SVC BIT STREAM EXTRACTION AND ITS APPLICATIONS

by

Xiaozheng Huang

B.E., Zhejiang University, 2007

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Applied Science
in the School
of
Engineering Science

SIMON FRASER UNIVERSITY

Summer 2011

APPROVAL

Name: Xiaozheng Huang

Degree: Master of Applied Science

Title of Thesis: SVC bit stream extraction and its applications

Examining Committee: Dr. Daniel Lee, Associate Professor, School of Engineering Science, Simon Fraser University
Chair

Dr. Jie Liang, Associate Professor, Engineering Science, Simon Fraser University
Senior Supervisor

Dr. Jiangchuan Liu, Associate Professor, Computing Science, Simon Fraser University
Supervisor

Dr. Mirza Faisal Beg, Associate Professor, Engineering Science, Simon Fraser University
Supervisor

Dr. Sami Muhaidat, Assistant Professor, Engineering Science, Simon Fraser University
Internal Examiner

Date Approved: 5 May 2011

Abstract

Scalable Video Coding (SVC) is the scalable extension of H.264/AVC which extends spatial, temporal and quality scalability on the H.264 video stream. In this thesis, a fast priority index (PID) assignment algorithm is first developed for the MGS (Medium Grain Scalability) packets in SVC. We formulate the index assignment problem as the quantization of the rate-distortion (R-D) slopes of MGS packets, and use the Lloyd-Max algorithm to find the optimal solution. The slope quantization index of a packet is used as its PID. The complexity of our method is much lower than existing method. The quantization-based PIDs facilitate the comparisons of packets from different video streams. Video multiplexing results show that the overall PSNR can be improved up to 1 dB.

A fair Peer-to-Peer (P2P) scalable video streaming scheme is also proposed in this thesis. To improve the quality fairness of multiple video streams, we modify our fast PID assignment method where the base-layer quality is also embedded in the Supplemental Enhancement Information (SEI) message of the SVC bit stream, in addition to the quantized rate-distortion slope of each MGS packet. To build a fair "contribute-and-reward" mechanism for P2P video streaming, we propose a multi-hierarchical topology that is based on peers' uploading bandwidth. Finally, we combine these two parts to build a SVC-based P2P network, which fully utilizes the quality scalability of SVC, and the end-user quality is determined by its uploading bandwidth contribution.

We also propose a caching mechanism for SVC video streaming in the P2P network based on the instant supply and demand estimation. We generalize the previous work from the literature into the SVC case and get the estimation formulation of the supply and demand in SVC video streaming on P2P network. We found that SVC video streaming can further reduce the normalized traffic load from the server compared with AVC video streaming in the same caching strategy set-up.

Key Words: Scalable Video Coding, Bitstream Extraction, Peer-to-Peer Video Streaming, Cache Optimization

Acknowledgments

I would give my sincere thanks to my senior supervisor, Dr. Jie Liang, for his patience, helpfulness and generosity. He not only gave me a lot of guidance and assistance on my research and thesis, he also set a good example for his students as how he treated his work with passion, diligence and conviction. I would also thank Hongfei Du for his help on our collaborative work. I am also grateful to Dr. Jiangchuan Liu, Dr. Daniel Lee, Dr. Sami Muhaidat and Dr.Mirza Faisal Beg for being my thesis committee members.

I would also thank lab mates including Jing Wang, Xiaoyu Xiu, Upul Samarawickrama, Mohammad Jafar Taghiyar, Dong Zhang, Calvin Che, Mehdi Seyfi, Parmida Beigi, Duncan Chan and Yu Gao for their valuable suggestions on my research.

Finally, I would like to thank my mother who gave me a lot of support during my master study abroad.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

Video coding, or video compression, is a modern technology for multimedia consumer electronic products and services. If uncompressed, the raw data of a two-hour movie would require the storage allocation in the range of 112-675 GB. A low quality video telephony with 352x288 spatial resolution requires the bit rate as high as 5-10 Mb/s if the video is uncompressed, which exceeds most of the typical wired or wireless channel bandwidth today. Digital storage and transmission technologies, especially in the early days of consumer digital video, could not meet this demand in an affordable manner. Even today's storage and transmission technologies improves dramatically, they still can not meet consumers' increasing demand for higher quality video. A standard definition (SD) or high definition (HD) uncompressed video require bit rates in the range of 125-750 Mb/s. However, typical broadcast channels only support a fraction of this rate per video stream: 1-3 Mb/s for SD videos and 10-15 Mb/s for HD videos [22].

Since the launch of Digital Video Interactive (DVI) in 1987, which is widely recognized as the first video coding standard for the consumer market, a few branch of video compression standards were developed and widely used [41]. Two main streams of video coding standards are MPEG standards developed by ISO/IEC and H.26x recommendations developed by ITU-T. MPEG-4 Advanced Video Coding (AVC), also known as H.264, is the latest international video compression standard developed by the Joint Video Team (JVT) of ISO/IEC and ITU-T. Compared to MPEG-2, it can encode video with similar quality at half the bit rate.

With the fast development of network infrastructure, storage capacity and computational power as well as the new technology and standardization in video codec and transmission protocols, the number of video applications is booming recent years. The applications vary from video conferencing or video telephony over wired and wireless channels, video sharing web sites like YouTube over the Internet, standard and high definition IPTV broadcast to Blue-ray Disc and HD DVD [27]. For different applications, the requirements for the transmission and storage system may be quiet different.

Modern video transmission using the wired or wireless Internet are usually based on TCP/IP or RTP/UDP/IP for conversational and streaming services. Most of these networks are characterized by various transmission qualities and receiving devices. The varying transmission quality is mainly because of the adaptive resource sharing mechanisms of wired networks and fluctuant physical condition of wireless networks. The variety of devices with different capabilities ranging from smart phones with palm-sized screens, limited processing power and unreliable wireless channel bandwidth to high-end HDTVs with high-definition displays and reliable wired channel bandwidth [27].

On the source coding part, one way to deal with the above-mentioned characteristics of the modern video transmission networks is video transcoding [5]. Transcoding usually falls into two categories: The first one is to code the video from one video standard format like H.264 into another format like MPEG-2. The second one is to code the video from a specific spatio-temporal resolution into a lower spatio-temporal resolution. During video transcoding, the original encoded video bitstream is fully or partially decoded and encoded into anther video bit stream with different format or resolution or both.

The scalable extension of H.264/AVC, known as Scalable Video Coding (SVC), is another appropriate solution to the challenges posed by the characteristics of modern video

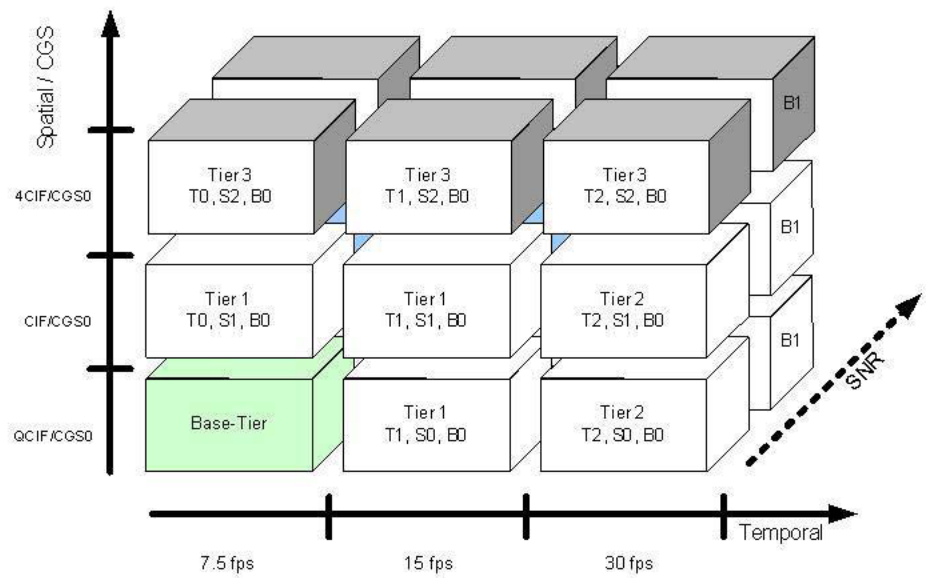


Figure 1.1: The layering structure of SVC bit stream [44]

transmission networks [27, 47]. The standardization objective of the SVC is to enable the encoding of a layering video bit stream that contains one or more subsets of bit streams that can themselves be decoded if parts of them are removed, while the computational complexity and bit rate overhead is just slightly increased compared to the counterpart achieved using H.264/AVC design with the same quality of video for end-users.

There are three types of scalability in the current Scalable Extension of H.264/AVC, which are temporal, spatial, and quality scalability [27]. Spatial scalability and temporal scalability define cases in which parts of the bit stream contain the same video content with a smaller picture size (spatial resolution) or frame rate (temporal resolution), respectively. For quality scalability, the subset of the bit stream offers the same spatiotemporal resolution as the original bit stream, but with a lower quality where quality is often defined as signal-to-noise ratio (SNR). Different types of scalability can also be combined, so that multiple representations with different spatial or temporal or both resolutions and bit rates can be supported within a single scalable bit stream [27, 32].

Quality scalability is classified into three granular levels. Fine Grain Scalability (FGS), owing to its complexity and rate-distortion inefficiency, it was not included in the final standardization of SVC. Coarse Grain Scalability coding (CGS) can be considered as a special case of spatial scalability with identical spatial resolution for base and enhancement layer.

Another granular level of quality scalability is the Medium Grain Scalability (MGS), which is included inside a CGS layer to increase the flexibility of the bit stream adaptation. The MGS is achieved by splitting the transformed coefficients within each block of the enhancement layer into several fragments. Each is assigned to one quality layer. This approach allows progressively improved quality and graceful degradation when fragments are discarded during bit rate adaptation [33].

The SVC bit stream is encapsulated into packets with an integer number of bytes, called Network Abstraction Layer Units (NALU). Each NALU belongs to a specific spatial, temporal, and quality layer [27]. To facilitate bit stream extraction, the H.264 SVC standard defines a 6-bit priority Index (PID) at the NALU header to signal the importance of the NALU [33]. This enables a lightweight bit stream parser in the transmission path to dynamically drop some NALUs to meet the network or end-user constraints, before the decoder is actually invoked to decode the extracted partial bit stream. Therefore the PID assignment algorithm plays an important role in determining the final performance of the SVC.

For the previous PID assignment algorithm, all NALUs are first sorted according to their operational R-D slopes. The sorted NALUs are then uniformly partitioned into some groups, and the group indexes are used as the PIDs of the NALUs. A hierarchical merging algorithm [26] is developed in and implemented in the current Joint Scalable Video Model (JSVM) software [39]. To obtain an extracted R-D curve that is as convex as possible, it first treats each NALU as a Quality Layer and sorts all NALUs according to their R-D slopes. After that, in each iteration, the two adjacent NALUs with the smallest merging cost are merged, where the merging cost is the increase of the area under the R-D curve. The merging operation is repeated until the target number of the quality layers is reached. However, the merging operations have relatively high complexity. Moreover, the PID in the current JSVM scheme is only valid within one video stream. Comparing PIDs from different bit streams, as needed in some video multiplexing applications, are usually meaningless, since the contribution of an NALU from one sequence with a lower PID might be much larger than an NALU from another sequence with a higher PID. Also this method can only show the relative importance within the current video stream, thus the absolute video quality when the packets with certain PID are included in the bit stream can not be indicated by the previous PID assignment method.

This thesis is mainly focused on the quality scalability of the Scalable Extension of H.264/AVC, especially the so-called Main Grain Scalability (MGS) in SVC. MGS technology tremendously increases the SVC bit stream extraction flexibility while the bit rate overhead and computational complexity is slightly increased. Using proper ordering of the packets within the bitstream, the extraction-flexible bit stream can show convex rate-distortion curve as the subsets of the bitstream increase to the complete bitstream. Thus it is an attractive solution for video streaming in heterogenous networks while the transmission condition is not fixed.

1.2 Main Contribution

In this thesis, by formulating the PID assignment as a Lloyd-Max quantization (LMQ) problem, we propose a fast algorithm to assign PID to MGS NALUs. The method also facilitates the comparison of NALUs from different video streams, with the help of SEI (Supplementary Enhancement Information) packets. This makes it useful for applications such as network-based video multiplexing. The algorithm can also be implemented in realtime and assign the PIDs adaptively based on the recent statistics of the video. Experimental results show that our method achieve the same R-D performance as the current JSVM in the extraction of one video bit stream. When multiplexing different videos, our method can achieve up to 1 dB of improvement. In addition, the real-time implementation of the algorithm does not sacrifice the performance of the off-line method.

We refine the above PID assignment scheme by embedding the absolute base-layer quality as well as the Lloyd-Max quantization-based PID into the SEI massage of SVC bit streams to achieve fair video quality among multiple video streams for scenarios like video multiplexing when bandwidth of the shared channel are restricted. The variance of the video quality among multiple video streams is largely decreased by applying this method. We also adopt a uploading bandwidth categorization of peers to build a fair contribute and reward mechanism for a P2P video streaming network. When the proposed PID assignment method is applied to this P2P network, end-users can obtain fair video quality based on its uploading bandwidth contribution while fully utilizing the quality scalability of SVC bit streams.

We also propose a caching mechanism for SVC video streaming in the Peer-to-Peer network based on the instant supply and demand estimation. We generalize the previous work from the literature into the SVC case and get the estimation formulation of the supply and demand in SVC video streaming on Peer-to-Peer (P2P) network. We found that SVC video streaming can further reduce the normalized traffic load from the server compared with AVC video streaming in the same caching strategy set-up.

The results presented in this thesis have been published in [60, 61]. A journal paper [24] is also published whose results are not included in this thesis.

1.3 Thesis outline

Chapter 2 covers the fundamentals and implementation details of Main Grain Scalability of SVC and the currently used Priority Index assignment method for MGS packets. Chapter 3 introduced the proposed Priority Index assignment method based on the Lloyd-Max Quantization and its application for video multiplexing. Chapter 4 investigates its application in fair P2P video streaming. Chapter 5 presents a caching mechanism in the P2P scalable video streaming network. Finally, Chapter 6 presents the conclusions and discuss some future works.

Chapter 2

Background

2.1 SVC video streaming application

Due to the adaptivity of Scalable Video Coding (SVC) bit stream, the main application for the scalable extension of H.264/AVC is video streaming, especially for multi-users with different requirements in heterogenous networks.

Some papers compared the performance difference between H.264/AVC and its scalable extension, SVC, for a specific purpose. In [69], the author compares H.264/AVC simulcast and SVC in terms of their required capacity in an IPTV network scenario where a number of TV channels is offered to the subscribers. In [44], the author investigates adaptive video streaming in P2P network based on SVC video streaming and evaluates the results compared to conventional AVC video streaming approaches. The bufferless statistical multiplexing of both AVC and SVC are investigated in [52]. It is shown that prioritizing the frames according to the number of dependent frames can increase the number of supported streams. The author of [23] examines the implications of video traffic smoothing on the numbers of statistically multiplexed SVC, AVC, and MPEG-4 Part 2 streams, the bandwidth requirements for streaming, and the introduced delay. It identifies the levels of smoothing that ensure that more H.264 SVC streams than H.264/AVC streams can be supported. Or their inter-standard adaptation between AVC and SVC like in [43] which designs a reconfigurable hardware implementation of AVC/SVC transcoding for real time video adaptation in order to address different terminals and bandwidths or in [45] which introduced an improved SVC-to-AVC rewriter.

More papers exploits the characteristics of SVC video streaming in specific networks and transmission environment. In [38], the author provides an analysis and evaluation of H.264/SVC in error prone environments, quantifying the degradation caused by packet losses in the decoded video. The investigation of transmitting SVC video over multi-input multi-output (MIMO) wireless systems is presented in [16]. The problem of SVC video streaming over orthogonal frequency division multiplexing (OFDM) channels is considered both in [62] and [9]. SVC video streaming on cellular networks is investigated both in [46] and [65]. A generic optimization framework to evaluate the SVC performance on mobile broadcasting systems, such as ISDB-Tmm is proposed in [14]. The author in [15] provides an analysis of SVC broadcast streaming services using a realistic LTE PHY layer while the authors in [21] and [55] both presents their results of SVC video streaming on DVB wireless transmission. A real-time video adaptation on WiFi router is also exploited in [34]. The author in [11] employs a combination of three transport techniques for SVC video streaming. SVC video streaming in non-infrastructure-based network types like mobile ad-hoc networks (MANETs) is investigated in [56]. The author in [2] also evaluates and demonstrates a technique for streaming H.264 SVC video over a DDS middle-ware. The architectural principles and design details necessary so that SVC can also enable secure adaptive streaming at a sender and secure R-D optimized adaptation at an untrustworthy mid-network node is presented in [35].

There is also a great number of paper focused on SVC video streaming on different protocols like [8] on TCP and DCCP (using CCID3), [40] on IEEE 802.16e network system, [36] on T-NASS protocol respectively. A few papers make some endeavors on SVC with networking coding as in [4] and [28].

There is another category which combined SVC with Unequal Error Protection. The author in [3] makes use of Reed Solomon codes for unequal error protection. An unequal error protection (UEP) scheme for scalable video coding using low density parity check (LDPC) codes is proposed in [64], which can allocate LDPC codes with unequal coding rates to different sub-streams that decreases the end-to-end distortion of the system as compared to equal error protection (EEP) scheme, resulting in greater system throughputs. An algorithm is proposed in [19] to estimate the overall distortion of decoder reconstructed frames due to enhancement layer truncation, drift and error concealment in SVC video and combines unequal error protection with the bit stream extraction. The author of [25] also proposed an unequal error protection (UEP) scheme for scalable video bitstream over packet-lossy networks using forward error correction (FEC), which exploiting jointly the unequal importance existing both in temporal layers and quality layers of hierarchial scalable video bitstream. It is addressed in [67] the problem of unequal error protection for the scalable video streaming over packet-erasure channel considering both the channel condition and the video characteristics.

For this thesis, we mainly focused on SVC bit stream extraction.

2.2 Quality scalability

2.2.1 Inter-layer prediction and Coarse Grain Scalability

For spatial scalability, inter-layer prediction is defined as the case when encoding the enhancement layer video, the encoded base layer bitstream can be used as a candidate mode for prediction. Several techniques are employed for inter-layer prediction including inter-layer intra-prediction, inter-layer residual-prediction and inter-layer motion prediction [32].

As we mentioned before, Coarse Grain Scalability (CGS) can be considered as a special case of spatial scalability with identical picture sizes for base and enhancement layer while it is also supported by the general case of quality scalability. The same inter-layer prediction methods as for spatial scalable coding are employed, but excluding some features like corresponding upsampling and the inter-layer de-blocking for intra-coded reference layer macroblocks. A major difference is that the inter-layer intra- and residual-prediction are directly performed in the transform domain. When employing inter-layer prediction for CGS in SVC, a refinement of texture information is typically achieved by requantizing the residual texture signal in the enhancement layer with a smaller quantization step size relative to that used for the lower CGS layer. However, the number of available rate points is the same as the number of layers. Switching between different CGS layers can only be done at selected points in the bit stream, which restricted the further application of CGS.

2.2.2 Medium Grain Scalability

Medium Grain Scalability (MGS) is provided within a CGS layer (or a dependency id, which is a syntax element) by coding successive refinements of the transform coefficients of the base quality of the layer. It splits the refinement coefficients of each block of the enhancement layer into several increments. This splitting of coefficients among increments allows a progressive refinement with MGS coding and graceful quality degradation when fragments are dropped during adaptation. The quality refinements are identified by the syntax element quality id, up to three quality increments are possible inside a dependency id layer [33].

As with H.264/AVC, the basic unit for the output of the SVC encoder and the input of the SVC decoder is a Network Abstraction Layer (NAL) Unit [51, 68]. Each NAL Unit (NALU) belongs to a specific spatial, temporal and quality layer and is signaled by certain syntax elements accordingly (using the high level syntax elements dependency id, temporal id, and quality id of the NAL header). For SVC, an access unit (AU or a more straightforward term as a picture) then consists of a set of NALUs from all spatial layers having the same temporal instant. For MGS layers, each MGS increment within one access unit is encapsulated as one NALU. An extractor can be used to adapt the stream to the target rate and/or resolutions, before the decoder is actually invoked to decode the extracted partial bit stream [27].

2.3 Bit stream extraction

2.3.1 Quality Layer

The basic idea of the concept ”Quality Layer” is to assign a prioritization order to each NAL unit within the whole scalable bit stream rather than the basic extractor [27]. However, the NAL units constituting the base quality of a specific spatial/temporal resolution must be included in the bitstream since NAL units with higher quality levels are dependent on them. Therefore, these NALUs are excluded from the optimization process of prioritization ordering. This prioritization poses another layering structure on the already encoded bit stream for bit rate adaptation. Whether an NAL unit is included in the SVC bit stream or nor is based on the bit rate constraint and the dependency among NAL Units [33].

While JPEG2000's priority within the bit stream [17], which is called the Embedded Block Coding with Optimized Truncation (EBCOT), is reflected in the hierarchical bit stream structure, the priority index of a NAL Unit in SVC is a post-compression processing of the encoded SVC bit stream. Thus the priority assignment is independent of the encoding parameters as long as it allows MGS or FGS quality scalability in the SVC bit stream. Furthermore, this prioritization mechanism can be used at any points in the video transmission scheme, for part of the video or the whole video stream, if only the bit stream structure conforms to syntax and semantics of the SVC Recommendation [27, 33].

Quality Layers should be derived and assigned before transmission. A Quality Layer is computed for each NAL unit and a priority_index as an indicator is inserted in the bit stream for further usage (e.g., transport or storage adaptation). After assigned, it is the priority_index which is utilized for bit rate adaptation. Bit rate adaptation using Quality Layers prioritization can be made during any points after encoding but before decoding. It can be made before transmission at the encoder, in a network node adaptation or at the receiver before decoding [27, 33].

The following are the general sub-steps to calculate the Quality Layer information of each fragment, specifically here is the NAL Units [27, 33]:

- Analyze and establish dependency constraints between NAL units
- Derive the size and distortion reduction associated of each NAL unit
- Sort NAL units based on the rate-distortion contribution
- Assign the priority index for each NAL Unit

For each NALU, its impact delta rate on the global rate is calculated, which is its own packet size.Its impact delta distortion on the global reconstruction quality is also considered. This impact on the distortion can be measured as the difference between the reconstructed image sequence with and without the NALU. Since one picture can be a reference picture for other pictures, the appearance of the quality increments of one picture can impact the quality of other pictures thus affecting the global quality. Dependency constraints should be taken into considerations when we calculate the distortion [27, 33].

2.3.2 Current Priority Index assignment for SVC in JSVM

Once quality increments have been sorted within the dependency constraints, Quality Layers are assigned. There are only 6 bits in the syntax elements to represent the Quality Layer information. The currently used method in JSVM [39] is presented below [26]. However, this method is firstly proposed for FGS, which is not adopted as an normative feature in the latest SVC standard. Here, we only consider bit stream extraction for MGS.

- Determining Rate-Distortion points R_i, D_i for all NAL units i that contain non-progressive refinement fragments;
- Sorting the NAL units by decreasing slopes $S_i = D_{imax} / R_{imax}$; Each NAL unit represents a separate initial Quality Layer;
- While(true)
- Determining minimum merging cost $\delta k = A_{k,k+1}$ - $(A_k + A_{k+1})$, which is demonstrated in Fig 2.1 and 2.2;

- If($N <= N_{target}$ && $k >= 0$) break;
- Merging Quality Layers k and k+1;
- Assigning Quality Layer Index Q to NAL units, so that $Q_j > Q_k$ when $j < k$

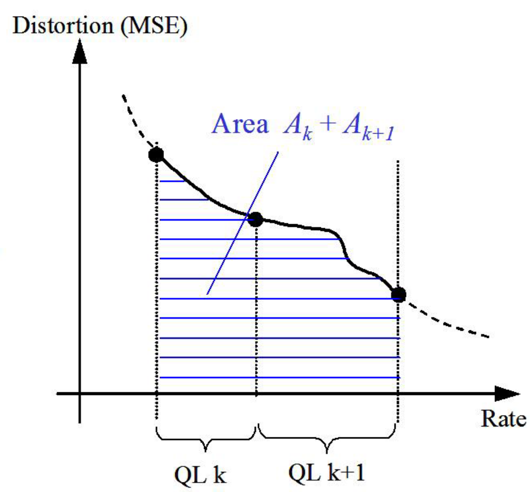


Figure 2.1: R-D curve before merging [26]

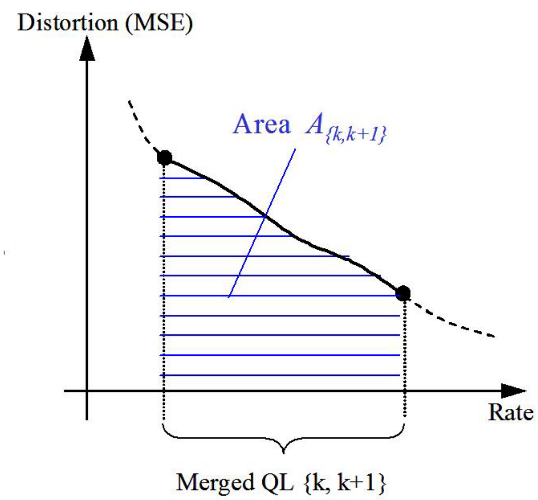


Figure 2.2: R-D curve after merging [26]

2.4 Other SVC Bit stream extraction methods and applications

Apart from the bit stream extraction using the above bit rate extraction scheme in JSVM, some bit rate extraction methods are also proposed in the literature. These endeavors try to resolve the problems which the current PID in JSVM has its limitations on.

2.4.1 Extraction based on trade-off of three scalabilities

The author of [13] proposes a method for evaluating the trade-off among three scalability dimensions: spatial, temporal, and quality scalability offered by the Scalable Extension of H.264/AVC. Firstly the evaluation for the trade-off of the difference in the metric of each scalability dimension is proposed. Three scalability dimensions are transformed into utility functions and an algorithm to find an efficient extraction path by using the points of inflection is proposed. Experimental results show that their approach can find a better extraction path as compared to the JSVM basic extractor.

2.4.2 Bit rate extraction for very low bandwidth

The author of [10] exploits the performance of SVC bit rate extraction for the case when the bandwidth is very low and proposed a global optimal GOP-adaptive Rate-Distortion optimization extraction algorithm for SVC streaming multicasting over P2P networks. It investigates the trade-off of temporal scalability and quality scalability for extremely low bandwidth and reassign the PIDs to the NAL units considering the bandwidth histogram. It shows better quality performance while parts of the network consists very low bandwidth transmission path.

2.4.3 Distortion estimation model based SVC bit rate adaptation

Due to the hierarchical prediction structure of the SVC and the concept of key pictures, SVC bit rate adaptation of SVC bit streams to intermediate bit rates is still a challenging task. One limitation is that the current method does not consider all NAL units from different layers for the optimization. For the current method, the distortion calculation is still based on a fixed order of including NAL units from NAL units from different levels progressively. While the actual extraction is based on the distortion calculation in this way, extraction pattern is most likely not in that way which the calculation is derived. Thus there is a mismatch of the distortion calculated before extraction and after extraction. It is proposed in [18] that a mathematical model based on Taylor expansion to estimate distortion. Based on discarding an arbitrary number of NAL units from multiple layers of a bitstream, the coefficients of Taylor series for the model are derived. Then, PIDs are assigned by utilizing this distortion estimation technique. However, the computational overhead should be large for this method since the parameters for estimation is unique for every video sequence.

A linear model of general predictive video coding is proposed in [54]. The authors then designed an algorithm to quantify the drift properties subject to prediction dependency and motion information, which can be utilized for Quality Layer assignment within the framework of SVC.

A simple distortion model to estimate the reconstruction distortion with drift error is also developed in [50]. They also designed a simple and fast weight-based priority setting algorithm to smooth the video quality in MGS bitstream extractions.

To our best effort, the above are the endeavors which focused on bit stream extraction that we can find in the literature.

2.5 Video streaming in P2P network

Since several years ago, there has been increasing interests from the academia as well as the industry to use Peer-to-Peer (P2P) network for large-scale video streaming. Viewers contribute their resources to a P2P overlay network to act as relays for the video streams [20]. An overlay network is a layer of virtual network topology in the application layer on top of the physical network, which directly interfaces to users [30]. It is really a challenge that typically a popular baseball match will attract over millions of viewers to watch the game simultaneously. The reason why P2P is an attractive solution is that

1. This technology is based on the current existing network infrastructure. Thus it saves the cost and time to deploy dedicated infrastructures;

2. Rather than the client-server model, in the P2P network, every peer is both a client to download the video content from other peers and a sever to stream video to others. Therefore, it is allowed to scale with the number of participants, as greater demand also results in more supplies. Although it still requires at least one single media source, which may be assumed not to fail, and is present throughout a broadcast session, the requirements of bandwidth of server and network is drastically loosened [37].

However, there are still some challenges for this technology to be used in video streaming:

1. It requires real-time delivery of the video content, though minor delays at the start-up or during later sessions are still tolerable unless the video is interrupted. Thus all connected users can get a TV-like viewing experience.

2. The peers may disconnect to the network voluntarily or passively at any time meanwhile packets often need to be relayed between several peers, thus increasing the unreliability of the network

3. The access bandwidth of the peers is often not enough to support high quality video, which requires at least several hundred Kbps. Therefore, there should be a way to enable video streaming with adaptivity and flexibility that accommodates bandwidth heterogeneity and fluctuations and to deliver videos with gracefully degradable quality. Scalable Video Coding (SVC), with its inherent scalability features, offers a promising solution to these problems [20, 37].

The P2P video streaming applications can be classified into two categories: the live (PPLive [49] and UUSee [58]) and Video-on-Demand (VoD) streaming (GridCast [7] and Vanderbilt VoD [6]), respectively.

In P2P live streaming, the video source is generated at the server in real-time and delivered to peers which are currently in the system simultaneously. In other words, all the peers are watching the same session of the video. The common viewing interests of peers will increase the video content overlap and encourage resource sharing among them. However, one more challenge of minimizing the streaming delay is posed against this application, specially when millions of users request to join the system at a same very short time period. Compared with live video streaming, VoD generally supports looser video content overlap among users and richer user interactions. Users can join in the system at any time and watch any part of the video) [63]. P2P VoD faces other challenges like content scheduling and caching problem.

2.6 Summary

In this chapter, we do a in-depth literature research on the mechanism of current SVC bit stream extraction, especially the Priority Index assignment which is highly related to the work in this thesis. We also investigate other bit stream extraction method as well as the main-stream applications of Scalable Extension of H.264/AVC.

Chapter 3

The proposed LMQ-based PID assignment

3.1 Introduction

In this chapter, by formulating the PID assignment as a Lloyd-Max quantization (LMQ) problem, we propose a fast algorithm to assign PID to MGS NALUs. The method also facilitates the comparison of NALUs from different video streams, with the help of SEI (Supplementary Enhancement Information) packets (with negligible overhead). This makes it useful for applications such as network-based video multiplexing. The algorithm can also be implemented in real-time and assign the PIDs adaptively based on the recent statistics of the video. Experimental results show that our method achieve the same R-D performance as the current JSVM in the extraction of one video bit stream. When multiplexing different videos, our method can achieve up to 1 dB of improvement. In addition, the real-time implementation of the algorithm does not sacrifice the performance of the offline method.

Table 3.1: The executing times of the merging method in [26] and the proposed method (sec.)

Sequence	[26]	Proposed	Time saved
coastguard	2.964	0.078	97.37%
mother-daughter	2.933	0.17	94.20%
mobile	2.761	0.22	92.03%
news	2.621	0.36	86.26%
Average	2.820	0.21	92.4%

3.2 Lloyd-Max quantization-based priority index assignment

Since the rate-distortion contribution of each MGS NAL unit is determined by its R-D slope, our objective is to embed the slope information in the PID syntax element. This can be achieved by using the quantized slope as the Priority Index. It is also desired that the PID assignment method can preserve as much slope information as possible, *e.g.*, it can minimize the reconstruction errors of the slopes. This will facilitate the PID-based comparisons of the significances of packets from different video streams in video multiplexing applications.

These requirements suggest that we can formulate the PID assignment as a scalar quantization problem, by treating the R-D slopes of all NALUs as the variables to be quantized. In particular, the quantizer generates fixed rate outputs, since each PID has 6 bits. It is well known that in this case, the optimal solution to minimize the mean squared error (MSE) of the reconstructed variables is given by the Lloyd-Max quantization (LMQ) algorithm [17], which assigns smaller quantization steps to the regions with higher probabilities.

The optimal solution of an M-level LMQ satisfies the following conditions [17]:

$$t_q = \frac{\hat{x}_{q-1} + \hat{x}_q}{2} \qquad q = 1, 2, 3 \ldots M-1 \tag{3.1}$$

$$\hat{x}_q = \frac{\int_{t_q}^{t_{q+1}} x f_X(x) dx}{\int_{t_q}^{t_{q+1}} f_X(x) dx} \qquad q = 1, 2, 3 \ldots M-1 \tag{3.2}$$

where $\hat{x}_q$'s are the reconstruction points, and t_q's are the decision boundaries. The optimal solution can be achieved by an iterative algorithm using the equations above [17].

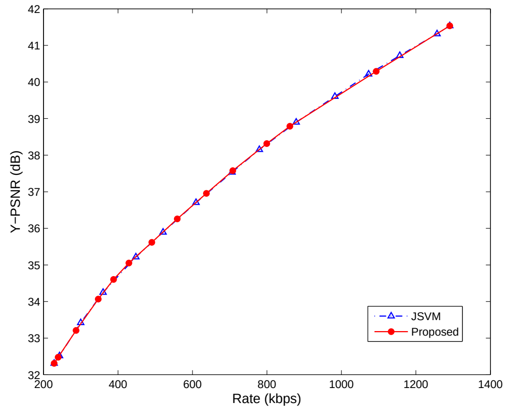
3.2.1 The LMQ-based PID Derivation Algorithm

Our LMQ-based PID assignment algorithm works as follows. As in [26, 31], we first sort all MGS NALUs according to their R-D slopes. We then apply the LMQ algorithm to quantize the slopes into 63 bins, with index from 0 to 62 (The PID of 63 is reserved for base layer packets). The PID of each NALU is then set to be the quantization index of the R-D slope of the unit. Our method terminates when the MSE change of the reconstructed slopes is less than 1%, which can usually be achieved within 10 iterations.

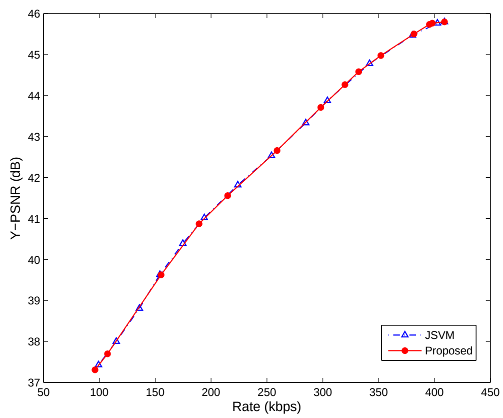
Since higher-layer packets depend on lower-layer packets, it is desired to drop the packets from higher layers. Therefore higher-layer packets need to have smaller PIDs. However, this is not always true if the PIDs are obtained by quantizing the R-D slopes of NAL units, because a higher-layer NALU might have larger slopes than a lower-layer NALU. In our algorithm, these conflicts are resolved after the termination of the LMQ by reducing the higher-layer NALUs' PIDs such that they are less than the PIDs of the NALUs in the lower layers.

To facilitate the comparison of the NALUs from different video streams at the network nodes or the decoders, we need to include the reconstruction values of the R-D slopes of NALUs in the bit stream. In this paper, we use the Supplementary Enhancement Information (SEI) packets to convey this information. The SEI packet contains a look-up table for the reconstructed R-D slopes with PID ranging from 0 to 62. Each reconstructed slope is represented by 4 bytes, so the total size is 252 bytes. Since only very few SEI packets are needed, the overhead is negligible. We also assume that the SEI packets are not lost during transmission. This can be achieved by applying additional protections to the SEI packets.

By applying the LMQ to the R-D slopes directly, our PIDs preserve the R-D information more faithfully than the uniform partition in [31], whereas the relationship between the PIDs and the R-D slopes is not clear in the merging method in [26], as will be shown in Fig. 3.2.



(a)

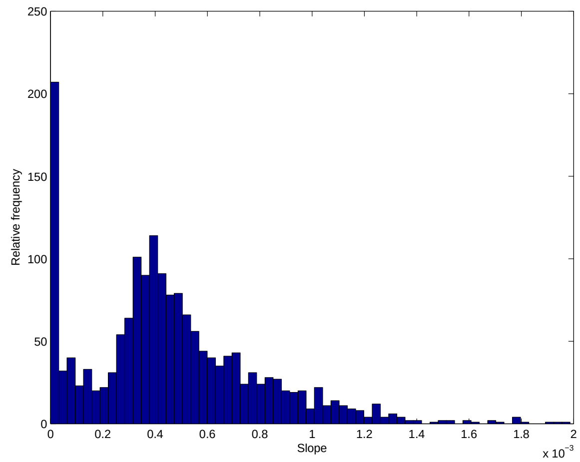


(b)

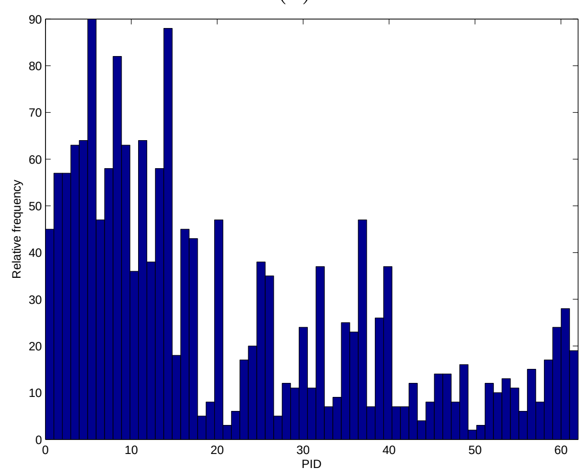
Figure 3.1: R-D curves of JSVM and our method for *Mobile* (a) and *News* (b).

3.2.2 Complexity and Performance

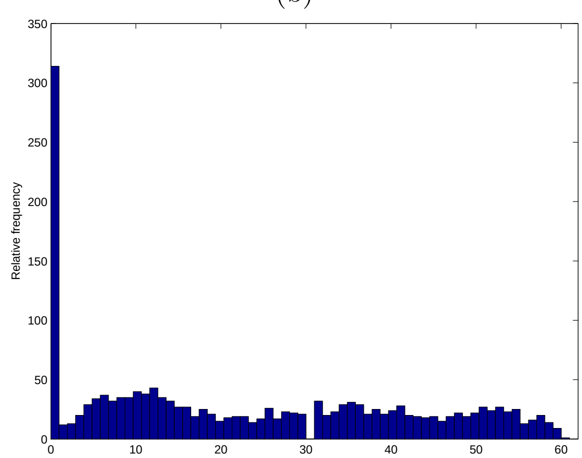
The hierarchical merging algorithm in [26] and the current JSVM needs to merge all possible pairs of neighboring layers to find the minimal cost. This is repeated in each iteration, until 63 layers are left. The number of comparisons is thus $O(N^2)$, where N is the number of MGS NALUs. Compared to this, the complexity of the proposed LMQ-based method is much lower. Table 3.1 lists the executing time of the merging algorithm in JSVM and the proposed LMQ method on a PC with Intel Core 2 Duo 2.4 GHz processor and 4GB memory. The time does not include the slope calculation and the sorting. The results show that the



(a)



(b)



(c)

Figure 3.2: (a) Histogram of NALU R-D slopes of *News*. (b) Histogram of the PIDs by JSVM. (c) Histogram of the PIDs by our method.

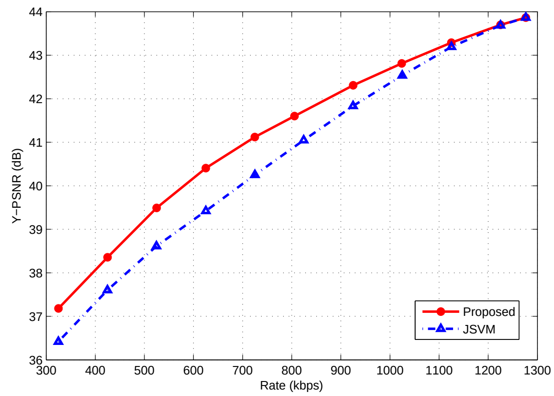
complexity of our method is only 7.6% of the method in [26]. In addition, our method is currently run in Matlab, and can be further sped up if implemented in C/C++. Therefore our method is more suitable for real-time applications.

To evaluate the R-D performance of the extracted bit stream, we encode four 289-frame QCIF video sequences with 1 base layer and 6 MGS enhancement layers. Fig. 3.1 shows the RD curves given by our method and the current JSVM. The curves in each graph are almost perfectly overlapped, which means that the two schemes have the same RD convexity for one video sequence.

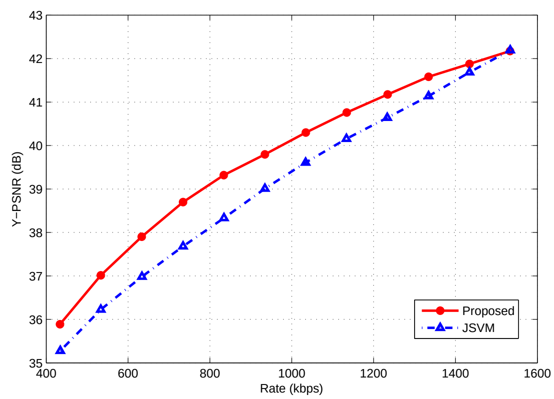
However, the assigned PIDs by the two methods are usually quite different, because the PID distribution in our method is more closely related to the distribution of the NALU R-D slopes. Fig. 3.2 shows the histogram of the NALU R-D slopes of the *News* sequence, as well as the PID histograms of the two methods. It can be seen that there are many NALUs with very small slopes. However, this is not reflected in the PIDs of the merging method. Moreover, its PID distribution shows more randomness than the original slope distribution, meaning that some PIDs are not used as often as they should be. The same observation can also be made in other sequences. Compared to this, the PID distribution in our method is much smoother. This can provide finer granularity when comparing NALUs from different video streams, as studied in the next section.

3.3 Application in Statistical Video Multiplexing

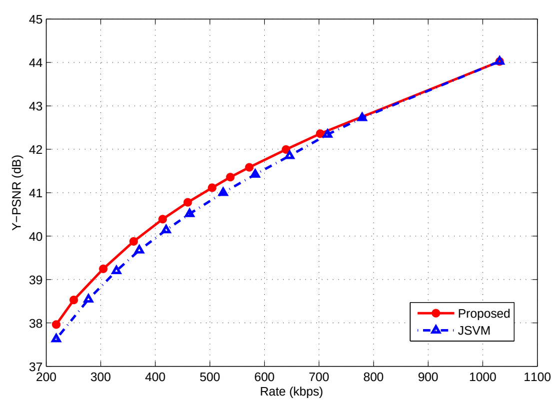
One promising application of the SVC is statistical video multiplexing. Most existing works on statistical video multiplexing focus on the joint encoding of multiple videos, where the encoder has the freedom of adjusting the coding parameters of all video sequences to meet the bandwidth constraint. However, in some applications, statistical video multiplexing is also necessary at a network node, where different pre-encoded incoming video sequences need to share a common output network link. In this case, it is impractical to decode and re-encode the videos. Instead, the network node can use the PID information to decide which NALUs from which videos should be dropped. The PIDs generated by the current



(a)



(b)



(c)

Figure 3.3: The average RD curves of the current JSVM and the proposed method in multiplexing of two videos. (a) Multiplexing *Coastguard* and *News*. (b) Multiplexing *Mobile* and *Silent*. (c) Multiplexing *Foreman* and *Mother-daughter*.

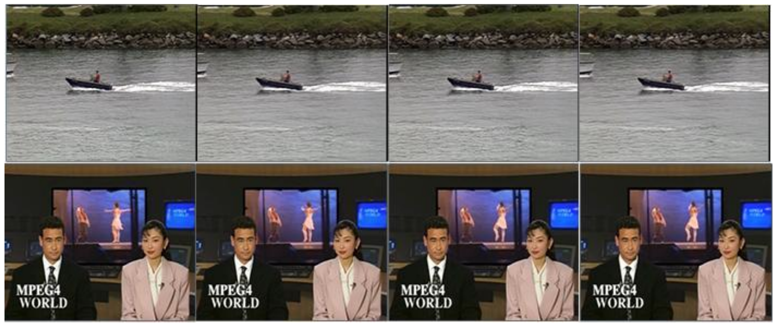
JSVM only reflect the relative contribution of the NALUs within each video sequence. In our method, since the PIDs and the SEI packets preserve the R-D slopes of NALUs, comparisons among NALUs from different videos is simplified greatly.

Fig. 3.3 shows the simulated average R-D curves of multiplexing two video sequences using the PIDs from the current JSVM and from our method, respectively. In each case, two video sequences are encoded with the same QP and MGS configurations, and the PIDs are assigned to each video independently, using the two methods. The rate extraction is assumed to be at a network node with enough buffer to hold one GOP of NALUs from the two sequences. When the current PID assignment method in the JSVM is used, the NALUs from the two videos with the smallest PIDs are dropped until the bit rate is within the target. When our PID assignment method is used, the extractor is able to sort the MGS NALUs from the two videos based on their reconstructed R-D slopes, with the help of the SEI packets. It then drops the packets with the smallest R-D slopes. PSNR (Peak Signal-to-Noise Ratio) is an engineering term for the ratio between the maximum possible power of a video signal and the power of noise that affects the quality of its representation, which is defined in 3.3 and 3.4 [29]. It is clear from Fig. 3.3 that our method yields up to 1 dB improvement. Fig. 3.4 also shows the subjective quality of the first 4 frames of two video multiplexing of two methods when the bit rate constraint is 750 kbps. The average PSNR of these 4 frames for the reference method and proposed method are 38.64 dB, 39.27 dB of *Coastguard* and 43.67 dB, 45.68 dB of *News*, respectively.

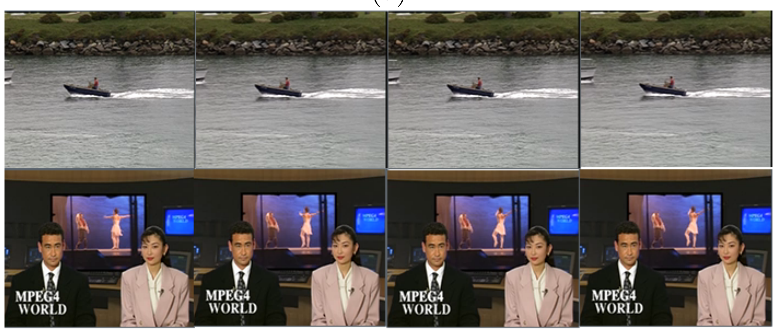
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]; \tag{3.3}$$

$$PSNR = 10\lg(\frac{255^2}{MSE}); \tag{3.4}$$

We now consider the multiplexing of more than two videos. Three methods are evaluated. The first method is static multiplexing, where each SVC bit stream is allocated its own channel. Since we show in Sec 3.2 that the R-D performance of our method in this case is



(a)



(b)

Figure 3.4: The first 4 frames of *Coastguard* and *News*. (a) reference method. (b) proposed method.

the same as the merging-based method in [26], only the results with our method are reported for the static method.

The other two methods use statistical multiplexing, where the base layer of each sequence is transmitted in an independent channel, and the enhancement layers of all sequences are transmitted in a shared channel. The merging-based PID assignment in the JSVM and the proposed LMQ-based PID assignment are tested in this case.

In the first example, seven video sequences are involved, each has a full bit rate larger than 384 kbps. For static multiplexing, each video is allocated 384 kbps. For statistical multiplexing, the bandwidth of each BL channel is 128 kbps, and the shared channel for ELs is $256 \times 7 = 1792$ kbps.

In the second example, five video sequences are used, each has a bit rate larger than 512 kbps. For static multiplexing, each video is allocated 512 kbps. For statistical multiplexing, the bandwidth of each BL channel is 128 kbps, and the shared channel for ELs is $384 \times 5 =$

Table 3.2: Avg. PSNR of multiplexing more than two videos

Example	Static	JSVM	Proposed
1	43.52	43.52	43.83
2	43.96	43.80	44.17

1920 kbps.

Table 3.2 shows that statistical video multiplexing with the proposed PID assignment achieves an average of 0.26 and 0.34 dB improvement over the static multiplexing and statistical multiplexing with the merging-based PID assignment.

3.4 Real-time Adaptive Priority Index Assignment

In all the examples in the previous sections, the entire video sequence is encoded before assigning PIDs to all of its NALUs. This offline PID assignment is not suitable for real-time applications. In fact, to the best of our knowledge, real-time adaptive PID assignment has not been reported in the literature. The low complexity of the proposed LMQ-based PID assignment makes it an ideal candidate for real-time applications.

Since the distortion associated with each NALU is calculated by considering the dependency among different frames in a GOP, the basic unit of the real-time PID assignment is chosen as one GOP.

The real-time PID assignment algorithm needs to solve two main issues: what kind of statistics should be used to generate the PIDs, and when to update the SEI messages. For the first issue, one way is to use the cumulative statistics from the beginning of the video sequence. Another option is to use the statistics of several recent GOPs within a window. The potential problem with the cumulative approach is that it could react too slowly when there is a scene change. The problem with the window approach is that it might not have enough statistics to take full advantage of the LMQ method.

Since the statistics used to calculate the PIDs are time-varying in the real-time algorithm, the SEI messages should be updated. This can be done periodically or adaptively. For

Table 3.3: Different Real-time PID Assignment Algorithms

Algorithm	Statistics	Updating freq.
1	Cumulative	Per GOP
2	Cumulative	Per 5 GOPs
3	Cumulative	Per 10 GOPs
4	Last 5 GOPs	Per GOP
5	Adaptive window	Adaptive

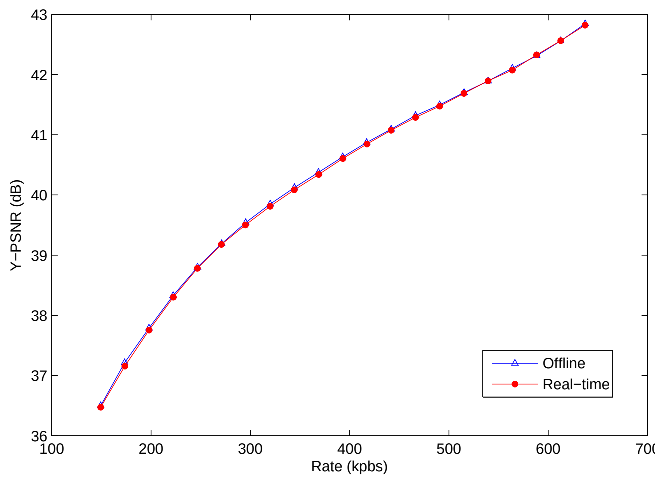


Figure 3.5: The RD curves of real-time Algorithm 4 vs. the off-line PID assignment for the concatenated sequence of *news*, *mother-daughter*, *coastguard* and *mobile*.

example, to use the adaptive updating, a new SEI packet is generated only when the MSE of the LMQ is above a threshold. In this case, the window size to calculate the PIDs can also be made adaptively. For instance, we can reset the statistics whenever a new SEI packet is generated.

In addition, an initial SEI packet should be included at the beginning of the bit stream. Since there is not much statistics, the uniform quantization method can be used.

To test these different options, we developed five different real-time algorithms, whose configurations are summarized in Table 3.3. Our experimental results show that the average R-D performances are more or less the same for all of these algorithms. An example is shown in Fig. 3.5 for a concatenated sequence with several scene changes. It can be shown that there is little difference between the real-time algorithms and the offline global algorithm, making the real-time implementation appealing to practical applications.

3.5 Summary

In this chapter, a fast PID assignment algorithm is proposed for the scalable extension of H.264/AVC, by formulating the problem as the Lloyd-Max quantization of the R-D slopes of the NAL units. The complexity of this method is much lower than the merging-based method in the current JSVM. Since the R-D information is embedded in the PIDs, video multiplexing at the network nodes can be simplified, and our method also achieves better multiplexing performances. In addition, we investigate the real-time implementation of our method, and conclude that there is no loss of performance by using real-time algorithm.

Chapter 4

The SVC bitstream extraction in P2P network

4.1 Introduction

For a P2P streaming network, the bandwidth and content availability are two main constraints. The bandwidth constraint, especially the uploading bandwidth of each peer, is often the bottleneck of the transmission path in the network. The reasons are threefold [42]. Firstly, due to the inherent characteristics of some networks, like ADSL, the uploading capacity is much less than the downloading capacity. Secondly, the uploading capacity is sometimes limited by the save-and-forward capacity of the computers. Thirdly, the end-users may prefer to impose an upper limit on their uploading bit rate for various reasons, such as saving the energy and computational power.

With the rapid improvements of broadband network coverage and the computational power of end-user devices, the third reason is usually the main obstacle that causes the bandwidth limit of P2P uploading capacity. The current P2P streaming schemes do not have effective incentives for users to upload more, because this does not guarantee an improved downloading service, especially if the videos that the users are interested in are not very popular, or are particularly bit-rate-demanding.

The proposed P2P video streaming method in this chapter is based on SVC with MGS. SVC offers a variety of advantages, such as rate adaptivity. Thus it is a promising technique for P2P video streaming. In [48] and [66], the performance of P2P video streaming using SVC is studied, which shows the benefits of a prioritization mechanism to react to network congestion. In [1], the popularity of videos is considered and the layered structure of SVC is exploited for P2P video streaming. In [53], the encoder configuration of SVC combined with multiple description coding for P2P video streaming is investigated. In [44], the author investigates adaptive video streaming in P2P network based on SVC video streaming and evaluates the results compared to conventional AVC video streaming approaches. There are also some other papers which applied SVC to P2P video streaming. However, they only considered existing characteristics of SVC without further exploitation.

In this chapter, we only consider short videos, such as the videos on YouTube, hence we assume each peer has the capacity to buffer the whole video stream within one peer's lifetime. We also assume that each peer can at least offer an uploading bandwidth that exceeds the base-layer bit rate of one video stream. A similar P2P structure was used in [42].

To facilitate the comparison of packets from different video streams, we adopt the previously developed Lloyd-Max quantization (LMQ) based Priority Index (PID) assignment method in Chapter 3. Moreover, to reduce the quality variation among different video streams, we also include in the SEI message the average base-layer quality of the video. This will help the server to make the decision.

Another contribution of this chapter is the development of multi-hierarchical topology for P2P network, which is based on the uploading bandwidth capacity of each peer, the requested video, and the request time instant.

By combining the topology construction method and the modified SVC PID assignment method, a fair video quality that depends on on each user's uploading bandwidth contribution can be achieved. Compared to the PID assignment method in the H.264 SVC reference software [39], significant improvement can be achieved by the proposed method.

4.2 The Modified LMQ-based PID Assignment Method

In Chapter 3, the quantization bin indices are taken as the PIDs and the reconstructed R-D slope values are embedded into a SEI message as an look-up table. By obtaining the reconstructed operational R-D slope values, the scheduler at intermediate network nodes along the transmission pathway can compare the R-D slopes of MGS packets from multiple video streams and decide which packets to drop if there is not enough bandwidth.

In this chapter, to further facilitate bit stream extraction in video multiplexing, we also include the average base-layer PSNR of each Group of Pictures (GOP) in the SEI message. The average PSNR is also easy to get since the PSNR of each frame is usually readily available at the encoder during the encoding process.

With the base-layer PSNR and the reconstructed R-D slope of each PID as the side information, we can use the following algorithm to achieve a fair video multiplexing scheme when multiple video streams need to share a common transmission channel and are expected to get the same quality:

1. Allocate only the base-layer packets (PID=63) to all videos, and find their PSNRs from the SEI packets.
2. Find the video with the lowest PSNR.
3. For PID = 62 : -1 : 0
   (a) Add all MGS packets with the same PID to the current video.
   (b) Update the PSNR of the current video.
   (c) If the total bit rate of all videos exceeds the channel bandwidth constraint, stop the bit allocation algorithm.
   (d) If the current PSNR exceeds the second lowest PSNR, go to Step 2.

Since the R-D slopes of packets with the same PID are identical and can be found from the SEI packet, the PSNR improvement in Step 3.b is simply the product of the R-D slope and the total size of these packets.

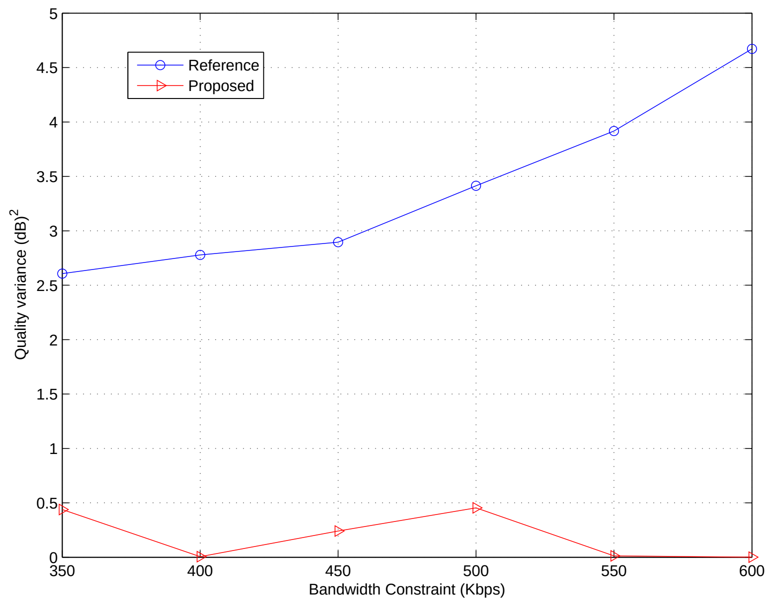


Figure 4.1: PSNR variance of multiplexing News and Coastguard.

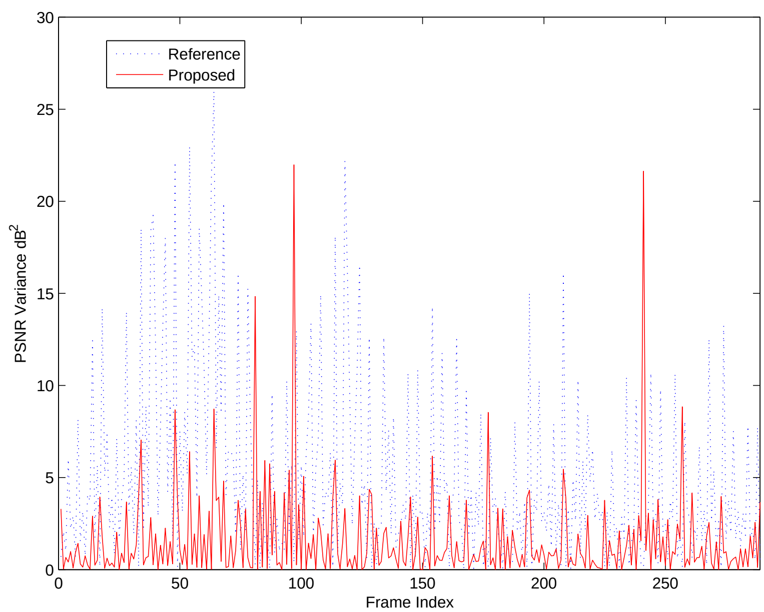


Figure 4.2: The frame-by-frame PSNR variance of multiplexing Coastguard and News.

To demonstrate the performance of this algorithm, Fig. 4.1 shows the video multiplexing results when two videos share a common channel with different bandwidths. The two videos used are News and Coastguard, both have 300 frames. The reference result is obtained using the default PID assignment method in the current JSVM reference software [39]. The bit stream extraction is achieved by keeping adding packets from the two videos with the same PID, starting from the highest PID, until the target bit rate is reached.

It can be seen that the PSNR variance of the proposed method is much less than that of the reference software, due to the improved PID assignment method and the capability of monitoring the PSNR closely in the new method. In the old method, since the PSNRs with the base layer and different enhanced layer packets are not available to the multiplexer, it can only use the PID information to extract the bit stream, but packets from different videos with the same PID do not necessarily have similar R-D contributions.

Fig. 4.2 shows an example of the frame-by-frame PSNR variance of News and Coastguard sequences using the two methods, with a channel bandwidth of 550 kbps. It can be observed that the PSNR variance is much smaller using the proposed method.

4.3 A Multi-Hierarchical Topology for P2P System

4.3.1 Overall Structure

The peer structure in P2P video streaming systems can be classified as cluster-based [68, 70], tree-based [20, 48], mesh-based and hybrid. Cluster-based approach provides good robustness and scalability. Tree-based approach has a pre-determined path and is easy to maintain. These two structures can also be employed together to form a hybrid overlay topology [12, 42]. However, these strucures only consider the resources within a single P2P video streaming overlay.

As mentioned before, we assume the uploading bandwidth is the main constraint for the P2P network. Meanwhile, the downloading bandwidth is usually not proportional to one's contribution, which discourages the end-users to allocate more uploading bandwidth, thereby causing more severe resource starvation. In this paper, we propose a scheme in which the available downloading resources are proportional to one's uploading bandwidth

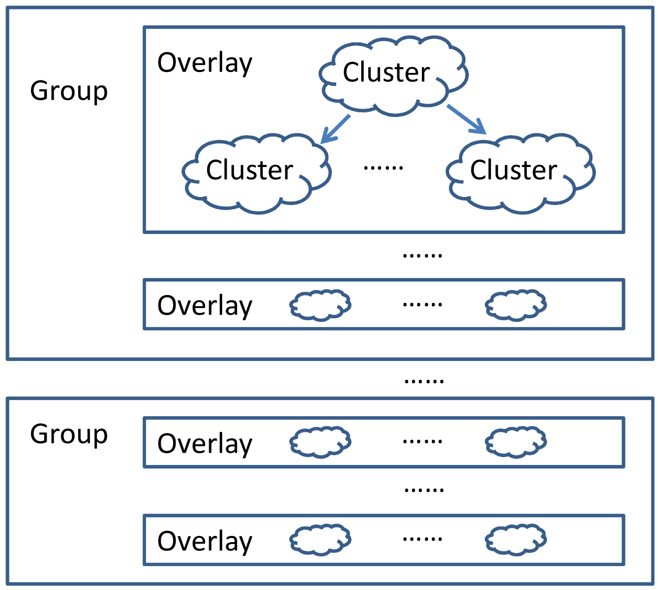


Figure 4.3: The proposed multi-hierarchical topology.

We define a multi-hierarchical topology for the P2P network, as shown in Fig. 4.3. The outer-most layer of the structure is based on the uploading bandwidth of a peer, *i.e.*, peers with similar uploading bandwidths are categorized into the same group. The transmission between peers can only happen within one group. As a result, peers who contribute more uploading bandwidths can benefit from peers who also allocate more uploading bandwidths. Thus the policy enjoys the property of "contributing more, getting more".

Within each group, peers may request different videos, so there are multiple overlays, each corresponds to one video. This constitutes the second hierarchy.

Within each overlay, we divide peers into different clusters, based on the time instants these peers request the video. Peers whose requested time instants are within a certain range are grouped as a cluster. Within the same cluster, there is no determined direction of transmission. Peers can transmit the video to each other as in a mesh network. These clusters form the third hierarchy. Also, peers who request the video earlier should have buffered more video contents; hence they should be put closer to the root in the tree. On the other hand, peers who just join the overlay should be the leaves of the tree because they have less to offer.

The proposed strucuture is mainly based on the hybrid P2P video streaming strucuture as those in [12, 42]. The new features are that we group peers based on their uploading bandwidths, and we also take advantage of the extra uploading bandwidth of other overlays.

4.3.2 Intra-overlay and Inter-overlay transmission

In the proposed topology, there could be more than one overlay within the same uploading bandwidth group. Usually, the videos are transmitted within one video overlay. Since the instant transmission bit rate is determined by the server based on the sum of the uploading bandwidths, in some cases there could be some extra uploading bandwidths for one overlay and some shortages for another overlay, depending on the number of peers in each overlay. Another reason of the imbalance of the uploading bandwidth among overlays is the inherent characteristics of the videos. Videos with a lot of scene changes can be bit-rate-demanding while the bit rate of videos with less activities can be quite low.

To avoid asking for videos directly from the server, the peers with extra uploading capacity can cache some video contents that they are not using and transmit them to the peers from another overlay who are interested in them, as shown in Fig. 4.4. The total uploading rate of intra-overlay and inter-overlay transmission will not exceed one's uploading bandwidth. Once cached, the video can be repeatedly transmitted to different peers from another overlay. The benefit of doing so is to lessen the downloading request from the server and to achieve fair video quality for peers that contribute similar uploading bandwidths but from different overlays.

4.3.3 The topology construction

There are two types of data transmitted in the network. The first type is control messages, which are used for collecting the network condition including the video request, uploading bandwidth information, and topology status. Another type is the actual video contents that the end-users requested to view [42].

Firstly, we assume the server has the full version of every video. It provides the original video contents to peers at the beginning. It also needs to meet downloading requests from the peers when the peers cannot feed themselves. It will be shown later that using the scheme proposed above, this scenario rarely happens. Each peer can get partial or full version of the video based on the instantaneous total uploading bandwidth.

For a peer who raises a video request to the server, the server first obtains the uploading bandwidth of this peer, and assigns the peer into the group that matches its uploading bandwidth and the overlay that corresponds to the requested video. The server then assigns the peer to a cluster in that overlay in which all peers have similar requesting time instants. This cluster is usually at the leaves of the tree.

Within each group, the server records the uploading bandwidths of all peers. Based on the PID and embedded SEI information of the SVC-coded video, the server uses the PID assignment algorithm in Sec. 4.2 to calculate the bit rate for each video, while the instant dowloading bit rate of each peer is constrained to be the same by the server and their sum will not exceed the sum of the uploading bit rate within the group. Since our rate allocation algorithm can achieve similar video quality for all the peers. It is the uploading capacity within the same group that determines the quality of received video for each peer of which it belongs to.

The root cluster within each overlay can only asks for the video content from the server, since there are no other peers who have cached the video content. However, except for the start-up, the root cluster usually contains the peers who have already finished viewing the video and only make uploading contribution. The cluster searches for the resources from higher-level clusters until the required downloading bit rate is reached. The transmission

Table 4.1: The uploading bandwidth probability distribution

ID	Uploading bandwidth (Kbps)	Probability
1	312	0.1
2	340	0.1
3	384	0.3
4	420	0.4
5	512	0.1

direction can only be one-way within a tree, which is from the top to the bottom, because the video requested by a peer in a lower cluster is only cached in the buffers of upper clusters. However, within each cluster, there is a control leader to negotiate with the peers to form a mesh based transmission path since the peers within one cluster requested the video at similar time instants.

4.4 Simulation and Results

We built a simulation environment to test the performance of the proposed P2P system. We assume that in each minute, the number of peers that join the P2P overlay follows the Poisson distribution, and the lifetime of each peer follows the Weibull distribution. We also assume that the uploading bandwidth distribution follows that in Table 4.1, and the downloading bandwidth is not constrained.

For simplicity, we categorize the peers into two groups based on their uploading bandwidths, and the threshold is chosen as 400 Kbps. We use two repeatedly concatenated videos News and Coastguard as the video contents for the two P2P video streaming overlays. The peers whose requesting time are within 1 minute to each others are considered as in the same cluster. The total simulation time is 5000 minutes and the average number of peers in the system is 326.

From the distribution, we can see that the average uploading bit rates are 360 kbps and 438 kbps, respectively. The instant PSNRs of the proposed method is shown in Fig. 4.5, from which we can see that the average PSNR of all peers in each group is proportional to

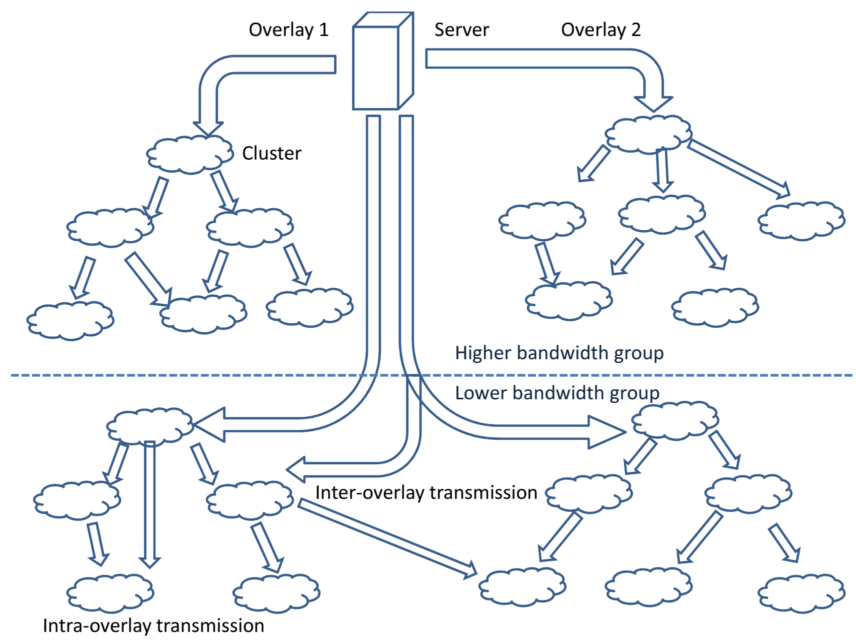


Figure 4.4: The flowchart of P2P video streaming

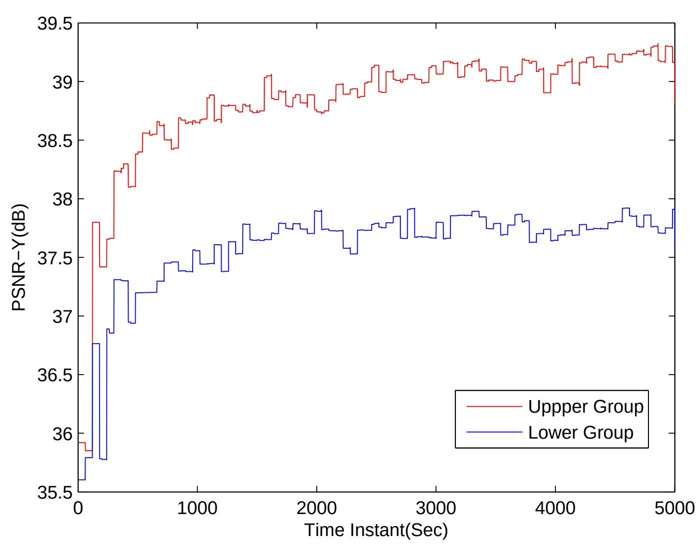


Figure 4.5: The instant average PSNR of two groups in the P2P network.

their uploading bandwidth distribution.

Fig. 4.7 comapres the variance of instant PSNR among peer groups by using the two PID assigning methods. It shows that the PSNR variance with the proposed method is lower than that using the JSVM reference software.

Fig. 4.6 shows the instantaneous server traffic of the whole simulation, from which we can see that, except at the beginning of the simulation, the traffic from the server is zero for most of the time, meaning that the peers can support themselves to stream the videos.

4.5 Conclusion

In this chapter, we first refine the Lloyd-Max Quantization based Priority Index assigning method for the Scalable Extension of H.264/AVC be embedding the base-quality and incremental quality into SEI message in the SVC bitstream. We show this method can be applied to video multiplexing to ensure the fairness of quality among multiple videos. We then develop a multi-hierarchical P2P topology, which can provide a fair "contribute-and-reward" mechanism for a P2P video streaming network. We combine these two parts to build a SVC-based P2P network. The performance of the scheme is demonstrated by simulation results.

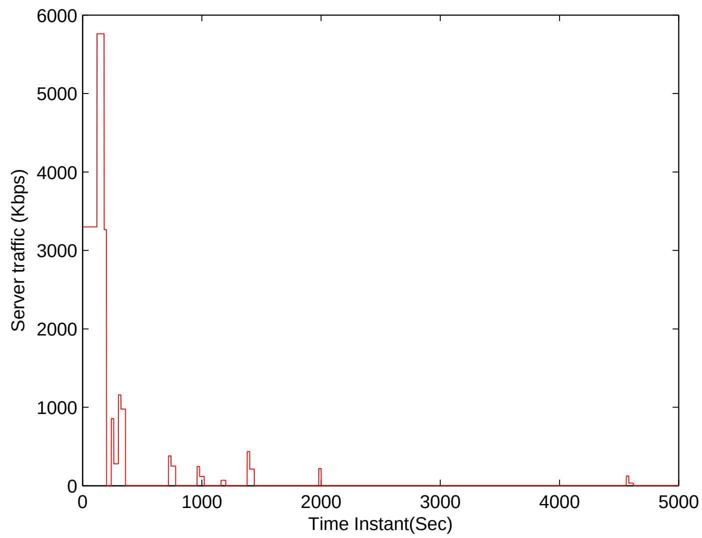


Figure 4.6: The instantaneous server traffic.

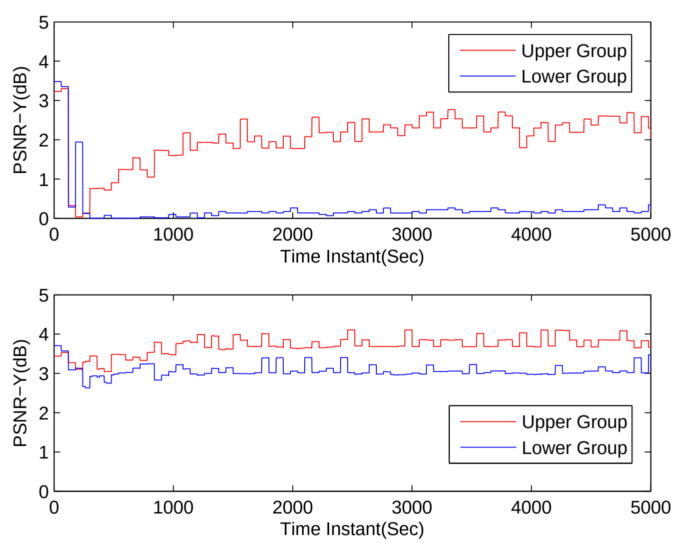


Figure 4.7: The comparison of instant PSNR variance of two groups. (top) By proposed PID method. (Bottom) By the reference method.

Chapter 5

The caching solution for SVC P2P video streaming

5.1 Introduction

In the previous chapters, we make the assumption that all the videos have enough cache space to remain the whole video in one's cache. It is only realistic for short videos like the videos on Youtube.com [59]. If the video streaming scenario is for Video-on-Demand (VoD) which offers 100 mins or longer video content, usually one peer's cache cannot hold the whole video. In this chapter, similar to [57], we also consider a hybrid P2P network that utilizes both media server and control server for VoD video streaming. The difference is that they only take non-scalable video into considerations. When we use the scalable video content like the scalable extension of H.264 [33] for P2P video streaming, the results show that average normalized load from the media server can be further reduced compared to the case when non-scalable video content is used for video streaming. And as far as we know, there is only one paper in literature [1] which investigates the caching optimization problem for scalable video streaming in P2P network. However, its method is greedy ad-hoc solution which does not considers the instant supply-demand relation in the P2P network.

5.2 System Model

In our set-up, each video is also divided into temporally-equal-length segments. There are mainly three components in the architecture: peers, the media server and the control server. Each peer should dedicate one segment-sized local cache and its uploading bandwidth to serve other peers as a "server" of a particular segment. The control server tracks the record of new peer joining and leaving. It also finds a list of peers from which a peer can download the requested segment. We assume it is fully aware of the current peer population as well as the P2P service capacity by identifying each peer's uploading bandwidth. It is also aware of each peer's individual demand and progress during one's video streaming session. Finally, the control server determines the caching policy of each peer and enforces it. The media server provides the original video content to the peers in the initial stage as well as the case when the peers cannot feed themselves and the deficit should be patched by the media server. Here we only consider that the uploading bandwidth is the mainly constraint in the system and the bottleneck is its connection to the backbone network [42].

The video streaming follows a relatively deterministic fashion [57]: from the first segment until the last one. For a long video that consists of many segments, it is reasonable if we estimate the future demand of the peers based on the current demand in the system. Although there are some factors which might cause random fluctuation in demand like skip, fast forward, backward or early termination. If the number in the system is large enough and the video is also long enough, the normal operation will take the most part of all the users and the randomness can also be offset. Therefore, we can still have a good estimate of the future demand.

5.2.1 System Assumptions

In the utility maximization problem, the authors in [57] compare three methods for P2P cache: cache pre-fetching, dynamic programming cache allocation and static optimal cache allocation. The latter two are basically the same since they both adaptively update each peer's cache based on the current supply and demand estimation. The only difference is

Table 5.1: Model parameters of P2P SVC video streaming cache

Variable	Definition
$r_{m,n}$	the rate of video n when the PID larger or equal to m is remained in the bitstream.
$B_{k,n}$	the uploading rate of peer k in video overlay n.
$N_{n,t}$	the total number of peers in the video overlay n, at time t.
$r'_{m,n,t}$	the specified rate of video n when the PID larger or equal to m is remained in the bitstream, at time t
$S_{n,t}$	the total uploading bandwidth of peers in video overlay n, at time t.
$ST_{m,n,t}$	the total server traffic while the video quality is $r'_{m,n,t}$ in video overlay n, at time t.
$N_{n,t}$	the total number of peers in the video overlay n, at time t.
$R_{n,t}$	the total requested rate for all peers in video overlay n, at time t.
f_n	the frame rate of video n
g_n	the gop size of video n
$T_{n,t}$	the media server traffic at time t
s_n	the segment size for cache in the video overlay n
P_n	the interval for updating cache
$I_{k,t}$	the GOP index of segment of peer k, at time t, i=1,2...M
$q_{k,i,t,n}$	the rate of the SVC bitstream in the cache of peer k, whose segment index is i in video overlay n
λ_n	the arrival rate of newly-joined peers in video overlay n

that the step-size for cache updating is different. However, the author of [57] only considers non-scalable video streams. If we use the SVC video streams instead of the non-scalable ones, there is one more dimension of freedom in this optimization problem.

Unlike [57], the supply and demand estimation should also include the rate adaptation of the SVC bit streams since the demand can be self-adjusted due to the rate adaptation of SVC bitstream. For a already-encoded SVC bitstream, if the rate of the bitstream is determined, the output video quality of bitstream is also determined. We will consider the rate of the SVC bitstream $r_{m,n}$ and the quality of SVC bitstream as the same conception in the following.

Table 5.1 list most of the variables which will be used in this chapter.

In the n^th video overlay, we define one peer's segment size for cache, s_n, to be the size of one GOP of SVC bitstream of the possible highest quality. For example, if the frame rate f_n

of the video n is 30 fps, the GOP size g_n is 16 and the highest quality SVC bitstream without truncation is $r_{0,n}$. Then we define the interval for updating cache is $P_n = 16/30 = 0.53sec$. The segment size for caching is defined as the size of 1 GOP of bitstream of the highest quality. Also, $S_{n,t}$ and $R_{n,t}$ are defined as below, which are the sum of uploading bandwidth and the requested rate of the video streams of all peers in the overlay n. $B_{k,n}$ and $r'_{m,n,t}$ are defined in Table 5.1:

$$S_{n,t} = \sum_{k=1}^{N_n}(B_{k,n}) \tag{5.1}$$

$$R_{n,t} = \sum_{k=1}^{N_n}(r'_{m,n,t}) \tag{5.2}$$

The optimization goal is to minimize the average traffic load from the sever defined as $NT(t)$ and to maximize the received video quality $r'_{m,n,t}$ of all peers:

$$NT(t) = \min(avg(S_{n,t}/N_{n,t})); \tag{5.3}$$

$$\max(avg(r'_{m,n,t})) \tag{5.4}$$

The constraints in the optimization problems are the segment size s_n for cache in the video overlay n and the uploading bandwidth of each peer $B_{k,n}$:

$$s_n = P_n \times r_{0,n} \tag{5.5}$$

$$B_{k,n}, k = 1, 2, 3.... \tag{5.6}$$

The parameters which need to be optimized in this problem are the bit rate of the video stream $r'_{m,n,t}$ (which is proportional to the video quality) at each time instant t as well as the caching decision $I_{k,t}$ of each peer k at time t:

$$r'_{m,n,t}, m = 0, 1, 2...63; \tag{5.7}$$

$$I_{k,t}, k = 1, 2, 3....N_{n,t} \tag{5.8}$$

There is a trade-off between the server traffic $ST_{m,n,t}/N_{n,t}$ and the video quality $r'_{m,n,t}$ of all the peers. Our goal is to find the optimal points in the $ST_{m,n,t}/N_{n,t}$ versus $r'_{m,n,t}$ curve. For simplicity, we consider all the peers receive the same output video quality $r'_{m,n,t}$ during the same time slot. However, the output video quality can vary in different time slots.

5.2.2 Supply and Demand Estimation

For the supply estimation, we suppose that the current quality of SVC bitstream is $r'_{m,n,t}$. For each segment index i, we search all the peers whose cache is exactly segment i at time t. Then we can divide the peers which caches the segment i into two categories. The peers whose cache has the SVC bitstream rate lower than $r'_{m,n,t}$ is in category $P_{low,i}$, and the peers whose cache has the SVC bitstream rate higher than $r'_{m,n,t}$ is in category $P_{high,i}$. Here $q_{k,i,t,n}$ is the rate of the SVC bitstream in the cache of peer k, whose segment index is i in video overlay n.

We define $L(m, n, i, t)$ as the number of copies of the video segments in the system

$$L(m, n, i, t) = \sum_{k \epsilon P_{low,i}} B_{k,n}/q_{k,i,t,n} \tag{5.9}$$

and $G(m, n, g, t)$ as the gap of bit rate for peers from $P_{low,i}$ to achieve quality $r'_{m,n,t}$.

$$G(m, n, i, t) = \sum_{k \epsilon P_{low,i}} (r'_{m,n,t} - q_{k,i,t,n}) \tag{5.10}$$

$H(m, n, i, t)$ is the sum of uploading bandwidth of peers in $P_{high,i}$.

$$H(m, n, i, t) = \sum_{k \epsilon P_{high,i}} B_{k,n} \tag{5.11}$$

For segment index i in video n at time t, we define $Gap(n, i, t, m)$ as the gap of bit rate which can not be obtained from other peers and need to download from the media server if the requested SVC video quality is $r'_{m,n,t}$.

if $H(m,n,i,t) > G(m,n,i,t)$, the rate of the peers from $P_{high,i}$ can compensate the gap of peers in $P_{low,i}$. So the total supply $Sup(m,n,i,t)$ of the i^th GOP of SVC bitstream with quality $r'_{m,n,t}$ at time t is

$$Sup(m,n,i,t) = L(m,n,i,t) \times r'_{m,n,t} + H(m,n,i,t) - G(m,n,i,t); \tag{5.12}$$

while there is no conditional extra downloading bitrate $Gap(m,n,i,t)$ from the media server

$$Gap(m,n,i,t) = 0; \tag{5.13}$$

Otherwise, if the $H(m,n,i,t) < G(m,n,i,t)$ then the total copies of the i^th GOP of SVC bitstream with quality $r'_{m,n,t}$ at time t is

$$Sup(m,n,i,t) = L(m,n,i,t) \times r'_{m,n,t}; \tag{5.14}$$

while the conditional extra downloading bitrate from the media server for quality $r'_{m,n,t}$ is

$$Gap(m,n,i,t) = G(m,n,i,t) - H(m,n,i,t); \tag{5.15}$$

For the demand estimation, we can divide the demand into two parts: the stochastic part and the the non-stochastic parts [57].

For the non-stochastic part, for a specified requested quality $r'_{m,n,t}$, we consider the demand $Dem(m,n,i,t)$ of segment i at time t of quality m in video overlay n is

$$Dem(m,n,i,t) := Dem(m,n,i-1,t-1), i = 1,2,3... \tag{5.16}$$

For the stochastic part, the number of peers newly joined the system is considered as stochastic process. In most previous study , it is usually assumed that it follows a Poisson distribution and the demand $Dem(m,n,i,t)$ of segment i at time t of quality m in video overlay n is

$$Dem(m,n,0,t) := r'_{m,n,t} \times \lambda \tag{5.17}$$

where λ is the arrival rate of newly joined peers in the system.

Now we can calculate the server traffic. For a specified requested quality $r'_{m,n,t}$, the server traffic $ST_{m,n,t}$ is defined as

$$ST_{m,n,t} = \sum_{i=1}^{M}(Dem(m,n,i,t) - Sup(m,n,i,t) + Gap(m,n,i,t)) \div N_{n,t} \tag{5.18}$$

so for each time slot, for a specified $r'_{m,n,t}$, the server traffic can be estimated with the equations stated above. The control server will search all the $r'_{m,n,t}$ to get the best quality under the minimum server traffic.

Now we can prove that if $r'_{m,n,t} = arg r'_{m,n,t} \times N_{n,t} \approx S_{n,t}$, it is the optimal point in the $ST_{m,n,t}$ versus $r'_{m,n,t}$ curve.

Lemma 1: For the PID m, if

$$r^{opt}_{m,n,t} = \lfloor S_{n,t}/N_{n,t} \rfloor \tag{5.19}$$

it is the optimal solution in the $ST_{m,n,t}$ versus $r'_{m,n,t}$ curve.

The reasoning is obvious. If we want to require additional layers of SVC video stream higher than $r^{opt}_{m,n,t}$ to get higher quality, we need to ask from the media server since the uploading capacity is the constraint for peers to download higher SVC layers from other peers. So the higher SVC quality layers are all downloaded from the media server. Once higher layers beyond the uploading capacity were cached in the peers, they can not be transmitted to other peer either due to the constraints on uploading bandwidth.

Also, if $r'_{m',n,t}$ is lower than $r^{opt}_{m,n,t}$, there should exist a caching strategy so that the segment supply is higher than the segment demand to let $\sum_{i=1}^{M} Dem(m,n,i,t) - Sup(m,n,i,t) + Gap(m,n,i,t) < 0$. So the uploading bandwidth is not fully utilized. Thus we can increase $r'_{m',n,t}$ to increase the output video quality without increase the server traffic.

5.2.3 Caching method

Now we consider how to make the cache decision for each peer to balance the segment supply and segment demand in the P2P system. Since the dynamic programming optimal caching method is the best one in [57], here we only consider the dynamic programming method in [57].

We define H as the number of steps in the cache decision path searching tree [57]. In other words, we estimate H future time instant of supply and demand to determine the caching decision of each peer. For each step, we need to specify the rate from (5.19) first. For future decisions, $S_{n,t+l}, l = 1, 2...H$ and $N_{n,t+l}, l = 1, 2...H$ are also not deterministic. Therefore, we also need to estimate them first. We define the number of peers who will leave the system in the next l time-step as $N_{leave,l}$ and their uploading bandwidth sum as $S_{leave,l}$. These two terms are deterministic since we make the assumption that the peers will leave the system immediately once they finished viewing the last segment.

$$S_{n,t+l} := S_{n,t} + l \times E[B_{k,n}] - S_{leave,l}; \tag{5.20}$$

$$N_{n,t+l} := N_{n,t} + l \times \lambda - N_{leave,l}; \tag{5.21}$$

Then $m' = arg(r'_{m',n,t+1}), l = 1, 2...H$ can be specified based on the estimated $S_{n,t+l}$ and $N_{n,t+l}$.

We denote $P_{k,H} = I_{k,t}, I_{k,t+1}, ...I_{k,t+H-1}$ as a caching path. The utility $U(P_{k,H}, H)$ function of $P_{k,H}$ is defined as

$$U(P_{k,H}, H) = \sum_{l=1}^{H} \min(Dem(m, n, I_{k,t+l}, t + l) - Sup(m, n, I_{k,t+l}, t + l) + Gap(m, n, i, t)), B_{k,n}) \tag{5.22}$$

And the optimization goal for each peer's cache is to find the path to maximize the utility. Therefor, it is a real-time strategy to optimize each peer's caching decision to achieve the minimum $NT(t)$

$$P_{k,H} = arg \max U(P_{k,H}, H) \tag{5.23}$$

Table 5.2: The normalized traffic between SVC and AVC

Sequence	low-end AVC	high-end AVC	low-end SVC	high-end SVC
Coastguard	20.3%	14.9%	10.6%	10.1%
News	27.6%	18.2%	16.6%	13.3%

5.3 Performance Evaluation

The normalized traffic is defined as the average bitrate per user from the media server divided by the bitrate of the received video stream. The testing video sequences are *Coastguard* and *News*. We set up two simulation scenarios with different uploading bandwidth distribution the low-end one is 192, 256, 384 kbps with the probability of 0.2, 0.2, 0.6. The high-end one is 256, 384, 512kbps with probability 0.2, 0.2, 0.6. The results are shown in Table 5.2.

The Fig 5.1 shows a snapshot of the normalized traffic $NT(t)$ from the media server per user of 10 mins of the video sequence *Coastguard* in low-end uploading bandwidth distribution.

From Table 5.2, we can see the performance is constrained by two factors: the caching algorithm itself and the service capacity of the network. For *Coastguard*, the two cases of *NT(t)* for SVC is almost the same due to the caching algorithm itself whereas the *NT(t)* for *News* is loosened for both AVC and SVC while the service capacity is increased. It indicates the service capacity is the main constraint for this scenario.

5.4 Conclusion

In this chapter, we generalize the supply-demand-based cache optimization into scalable video streaming for P2P network. It shows that the average load per peer is further reduced compared with non-scalable video streaming. The scalable video streaming can further increase the scalability of the P2P video streaming network.

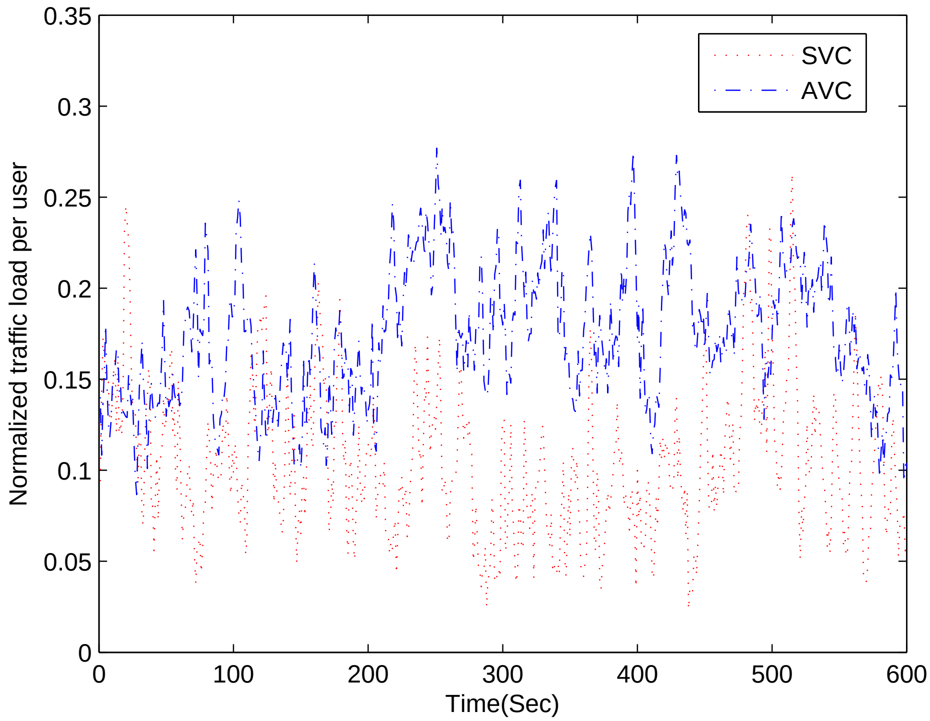


Figure 5.1: The normalized traffic load per user VS Time

Chapter 6

Conclusions and future work

6.1 Conclusions

In this thesis, we firstly formulated the PID assignment as a Lloyd-Max quantization (LMQ) problem and proposed a fast algorithm to assign PID to MGS NALUs. Then we refined the PID assignment method and applied it to a P2P network. Finally, we proposed a caching method for the P2P scalable video streaming.

The new method facilitates the comparison of NALUs from different video streams, with the help of SEI packets. This makes it useful for applications such as network-based video multiplexing. The algorithm can also be implemented in realtime and assign the PIDs adaptively based on the recent statistics of the video. Experimental results show that our method achieve the same R-D performance as the current JSVM in the extraction of one video bit stream while multiplexing different videos, our method can achieve up to 1 dB of improvement. In addition, the real-time implementation of the algorithm does not sacrifice the performance of the off-line method.

We refine the above PID assignment scheme by embedding the absolute base-layer quality as well as the Lloyd-Max quantization-based PID into the SEI massage of SVC bit streams to achieve fair video quality among multiple video streams for scenarios like video multiplexing when bandwidth of the shared channel are restricted. The variance of the video quality among multiple video streams is largely decreased by applying this method. We also adopt a uploading bandwidth categorization of peers to build a fair contribute and reward mechanism for a P2P video streaming network. When the proposed PID assignment method is applied to this P2P network, end-users can obtain fair video quality based on its uploading bandwidth contribution while fully utilizing the quality scalability of SVC bit streams.

We also propose a caching mechanism for SVC video streaming in the P2P network based on the instant supply and demand estimation. We generalize the previous work from the literature into the SVC case and get the estimation formulation of the supply and demand in SVC video streaming on P2P network. We found that SVC video streaming can further reduce the normalized traffic load from the server compared with AVC video streaming in the same caching strategy set-up.

6.2 Future work

The proposed SVC bitstream extraction method can be improved and further applied to other areas.

6.2.1 Different distortion metrics

The current distortion only considers the PSNR as the measure of distortion. However, PSNR is only a objective metrics which can not always reflect human's subjective feeling of the video quality. If the distortion metrics is changed into another one, the whole slope information should be reformulated as well as the correlation between the distortion metrics and the scalability of SVC bitstream can be further exploited.

6.2.2 Different network

For different networks, each has its unique feature which need to be considered when SVC video streaming is applied to. So there is still a lot of work to do if SVC video streaming is applied to different network like 3G Satellite Networks.

Bibliography

[1] A. Bezerra, C. Melo, D. Vieira, Y. Ghamri-Doudane and N. Fonseca. A content-oriented web cache policy. In *LATINCOM '09. IEEE Latin-American Conference on Communications*, Medellin, 10-11 Sept. 2009.

[2] A. Detti, P. Loreti, N. Blefari-Melazzi and F. Fedi. Streaming H.264 scalable video over data distribution service in a wireless environment. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, Montreal, QC, Canada, 03 Aug. 2010.

[3] A. Naghdinezhad, M.R. Hashemi and O. Fatemi. A Novel Adaptive Unequal Error Protection Method for Scalable Video over Wireless Networks. In *Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on*, 2007.

[4] A. T. Nguyen, B. Li and F. Eliassen. Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding. *INFOCOM, 2010 Proceedings IEEE*, 2010.

[5] A. Vetro, C. Christopoulos, H. Sun. Video transcoding architectures and techniques: an overview. *Signal Processing Magazine, IEEE*, 20(2):18–29, 10 March 2003.

[6] B. Chang, L. Dai, Y. Cui and Y. Xue. On Feasibility of P2P On-Demand Streaming via Empirical VoD User Behavior Analysis. *In Proc. of ICDCS'08 Workshop.*

[7] B. Cheng, L. Stein, H. Jin, X. Liao and Z. Zhang. GridCast: Improving Peer Sharing for P2P VoD. *ACM Transactions on Multimedia Computing, Communications and Applications*, 4(4):131, Oct. 2008.

[8] B. Gorkemli and M. Reha Civanlar. SVC Coded Video Streaming over DCCP. In *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, San Diego, CA, Dec. 2006.

[9] C. Chi, Y. Zhang and L. Wang. Joint Power Control and FEC Unequal Error Protection for Scalable H.264 Video Transmission over Wireless Fading Channels. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, Honolulu, HI, 04 Mar. 2010.

[10] C. Li, C. Yuan and Y. Zhong. A novel substream extraction for Scalable Video Coding over P2P networks. *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, 03:1611–1615, 2009.

[11] C. Lin, J. K. Zao, W. Peng, C. Hu, H. Chen and C. Yang. Bandwidth Efficient Video Streaming Based Upon Multipath SVC Multicasting. In *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*, Crete Island, 15 Aug. 2008.

[12] D. A. Tran, K.A. Hua and T. T. Do. A peer-to-peer architecture for media streaming. *IEEE Journal on Selected Areas in Communication*, 22:121–133, Jan. 2004.

[13] D. Lee, Y. Lee, H. Lee, J. Lee and H. Shin. Determining Efficient Bit Stream Extraction Paths in H.264/AVC Scalable Video Coding. *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 2233–2237, 2008.

[14] D. Munaretto, D. Jurca and J. Widmer. Scalable video broadcast in cellular networks: impact on QoS and network resources. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, Riccione, Italy, 12 Aug. 2010.

[15] D. Munaretto, J. D. Urca and J. Widmer. Broadcast video streaming in cellular networks: An adaptation framework for channel, video and AL-FEC rates allocation. In *Wireless Internet Conference (WICON), 2010 The 5th Annual ICST*, Singapore, 1-3 Mar. 2010.

[16] D. Song and C. W. Chen. Scalable H.264/AVC Video Transmission Over MIMO Wireless Systems With Adaptive Channel Selection Based on Partial Channel Information. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17:1218–1226, 2007.

[17] D. Taubman and M. Marcellin. *JPEG 2000 Image Compression Fundamentals, Standards and Practice*, chapter 3, pages 98–101. Kluwer Acedemic Publisher, 2002.

[18] E. Maani and A. Katsaggelos. Optimized Bit Extraction Using Distortion Modeling in the Scalable Extension of H.264/AVC. *IEEE Trans. Image Proc.*, 18(9):2022 – 2029, Sep. 2009.

[19] E. Maani and A.K. Katsaggelos. Unequal Error Protection for Robust Streaming of Scalable Video Over Packet Lossy Networks. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20:407–416, 2010.

[20] E. Setton, P. Baccichet and B. Girod. Peer-to-peer live multicast: A video perspective. *Proceedings of the IEEE*, 96:25–38, Jan. 2008.

[21] G. Gomez-Barquer, D. mez-Barquero, K. Nybom, D. Vukobratovic and V. Stankovic. Scalable video coding for mobile broadcasting DVB systems. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, Suntec City, 19-23 Jul. 2010.

[22] G. Sun. Multiple Description Coding with Prediction Compensation. Master's thesis, Simon Fraser university, 2009.

[23] G. Van der Auwera and M. Reisslein. Implications of Smoothing on Statistical Multiplexing of H.264/AVC and SVC Video Streams. *Broadcasting, IEEE Transactions on*, 55:541–558, 2009.

[24] H. Du, X. Huang, J. Liang, J. Liu, B. G. Evans and I. Chlamtac. Cross-layer Quality-driven Adaptation for Scheduling Heterogeneous Multimedia over 3G Satellite Networks. *ACM/Springer Wireless Networks Journal (WINET)*, 16(4):1143–1156, 2010.

[25] H. Ha and C. Yim. Layer-weighted unequal error protection for scalable video coding extension of H.264/AVC. *Consumer Electronics, IEEE Transactions on*, 54:736–744, 2008.

[26] H. Schwarz, D. Marpe and T. Wiegand. Closed-loop coding with quality layers. In *JVT-Q030, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, Nice, Oct. 2005.

[27] H. Schwarz, D. Marpe and T.s Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. Circ. Syst. Video Tech.*, 17(9):1103–1120, Sep. 2007.

[28] H. Wang, S. Xiao and C.-C.J. Kuo. Towards an Improved SVC-to-AVC Rewriter. In *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, Washington, DC, 26-30 Nov. 2007.

[29] http://en.wikipedia.org/wiki/Peak_signal-to noise_ratio.

[30] http://www.cse.ohio-state.edu/hpcs/WWW/HTML/internet p2p.html.

[31] I. Amonou and N. Cammas and S. Kervadec and S. Pateux. Layered quality optimization of JSVM3 and closed loop. In *JVT-Q081, Joint Video Team (JVT) of ISO-IEC MPEG and ITU-T VCEG*, Nice, Oct. 2005.

[32] I. Amonou, N. Cammas, S. Kervadec and S. Pateux. Spatial Scalability Within the H.264/AVC Scalable Video Coding Extension. *IEEE Trans. Circ. Syst. Video Tech.*, 17(9):1121–1135, Sep. 2007.

[33] I. Amonou, N. Cammas, S. Kervadec, S. Pateux. Optimized Rate-Distortion Extraction With Quality Layers in the Scalable Extension of H.264/AVC. *IEEE Trans. Circ. Syst. Video Tech.*, 17(9):1186–1193, Sep. 2007.

[34] I. Kofler, R. Kuschnig, H. Hellwagner. In-Network Real-Time Adaptation of Scalable Video Content on a WiFi Router. In *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, Las Vegas, NV, 18 Feb. 2009.

[35] J. G. Apostolopoulos. Architectural Principles for Secure Streaming & Secure Adaptation in the Developing Scalable Video Coding (SVC) Standard. In *Image Processing, 2006 IEEE International Conference on*, Atlanta, GA, 20 Feb. 2007.

[36] J. Kim, J. Koo and K. Chung. T-NASS: TCP-friendly network adaptive SVC streaming protocol. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, Beijing, 11 Sep. 2009.

[37] J. Liu, S. Rao, B. Li and H. Zhang. Opportunities and challenges of P2P internet broadcast. *Proc. IEEE*, 96(1), 2008.

[38] J. M. Monteiro, C. T. Calafate and M. S. Nunes. Evaluation of the H.264 Scalable Video Coding in Error Prone IP Networks. *Broadcasting, IEEE Transactions on*, 54:652–659, 2008.

[39] J. Vieron. Draft reference software for SVC. In *JVT-AB203, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, Hannover, Jul. 2008.

[40] J. Wu, J. Tu and C. Hsu. QoS optimized cross multi-layer transmission for SVC coded video stream over IEEE 802.16e network. In *Advanced Networks and Telecommunication Systems (ANTS), 2009 IEEE 3rd International Symposium on*, New Delhi, 14-16 Dec. 2009.

[41] M. A. Isnardi. Historical Overview of Video Compression in Consumer Electronic Devices. *Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on*, pages 1–2, 2007.

[42] M. FESCI-SAYIT, E. T. TUNALI and A. M. TEKALP. BANDWIDTH-AWARE MULTIPLE MULTICAST TREE FORMATION FOR P2P SCALABLE VIDEO STREAMING USING HIERARCHICAL CLUSTERS. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, Cairo, Nov. 2009.

[43] M. Guarisco, H. Rabah and A. Amira. Dynamically reconfigurable architecture for real time adaptation of H264/AVC-SVC video streams. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, San Francisco, CA, 13-18 Jun. 2010.

[44] M. Mushtaq and T. Ahmed. Smooth Video Delivery for SVC Based Media Streaming Over P2P Networks. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, Las Vegas, NV, 10-12 Jan. 2008.

[45] M. Sablatschan, M. Ransburg and H. Hellwagner. Towards an Improved SVC-to-AVC Rewriter. In *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on*, Athens, 13-19 Jun. 2010.

[46] M. Shoaib and M. Waheed. Streaming Video in Cellular Networks Using Scalable Video Coding Extension of H.264-AVC. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, Dalian, 18 Nov. 2008.

[47] M. Wien, H. Schwarz and T. Oelbaum. Performance Analysis of SVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17:1194–1203, 2007.

[48] P. Baccichet, T. Schier, T. Wiegand and B. Girod. Low-delay peer-to-peer streaming using scalable video coding. In *IEEE Packet Video Workshop*, 2007.

[49] PPLive. http://www.pplive.com/.

[50] R. Li, J. Sun and W. Gao. Fast weighted algorithms for bitstream extraction of SVC Medium-Grain scalable video coding. *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 249–254, 2010.

[51] S. Wenger, Y. Wang and T. Schierl. Transport and Signaling of SVC in IP Networks. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17:1164–1173, 2007.

[52] S.K. Srinivasan, J. Vahabzadeh-Hagh, M. Reisslein. The Effects of Priority Levels and Buffering on the Statistical Multiplexing of Single-Layer H.264/AVC and SVC Encoded Video Streams. *Broadcasting, IEEE Transactions on*, 56, 27 May 2010.

[53] T. B. Abanoz and A. M. Tekalp. Optimization of encoding configuration in scalable multiple description coding for rate-adaptive P2P video multicasting. In *2010 16th IEEE International Conference on Image Processing*, Cairo, 7-10 Nov. 2009.

[54] T. Rusert and J. R. Ohm. Backward Drift Estimation with Application to Quality Layer Assignment in H.264/AVC Based Scalable Video Coding. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 1:I–653–I–656, 2007.

[55] S. Mirta K. Gruneberg T. Schierl, C. Hellge and T. Wiegand. Using H.264/AVC-based Scalable Video Coding (SVC) for Real Time Streaming in Wireless IP Networks. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, New Orleans, LA, 25 Jun. 2007.

[56] T. Schierl, K. Ganger, C. Hellge, T. Wiegand and T. Stockhammer. SVC-based multisource streaming for robust video transmission in mobile ad hoc networks. *Wireless Communications, IEEE*, 13:1536–1284, 2006.

[57] U. C. Kozat, O. Harmanci, S. Kanumuri, M. U. Demircin and M. R. Civanlar. Peer Assisted Video Streaming With Supply-Demand-Based Cache Optimization. *Multimedia, IEEE Transactions on*, pages 494 – 508, Apr. 2009.

[58] UUSee. http://www.uusee.com/.

[59] X. Cheng, J. Liu and C. Dale. Understanding the Characteristics of Internet Short Video Sharing: A YouTube-based Measurement Study. *IEEE Transactions on Multimedia, 2010*, 2010.

[60] X. Huang, J. Liang, H. Du and J. Liu. Lloyd-Max quantization-based priority index assignment for the scalable extension of H.264/AVC. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, Paris, Jun. 2010.

[61] X. Huang, Y. Ding, J. Liang and J. Liu. A Fair P2P Scalable Video Streaming Scheme Using Improved Priority Index Assignment and Multi-Hierarchical Topology. In *2011 International Conference on Image Analysis and Recognition (ICIAR)*, Burnaby, BC, Canada, Jun. 2011.

[62] X. Ji, J. Huang, M. Chiang, G. Lafruit and F. Catthoor. Performance Analysis of SVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19:1549–1555, 2009.

[63] Y. Ding. PEER-TO-PEER 3D/MULTI-VIEW VIDEO STREAMING. Master's thesis, Simon Fraser University, 2011.

[64] Y. Liu, S. B. Qaisar, H. Radha and A. Men. On unequal error protection with low density parity check codes in scalable video coding. In *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, Baltimore, MD, 18-20 March 2009.

[65] Y. Liu, S. Zhang, S. Xu and Y. Zhang. Research on H.264/SVC compressed video communication in 3G. In *Computer Science & Education, 2009. ICCSE '09. 4th International Conference on*, Nanning, 25-28 Jul. 2009.

[66] Y. Sanchez, Y. Nchez, T. Schierl, C. Hellge and T. Wiegand. P2P group communication using Scalable Video Coding. In *2010 17th IEEE International Conference on Image Processing*, Hong Kong, 26-29 Sep. 2010.

[67] Y. Wang, L. Chau and K. Yap. GOP-based unequal error protection for scalable video over packet erasure channel. In *Broadband Multimedia Systems and Broadcasting, 2008 IEEE International Symposium on*, Las Vegas, NV, Mar. 31 2008-Apr. 2 2008.

[68] Y. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis and S. Wenger. System and Transport Interface of SVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17:1149–1163, 2007.

[69] Z. Avramova, D. De Vleeschauwer, K. Spaey, S. Wittevrongel, H. Bruneel and C. Blondia. Comparison of simulcast and scalable video coding in terms of the required capacity in an IPTV network. In *Packet Video 2007*, Lausanne, Switzerland, 06 Dec. 2007.

[70] Z. Xiang, Q. Zhang, W. Zhu, Z. Zhang, Y.Q. Zhang. Peer-to-peer based multimedia distribution service. *IEEE Transactions on Multimedia*, 16:343–355, Jan. 2008.