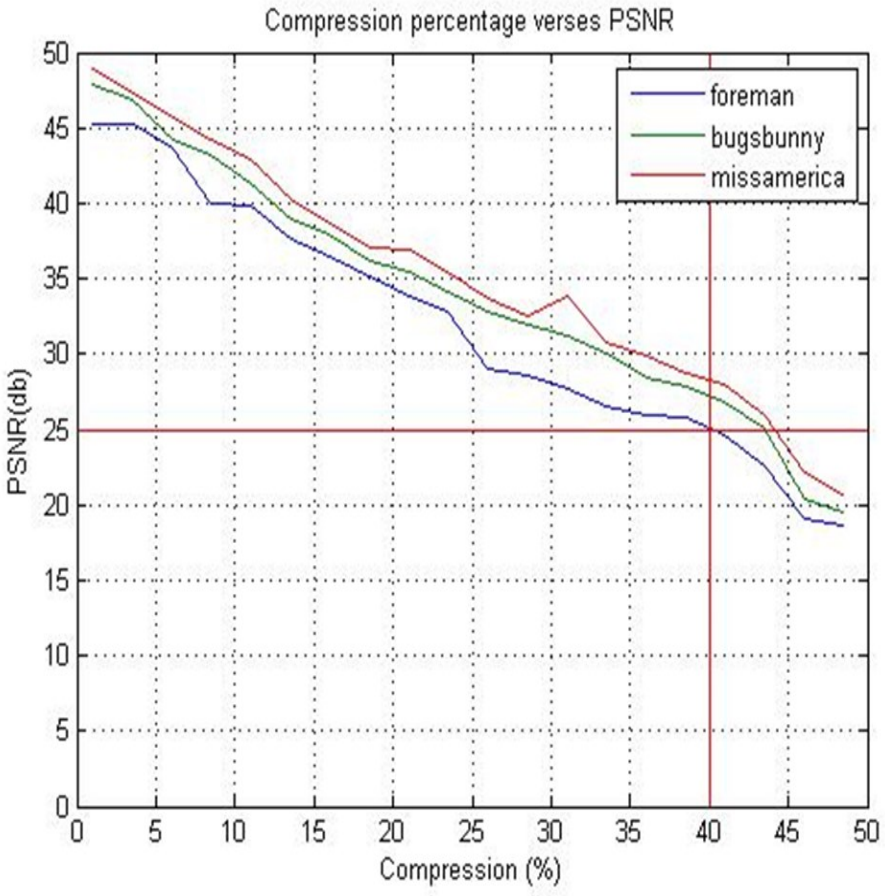


Figure 9.8 Compression percentage and psnr threshold

Observations:

As the compression percentage increases the PSNR value decreases as there will more distortion. The threshold for my algorithm is 25db for PSNR and 40% for the compression ratio as shown by the red line above because it gives an acceptable level of PSNR which is of good quality. Once the compression ratio increases more than 40% the quality is not acceptable.

As the percentage of compression increases the number of videos streamed increases with a lowered quality but without stalling as explained earlier.

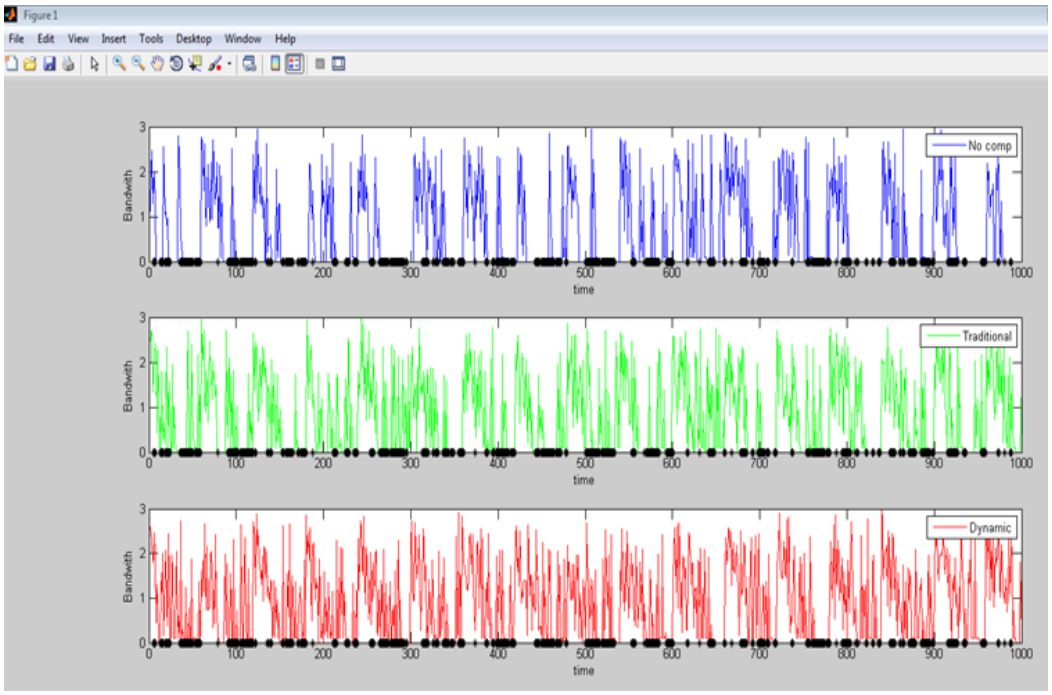


Figure 9.9 Variation of the bandwidth during 1000 second time interval

Observations:

From the Figure 9.9 it can be seen that the number of stalls gets reduced when the amount of compression is increased.

The black dots(when the graph goes to zero) represent a stall

From Figure (a) it can be seen that the graph goes to zero a lot due to the lack of bandwidth with congestion in the network.

When Traditional compression(Figure (b)) is applied ,the number of stalls are reduced compared to no compression but still there are many pauses

When adaptive compression(Figure(c)) is applied the number of stalls are reduced to a very great extent compared to traditional and no compression

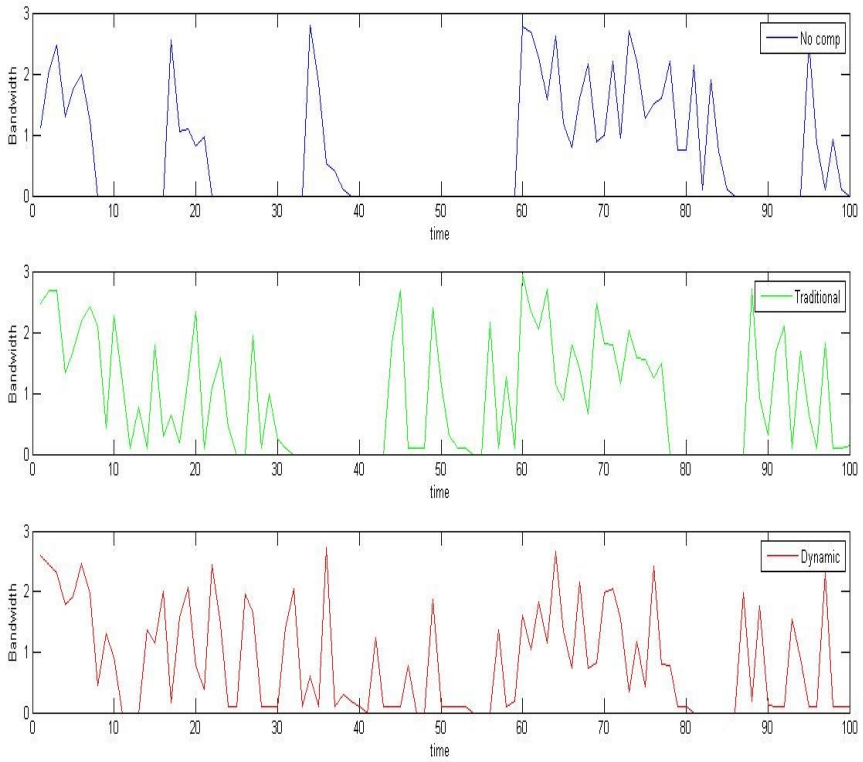


Figure 9.10 Variation of the bandwidth during 100 seconds time interval

Observations:

Figure 9.10 is a subset of figure 9.9 which considers 100 seconds worth of playback time and corresponding bandwidth.

We observe that bandwidth goes to zero more frequently in traditional and no compression compared to adaptive compression. It depicts the chance of occurrence of a stall which is considerably decreased in dynamic optimization.

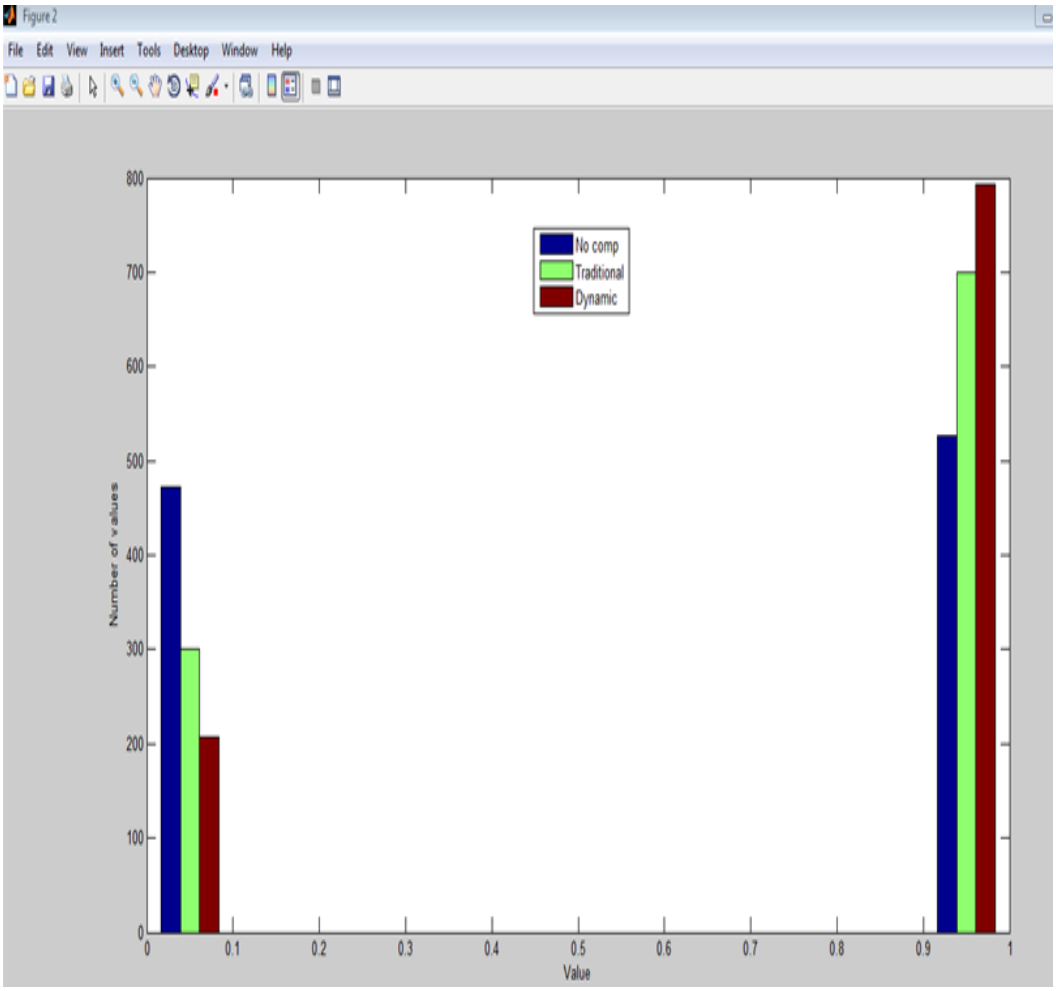


Figure 9.11 Number of stalls and no stalls for the three compression schemes

Observations:

In the Figure 9.11, the left side of the graph indicates the number of stalls. the number of zeros represent the number of stalling events

In the Figure 9.11, the right side of the graph indicates the number of no stall. The number of one's represents the number of non-stalling events.

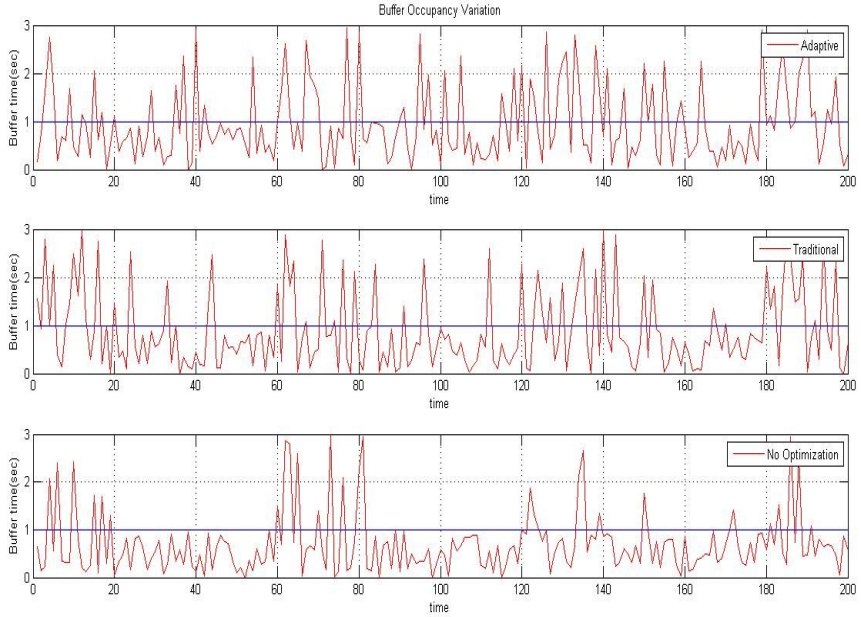


Figure 9.12 Buffer occupancy variation for 200 second time interval for the three compression schemes

Observations:

The Figure 9.12 shows the video playback data that is stored in a buffer. When the buffer value goes below 1 it indicates the buffer is not filled. When it is greater than 1 it indicates the buffer is filled.

The percentage of non-stalling time/video playback time during the entire video session is shown in Figure 9.12 for different schemes.

Adaptive optimization is 68% of the total video session

Traditional optimization is 41% of the total video session

No Optimization is 32% of the total video session

It can be concluded that adaptive optimization stalls for very less time compared to traditional and no-optimizatio

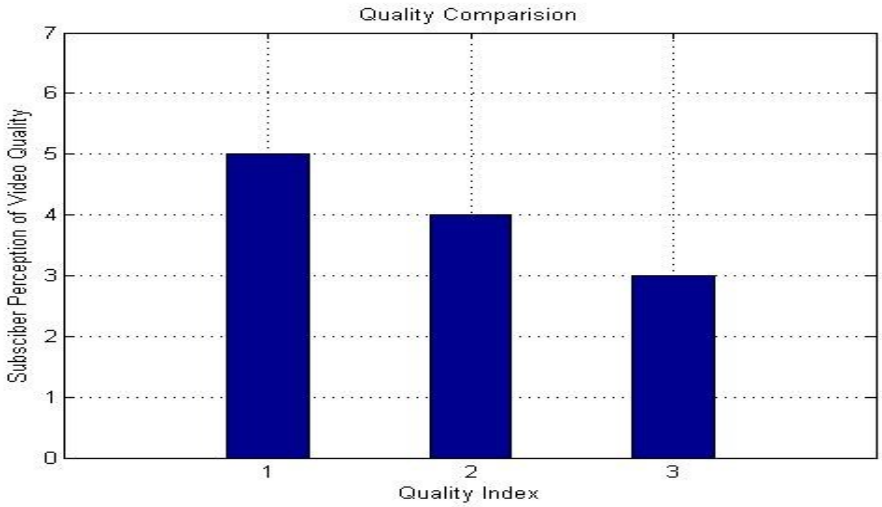


Figure 9.13 Quality index verses subscriber perception of video quality comparison for the three compression schemes (lower is better)

Observations:

In reference to figure 9.13:

The subscriber rates the quality of the video on a scale of 0-5.

The Quality Index is calculated by the summing up the benefit and the penalty (stalls).

Bar 1 represents no optimization which has the best quality and the MOS is 5.However it has the maximum number of stalls.

Bar 2 represents traditional optimization which has the decreased quality because of compression and the MOS is 4.However it has less number of stalls compared to no optimization.

Bar 3 represents adaptive optimization which has the least acceptable quality because of dynamic compression and the MOS is 3.However it has least number of stalls.

Subscriber experience for Dynamic > Traditional > No Optimization performance.

The performance of dynamic optimization is the best it leads to less number of stalls which in turn leads to the best quality of experience.

The maximum achievable subscriber experience is calculated in term of mean opinion Score which is as below:

No –optimziation-5

Traditional-4

Dynamic-3

Chapter 10

Conclusion

Due to network congestion and bandwidth constraints in the network, we experience pauses (stalls) while streaming the video. As the number of subscribers increase the network resources available decreases which cause more number of interruptions. The subscriber satisfaction is measured in terms of the quality and number of stalls/interruptions occurred in the video stream.

The major results of this thesis are shown below:

1. The threshold of the adaptive compression scheme is found to be 40% with a PSNR of 25 dB which equals to a MOS value of 3.

2. The percentage of non-stalling events for adaptive is 68%, traditional 41% and no optimization 32%

3. For the already existing work eliminating stalls is not the primary goal but in my work eliminating stalls with acceptable level of quality is the goal.

4. Adaptive streaming user buffer approach has been investigated in my paper. It was a future work that had to be considered for [10]

5. The subscriber quality index is purely subjective .Based on MOS, video quality and number of stalls dynamic optimization is considered to be the best approach for compression in wireless networks.

The best method to ensure subscriber experience in the modern wireless network is dynamic adaptive video optimization because it makes efficient use of the available bandwidth.

Adaptive Optimization is shown to provide bandwidth savings, reduce stalls, and not impact image quality significantly It improves the perceived network performance and

protects the subscriber experience within the finite limitations of the deployed infrastructure.

Chapter 11

Future Work

1. Improve the buffer capacity which requires more sophisticated hardware and software
2. Better Adaptive bandwidth estimation in a live network
3. Advanced compression schemes other than H.264

Appendix A

Acronyms

3GPP Third Generation Partnership Project

LTE Long Term Evolution

QP Quantization parameter

MOS Mean Opinion Score

PSNR Peak Signal To Noise Ratio

SSIM Structural Similarity Index

RD Rate Distortion

QOS Quality of Service

KKT Karush Kuhn Tucker

NAL Network Abstraction Layer

ME Motion Estimation

MC Motion Compensation

TCP Transport Congestion Protocol

TCPW TCP Westwood

References

[1] www.youtube.com

[2] www.vimeo.com

[3] Caching Eliminates the Wireless Bottleneck in Video Aware Wireless Networks

[4] “Cisco visual networking index: global mobile data traffic forecast update, 2012–2017,”http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI-Forecast_QA.html.

[5] JVT Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264-ISO/IEC 14496-10 AVC), March 2003, JVT-G050 available on http://ip.hhi.de/imagecom_G1/assets/pdfs/JVT-G050.pdf

[6] T. Wiegand et al, “Overview of the H.264/AVC Video Coding Standard”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 560-576, Jul. 2003.

[7] H.264 tutorial by I.E.G. Richardson: https://www.vcodex.com/h264.html

[8] MPEG 2 Test Model 5, Rev. 2, Section 10: Rate Control and Quantization Optimization, ISO/IEC/JTC1SC29WG11, April 1993

[9]Thang, Truong Cong, et al. "An evaluation of bitrate adaptation methods for HTTP live streaming." Selected Areas in Communications, IEEE Journal on32.4 (2014): 693-705.

[10] Casetti, Claudio, et al. "TCP Westwood: end-to-end congestion control for wired/wireless networks." Wireless Networks 8.5 (2002): 467-479.

[11] Cevher, Volkan, Steffen Becker, and Martin Schmidt. "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics."Signal Processing Magazine, IEEE 31.5 (2014): 32-43.

[12] Combettes, Patrick L., and Jean-Christophe Pesquet. "Proximal splitting methods in signal processing." Fixed-point algorithms for inverse problems in science and engineering. Springer New York, 2011. 185-212.

[13] Toni, Laura, et al. "Optimal set of video representations in adaptive streaming."Proceedings of the 5th ACM Multimedia Systems Conference. ACM, 2014.

[14] https://www.ffmpeg.org/download.html

[15] Access to JM 18.6 Reference Software: http://iphome.hhi.de/suehring/tml/

[16]] Swapnil Shingvi “Adaptive rate control algorithm to improve the performance of h.264 video codec in varying network conditions “

[17] Sheikh, Hamid Rahim, and Alan C. Bovik. "Image information and visual quality." Image Processing, IEEE Transactions on 15.2 (2006): 430-444.

[18] W.Stallings, High-Speed Networks: TCP/IP and ATM Design Principles. Upper Saddle River, NJ: Prentice-Hall, 1998

[19]Theodore Rappaport, "Wireless Communications: Principles and Practices," Upper Saddle River, NJ, Prentice Hall PTR, 1996, Print.

[20] Geometric Programming for Communication Systems- http://www.princeton.edu/~chiangm/gp.pdf

[21] Gary J. Sullivan, Video Compression—From Concepts to the H.264/AVC Standard

[22] Xiaoling Qiu, Haiping Liu, Dipak Ghosal, and Biswanath Mukherjee University of California, Davis Adaptive Video Compression Rate Optimization in Wireless Access Networks

[23] Xiaoling Qiu, Haiping Liu, Deshi Le, Song Zhang Dipak Ghosal, Biswanath Mukherjee: OPTIMIZING HTTP-BASED ADAPTIVE VIDEO STREAMING FOR WIRELESS ACCESS NETWORKS

[24] Luigi Alfred0 Grieco, Saverio Mascolo, Eugenio Di Sciascio: A Mathematical Model for the Steady State Throughput of the Westwood TCP Congestion Control Algorithm

[25] https://helpx.adobe.com/adobe-connect/kb/configure-ports-1935-443-80.html

[26] http://www.princeton.edu/~chiangm/gp.pdf

[27] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning.” in Proc. Advances in Neural Information Processing Systems (NIPS), vol. 20, pp. 161–168, 2008.

[28] R. T. Rockafellar, Convex Analysis. Princeton University Press, 1970

[29] Jain R. The art of computer systems performance analysis, John Wiley and Sons,1991.

[30] Mascolo S., Casetti C., Gerla M., Sanadidi M., Wang R. TCP Westwood: End--End Bandwidth Estimation for Efficient Transport over Wired and Wireless Networks. Proceedings of ACM Mobicom July 2001, Rome, Italy. To appear on ACM Wireless Networks (WINET), Special Issue on Wireless Networks with selected papers from MOBICOM 200 1.

[31] Hoe J.C. Improving the Start-up Behavior of a Congestion Control Scheme for TCP. Proceedings of ACM Sigcomm 1996.

[32] N. Ling, “High efficiency video coding and its 3D extension: A research perspective,” Keynote Speech, ICIEA, pp. 2150-2155, Singapore, July 2012

[33] http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC

[34] I.E.G. Richardson, “Video Codec Design: Developing Image and Video Compression Systems”, Wiley, 2002

[35] A. Puri et al, “Video coding using the H.264/MPEG-4 AVC compression standard”, Signal Processing: Image Communication, vol. 19, pp. 793-849, Oct. 2004.

[36] N. Ahmed, T. Natarajan and K.R. Rao, “Discrete Cosine Transform”, IEEE Transactions on Computers, Vol. C-23, pp. 90-93, Jan. 1974.

[37] Zongze Wu, Shengli Xie, Kexin Zhang and Rong Wu ,”Rate Control in Video Coding”

[38] http://www.pixeltools.com/rate_control_paper.html

[39] www.3gpp.org

Biographical Information

Karishma Katara was born in Hyderabad, India in 1988. She completed her Bachelor's in Electronics & Communications Engineering from Devarakonda Vittal Rao College of Engineering and Technology (DVRCET) affiliated to Jawaharlal Nehru Technological University, India in 2010

.She worked at HCL Technologies Ltd, India as a Software Engineer for two years eight months prior to joining the University of Texas at Arlington, Texas to pursue her Masters in Electrical Engineering specializing in Wireless Communication.

Karishma worked as Wireless Quality Analyst intern at Jasper Wireless for eight months, with QA testing group. Her research interests lies in various aspects of Wireless Communications, and Embedded systems.