Adaptive Video Streaming over Peer-to-Peer Networks

by

Kadir Tolga Bağcı

A Thesis Submitted to the

Graduate School Sciences and Engineering

in Partial Fulfillment of the Requirements for

the Degree of

Master of Science

in

Electrical and Electronics Engineering

Koç University

September 2012

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Kadir Tolga Bağcı

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

____

Prof. A. Murat Tekalp (Advisor)

____

Prof. Özgür Barış Akan

____

Assoc. Prof. Öznur Özkasap

Date: ______________________________

To my beloved wife…

ABSTRACT

Although most popular video services are based on the server-client architecture, growing need for scalability of bandwidth capacity, processing power and storage space makes P2P solutions a promising alternative. However, P2P has its own drawbacks, such as peer unpredictability, since the bandwidth of the peers may change over time or they may simply quit without any graceful notification. Solutions such as scalable video coding (SVC) and multiple description video coding (MDC) have been proposed for better adaptation to unpredictable channel/peer conditions but complete and realistic results are rarely reported over current and future P2P systems and protocols. Most results reported in the literature either have unrealistic P2P protocols such as centralized tree-based solutions, or they employ very basic encoding approaches with limited results from both network utilization and end-user video quality perspectives. In this thesis, we first propose a new chunk scheduling method based on buffer driven prioritization of layered chunks for better adaptation and higher network utilization. Second, we evaluate use of higher number of SVC layers in pull-based P2P streaming. Third, we extensively evaluate the performances of SVC and MDC video steaming over P2P networks, and provide answers to key questions such as if and when SVC and MDC methods should be preferred over P2P networks.

ÖZETÇE

Popüler video servislerinin çoğunluğu sunucu-kullanıcı mimarisine dayalı olsa da, bant genişliği kapasitesinin ölçeklenebilirliğine, işlem gücüne ve depolama kapasitesine olan ihtiyaçların artması, eşler arası (P2P) çözümleri gelecek vaadeden alternatifler haline getirmiştir. Fakat P2P teknolojilerin kendine özgü zorlukları ve engelleri vardır. Eşlerin sahip oldukları bant genişliklerinin zamanla değişebilmesi veya eşlerin habersizce sistemden çıkabilmeleri bu zorluklara örneklerdir. Ölçeklenebilir video kodlama ve çoklu betimle video kodlama teknikleri, öngörülemez eş ve kanal durumlarına karşı daha iyi adaptasyon sağlamak amacı ile tasarlanmıştır. Fakat bu video kodlama tekniklerini kullanarak, bugünkü ve gelecekteki P2P sistemleri ve protokollerine dayanan eksiksiz ve gerçekçi araştırma sonuçları çok nadiren bildirilmiştir. Yapılan araştırmaların çoğunluğu merkezi-ağaç topolojilerini kullananlar gibi ya gerçek dışı P2P protokollere dayanıyor ya da basit video kodlama tekniklerini kullanıyor. Bununla beraber, yayınlanan sonuçlar da hem ağın genel durumu bakımından hem de her bir kullanıcının ya da eşin aldığı görüntü kalitesi bakımından oldukça sınırlı. Bu tezde, ilk olarak iyi bir adaptasyon sağlayan ve ağdan daha çok faydalanan ara bellek güdümlü parça isteği zaman tablosu oluşturma yöntemi öneriyoruz. İkinci olarak, ölçeklenebilir video kodlamada iki katman ve daha çok katman kullanımının P2P video akıtımındaki başarımlarını inceliyoruz. Üçüncü ve son olarak, ölçeklenebilir video kodlama ve çoklu betimle video kodlama tekniklerinin P2P ağlar üzerindeki başarımlarını kapsamlı deneyler ile değerlendiriyoruz ve hangi durumda hangi video kodlama tekniği tercih edilmelidir gibi önemli anahtar sorulara cevaplar veriyoruz.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and profound respect to Prof. A. Murat Tekalp for his excellent advisory, inspiration and reliable guidance. I am also grateful to members of my thesis committee Prof. Özgür Barış Akan and Assoc. Prof. Öznur Özkasap for their valuable comments and guidance regarding the future of the thesis work.

First and foremost, I would like to thank Göktuğ who always helped me and provided his genuine guidance, inspiration and everlasting motivation for both academic and social life. It would not have been possible to write this thesis without his knowledge, and my last two years would not have been that confident and fluent without his brotherly support. Next, I specially would like to thank my friends Ümit and Yalçın for their sincere brotherhood. I also thank to my lab mates Hilmi, Tahsin, Yusuf, Serkan, Tuğtekin, Sedef and Elif for beautifully remembered funny memories. Gratitudes to all my friends making Koç unforgettable: Burak, Duygu, Remziye, Şaziye, Baran, Özerk, Bekir, Pelin, Merve E., Merve G., Meltem, Nihan, Mali, Emre, Enis, İrem and Seyhan. I also would like to thank TÜBİTAK and European Union FP7 for providing financial support for my research.

Last but not least, I would like to thank to my mother Gülseren and my father Hüseyin for their love and support. Moreover, I am grateful to my sister Tuğba and brother in law Tamer for their significant support and motivation. Finally, my everlasting love to my beloved wife Şebnem to whom I dedicated this work. There would be none of this without her endless support and patience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

ARQ	Automatic Repeat Request
AVC	Advanced Video Coding
CDN	Content Distribution Networks
CGS	Coarse Grain Scalability
DCT	Discrete Cosine Transform
FEC	Forward Error Correction
FPS	Frame per Second
GOP	Group of Picture
IP	Internet Protocol
JVT	Joint Video Team
LC	Layered Coding
MBR	Multiple Bitrate Coding
MDC	Multiple Description Coding
MPEG	Moving Picture Experts Group
NALU	Network Abstraction Layer Unit
P2P	Peer-to-Peer
PSNR	Peak Signal to Noise Ratio
RD	Rate-Distortion
RTT	Round Trip Time
QoE	Quality of Experience
SNR	Signal-to-Noise Ratio
SVC	Scalable Video Coding
TCP	Transmission Control Protocol
VCL	Video Coding Layer
VoD	Video on Demand

Chapter 1

INTRODUCTION

1.1 Motivation

During the last decade, the Internet has become a platform over which multimedia content can be exchanged as easily as text data. For example, YouTube reported that 24 hours of video were uploaded to its servers every minute [1]. Today, the majority of the Internet services are based on the server-client architecture. However, due to increasing data sizes of multimedia applications and growing demand for these applications, conventional server-client systems become inadequate for serving the clients without delays. Content distribution networks (CDN) is the most common solution to overcome bandwidth scarcity problem by deploying multiple content servers that are geographically distributed. As an alternative to CDNs, peer-to-peer (P2P) offers great solutions for lower costs with its increasing popularity. In P2P, the task of servers in conventional server-client systems, which is to provide data to multiple clients, is distributed among peers where each peer acts as both the server and the client at the same time so that upload capacity of each peer is utilized. This approach is highly attractive for server owners since the available bandwidth scales with the demand. According to a Cisco report, P2P traffic is expected to double to 7 petabytes per month in the next four years [2]. On the same report, it is stated that even with this huge growth on P2P traffic (due to file exchange), the corresponding percentage of the IP traffic will drop from 39% in 2010 to 17% by 2014, due to the dramatic increase of the video traffic. From these motivations, it is obvious that the Internet is dominated by the two major trends: P2P data exchange and access to video content. The combination of these trends enforces exploration of new technologies for P2P video transport over IP ensuring some level of Quality of Experience (QoE). These technologies require joint design and optimization of video coding and networking aspects; and applications such as Video on Demand (VoD) and real-time live WebTV services may have different video coding requirements.

In video coding, there are different choices which trade compression efficiency with adaptation to network rate variations and peer stability. For example, H.264/MPEG-4 AVC provides the best compression efficiency but does not provide adaptation, except for stream switching that requires high storage capacity. Scalable video coding (SVC) [3] has been added as an annex to the AVC standard to provide a base and one or more spatial, temporal and quality enhancement layers for effective adaptation of video rate to the available network rate in exchange to slight reduction in compression efficiency. Multiple description coding (MDC) [4], on the other hand, generates independently decodable streams, with more redundancy introduced, that are transported over independent links, e.g., wireless links, to provide robustness against packet losses and/or link failures. In SVC, how to allocate rate to base and enhancement layers and in MDC, how to design the MDC scheme, and the allocate rate and level of redundancy between descriptions are important design parameters that should be optimized according to the P2P protocol, network topology, and the type of provided service.

In P2P networking, there are different overlay networks such as tree-based and mesh-based. Tree-based solutions can provide an efficient transport mechanism to deliver content from single originator at the top of the tree to peers that are connected to each other in a parent-child fashion. Data flow starts immediately once a peer joins the network allowing lower latency in data dissemination compared to mesh-based solutions in which peers request data and some amount of pre-buffering delay is needed for timely constrained applications. Therefore, tree-based solutions fulfill the requirements of time critical applications such as IPTV. However, ensuring high service quality using tree-based P2P solutions is not trivial and major challenges and problems exist. The complexities involved with the maintenance of the tree structure and lack of sufficient upload capacity to feed multiple peers due to asymmetric Internet connections are main drawbacks of tree-based solutions. In case of an ungraceful peer exit, the child peers receiving from the exited peer and their descendants suffer since data dissemination stops at that level of the hierarchy. Similar problem occurs if peers' upload capacities are not sufficient for sending data to multiple child peers in time. In literature, highly cited solutions dealing with these challenges are published. [5] and [6] propose that each child has a backup parent along with the actual parents, while [7]-[10] employ multiple-tree formation. However, commercial P2P deployments that rely purely on tree-based solutions do not exist due to practical difficulties of the proposed approaches. For instance, hierarchical ordering of the peers may not be always straightforward. It may be a challenge to identify the cause of the bottleneck, which is due to either upload insufficient parent or download constrained child, and to locate peers accordingly within the hierarchy. On the other hand, in mesh-based solutions, data is distributed over an unstructured network, in which each peer can connect to multiple peers with two way connections. By having multiple connections simultaneously, the possibility of receiving data on time increases and peers can deal with ungraceful peer exits accordingly. Mesh overlays are based on the self-organization of the peers and building multiple connections or search for a new peer to connect requires a certain amount of time. For this reason, mesh-based solutions for video streaming applications, e.g., VoD, require some initiation delay. Undoubtedly, BitTorrent [11] is the most widely used application for distributing large data files over mesh-based P2P networks. Inspired by its success, many research groups have proposed solutions that enable timely video delivery by modifying the chunk scheduling algorithm of BitTorrent [12][13][43].

As an alternative to classification of media streaming systems into server-client and P2P, they can also be categorized from different perspective as push-based and pull-based systems. The main motivation of this type of classification is to identify who drives data into the network. In push-based systems, sender directly sends data from its buffer (sender-driven), while in pull-based systems receiver requests the stored data according to its needs (receiver-driven). Considering adaptive streaming mechanisms, in sender-driven systems, senders intelligently decide on the level of adaptation and accordingly they may drop enhancement packets from its buffer. Although fine-grain adaptation is achievable in sender-driven traditional server-client systems, some problems may occur for multiple-tree based networks employing sender-driven solutions. For instance, a child peer may receive the same data pushed from different parents located at different trees and consequently, the link utilization reduces due to redundant data flow. For such cases, MDC technique becomes a promising solution decreasing the level of redundancy to some degree [14][15][16], although it is not adopted by the industry. Push-based systems should not be matched with tree-based solution, and in literature mesh-push based streaming solutions exist [17][18][19]. On the other hand in pull-based systems, in which the decision mechanism is the receiver itself, peers can intelligently request data according to their needs. For example, they can choose what to request and whom to request from, so that they can perform adaptation and maximize throughput. However, the precision of adaptation cannot be as high as in push-based systems. In order to obtain fine-grain adaptation capability, peers need to request for each one of the small data packets, e.g., NAL units in H.264 based encoding. Nevertheless, such approach is not feasible with very small transmission units, since the messaging overhead and the delays dramatically increase.

1.2 Contributions and Organization

Most results reported in the literature regarding adaptive P2P streaming either have unrealistic P2P protocols that are not utilized in industry such as centralized tree-based solutions or they employ very basic encoding approaches with very limited results from both network utilization and end-user video quality perspectives. We believe that a successful P2P streaming solution should jointly consider video coding, chunk generation and P2P networking aspects at the same time. In this thesis, we mainly propose a new BitTorrent like adaptive P2P streaming system with buffer driven adaptation approach. The main contributions of this thesis can be summarized in the following three categories:

- We propose a robust chunk prioritization and selection method based on buffer condition that ensures continuity of play-out, maximizes bandwidth utilization, minimizes bandwidth wasting and finally reduces server load by maximizing P2P activity.
- Employing the proposed chunk scheduling approach in our BitTorrent-like P2P streaming protocol, we first compare RD performances of two layered SVC and multiple layered SVC; and in addition we evaluate the streaming performances of these scalable coding approaches for different test scenarios covering different bandwidth adaptation conditions. The results are presented from both overall network and also individual peer's perspective.
- We comprehensively evaluate performance of video streaming using SVC and MDC under different conditions, such as varying congestion levels, ungraceful peer exits and different Round-Trip-Times (RTT). Moreover, we provide answers to key questions, such as if and when SVC and MDC encoding methods should be preferred over BitTorrent-like P2P networks.

The rest of the thesis is organized as follows: Chapter 2 covers video coding solutions for P2P video streaming. Chapter 3 discusses the BitTorrent-like streaming protocol that we implement, the design goals of a proper chunk scheduling method, and describes our approach for robust chunk scheduling meeting the design goals. Chapter 4 extensively evaluates video coding solutions for adaptive P2P video streaming. First, the effect of increasing number of layers in SVC is analyzed. Second, SVC vs. MDC usage is compared for adaptive video streaming over both sender-driven and receiver-driven P2P networks under different combinations of network dynamics. Finally in Chapter 5, we draw our conclusions.

Chapter 2

VIDEO CODING FOR P2P STREAMING

Advancements in video coding and networking technologies in the recent past have enabled transmission of digital video over heterogeneous networks and platforms. However, video applications, e.g., Video on Demand, IPTV, over various networks such as mobile, server-client, P2P and the corresponding clients have different demands on video quality and different capabilities for receiving and decoding of a video. A conventional video coding system encodes a video sequence in a fixed bitstream that is adequate for a given application. Therefore, serving different clients requires transcoding of a given video sequence, which is not computationally efficient and may reduce video quality. Furthermore, particular applications like P2P can even change demands on video bitrate during a single video transmission session due to varying bandwidth conditions of the peers or other network dynamics such as packet delays or losses.

Multiple alternatives exist for video coding, which offer different trade-offs between compression efficiency, granularity of adaptation to rate variations, and robustness to packet/loss delay. For dealing with congestion, the most common solution deployed in existing video streaming platforms is multiple bitrate coding (MBR) using H.264/AVC, or version coding. In MBR coding, the video is encoded at different bitrates generating self-decodable bitstreams at different qualities. An adaptation method using MBR encoding is stream switching, where the client may increase or decrease the quality of video during streaming by requesting pieces of higher or lower quality streams to match the video rate to available network rate [37]. MBR coding provides better encoding efficiency compared to scalable video coding, since scalable video coding has an overhead of scalability. However, it requires larger storage space at the sender side to store video encoded multiple times. However, as this traditional approach does not provide at the same time a low-cost adaptation and requires larger storage space at the sender side to store video encoded multiple times, a need for a new technology is evident.

Scalable video coding provides a straightforward solution for a universal system for video coding that can serve a broad range of applications. Its layered syntax can be utilized to serve for both terminal adaptation and rate adaptation. Moreover, in order to overcome packet losses or delays, multiple description coding can provide robustness.

2.1 Scalable Video Coding

Scalable video coding (SVC) is an extension of the H.264/AVC standard, which has been standardized by the Joint Video Team (JVT) of ITU-T VCEG and ISO/IEC MPEG [38]. SVC provides a base and one or more spatial, temporal and/or quality enhancement layers for effective adaptation of video rate in return to some reduction in compression efficiency. For backward compatibility, the base layer of SVC is compliant with the H.264/AVC syntax. When compared with the H.264/AVC, SVC provides better Quality of Experience (QoE) in case of limited resources, such as display resolution or link capacity.

2.1.1 Scalability Types

SVC enables two types of adaptation: terminal adaptation and rate adaptation. The former aims to adapt the video to the display resolution or the processing power of the user terminal while the latter aims to adapt the video rate dynamically to the available bit rate of the network. In order to qualify these adaptations, SVC provides three types of scalability: spatial, temporal and quality scalability. Temporal scalability is achieved by the usage of hierarchical prediction structures, whereas spatial and quality scalabilities are supported by multilayer coding [38].

2.1.1.1 Temporal Scalability

Temporal scalability associates grouping the video layers temporally into hierarchical classes. Each class predicts from the previous one. Figure 2-1 illustrates hierarchical prediction structure of a video coded with B-frames. T stands for temporal layer identifier that is T0 for base layers and increases by 1 for each of the enhancement layers in temporal domain. Given a layer identifier number k, by removing all frames with temporal layer identifier numbers greater than k, another decodable bit stream can be formed [38].

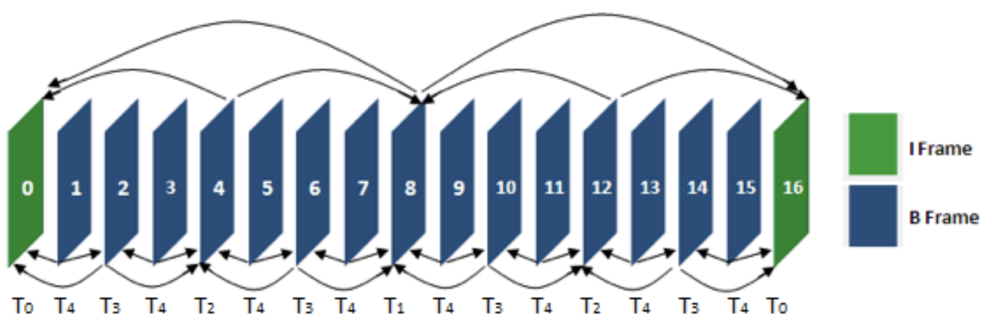

Figure 2-1 Hierarchical prediction structure with B-frames of GOP size 16

The target of the temporal scalability is mainly the adaptation to different terminal conditions. Different types of end-user devices have varying processing powers and display refresh rates. A device may require higher temporal resolutions. In order to provide compatibility with the ones working in lower temporal resolutions, the content needs to be encoded as temporally scalable. Compared to spatial scalability and quality scalability, temporal scalability has limited adaptation capability since dropping temporal enhancement layers yields marginal reduction in bitrate.

2.1.1.2 Spatial Scalability

Spatial scalability constitutes layers with different spatial resolutions. Using an oversampled pyramid approach the original high-resolution image is filtered and decimated [39]. Lower layers provide spatially decimated resolutions, and after spatial interpolation, they can be used in inter-layer prediction of higher layers with higher resolutions so that the bitrate can be reduced. The lowest resolution image is independently decodable and backward compatible with H.264/AVC. In each spatial layer, motion-compensated prediction and intra-prediction are employed. Its advantage over simulcast coding of each spatial layers is the inter-layer prediction usage as indicated by arrows in Figure 2-2. SVC also allows mixed scalabilities. Figure 2-2 depicts the combination of temporal scalability and spatial scalability.

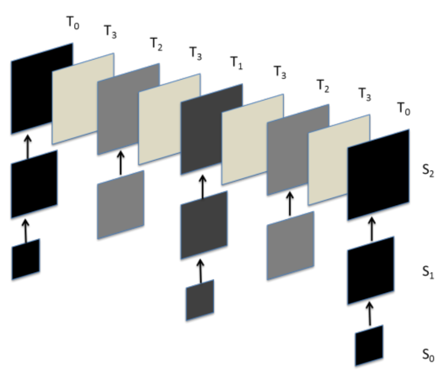

Figure 2-2 Combination of temporal and spatial scalability

2.1.1.3 SNR Scalability

Quality scalability, Signal-to-Noise Ratio (SNR) scalability, can be considered as a special case of spatial scalability, where in each quality layer the spatial resolutions of the images are identical. Quality scalability is supported in two modes: Coarse-grained scalability (CGS) and medium-grained scalability (MGS). In CGS, the same inter-layer prediction mechanism with the spatial scalability is utilized without using the spatial interpolation

[38]. Thus, rate adaptation can be performed only on full layer basis, whereas MGS allows higher adaptation granularity by providing packet-based scalability [40]. Also it is possible to fragment an MGS layer into multiple sub-layers by grouping transform coefficients in zigzag order and also to increase the number of rate adaptation possibilities.

2.1.2 **Rate-Distortion Analysis**

Scalable video coding evolved from two main branches of conventional video coding: integer cosine transform and 3D wavelet transform. The standard scalable video codec H.264/SVC employs integer cosine transform. In this section, the rate-distortion (RD) performance comparisons of H.264/SVC and an alternative wavelet-based scalable codec [41][42] are performed. Moreover, the effect of increasing number of enhancement layers in standard SVC on the encoding efficiency is also analyzed.

To evaluate RD performances, we consider three different contents encoded with group-of-picture (GOP) size 16: *Foreman* sequence with CIF resolution at 30fps, *Soccer* sequence with 4CIF resolution at 30fps, and *InToTree* sequence with 720p resolution at 50fps. Among these contents, *Soccer* is the highest motion featuring one, *In to* Tree has the highest spatial variation, and *Foreman* locates between the other two contents for both its spatial and temporal characteristics. Figure 2-3 presents comparison results for all contents. The legends with M-SVC stand for standard Mpeg H.264/SVC, while W-SVC indicates wavelet-based SVC. Standard SVC performs better at low bitrates for all choices of number of quality layers. However, as the number of layers increases, the overhead of scalability also increases in the standard SVC. On the other hand, in wavelet-based SVC, there is not any additional overhead for encoding with higher number of layers. Thus, increasing the number of quality layers in the standard SVC, the RD curve gets closer to curve for wavelet-based SVC. Also at high bitrate regions, it lies under the wavelet-based SVC curve. For the RD evaluations of the two codecs, we conclude at two important points.

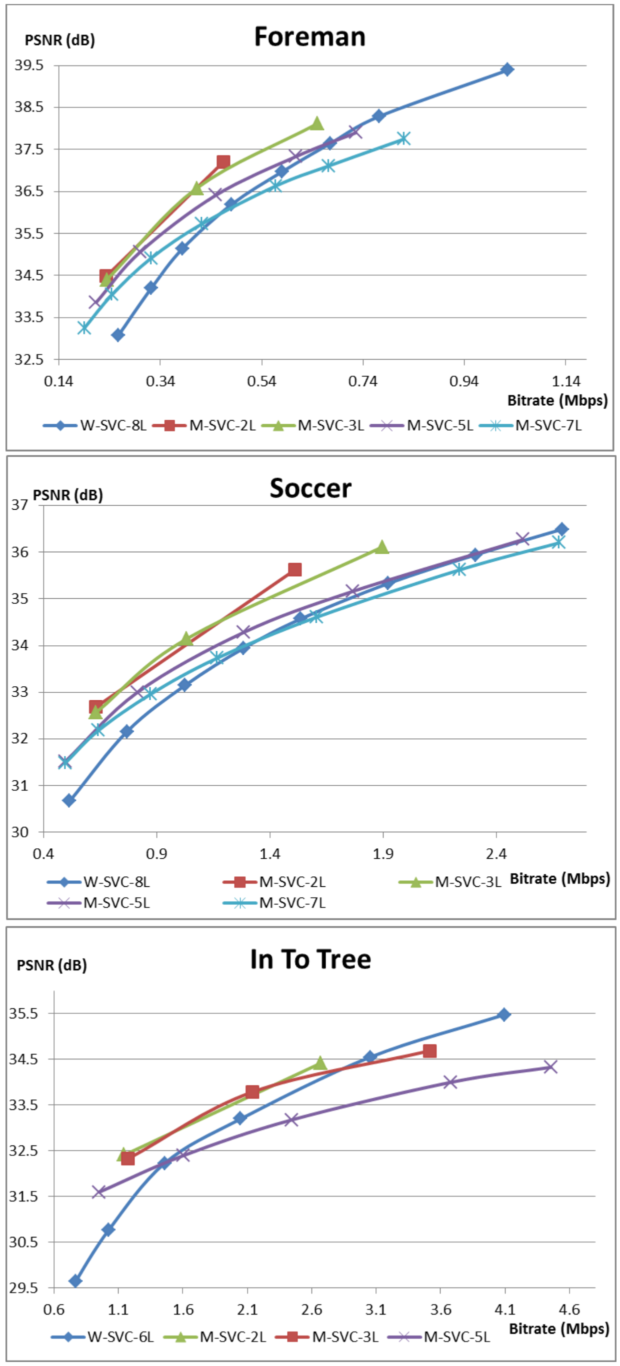

Figure 2-3 Rate-distortion comparisons of H.264/SVC and Wavelet-based SVC

First, when two layers standard SVC and eight layers wavelet-based SVC are compared, a significant point that needs to be elaborated emerges. The same quality can be achieved with the standard codec for less bitrate, especially for the *Soccer* sequence. However, there is trade-off on the level of scalability since the adaptation to varying bandwidth conditions may become coarser compared to wavelet-based SVC. Consequently, this may affect the QoE perceived by the end users due to higher jumps between layers. Second, with higher number of layers, the standard SVC performs better at low bitrates as compared to wavelet-based codec; at medium bitrates the performance of wavelet-based codec gets closer to the standard SVC; and at very high bitrates wavelet-based SVC performs better with similar capability of scalability. Similar conclusions apply for each test sequence.

2.2 Multiple Description Coding

2.2.1 Overview

Multiple description coding [4] is a promising solution for delay intolerant applications, e.g., video streaming, addressing the problem of video delivery over unreliable networks where bursty packet losses occur. In MDC, video is encoded into two or more independently decodable bitstreams, called descriptions. Descriptions can be balanced or unbalanced in terms of bitrate and contain complementary data to each other in addition to the redundant data. Thus, employing MDC, each received description augments the video quality by utilizing the complimentary data. On the other hand, it provides immunity and robustness against events such as packet losses/delays due to inherent redundancy.

In MDC, although each received description enhances the quality of multimedia content, there is a trade-off between the level of redundancy among descriptions and error resilience. In order to fully utilize MDC, each description should be transmitted over independent links, e.g., different paths of P2P overlay or wireless links, to provide robustness against packet losses and/or link failures. In the case of dependent links, the

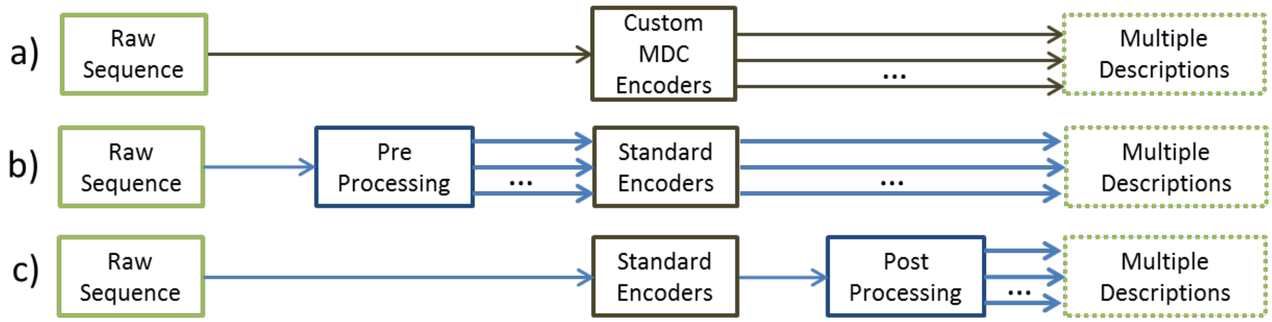

Figure 2-4 High Level Classification of Techniques for Designing MDC Coder

benefit of using MDC diminishes since an error in one of the links (i.e., packet loss due to congestion) affects other links causing data loss in each description.

Various techniques exist for designing a multiple description coder, and they can be basically classified into two: 1) Custom designed MD coders and 2) Codec agnostic techniques with pre/post processing. In literature, there are different approaches for MD generation using custom coders and some examples for this approach are data partitioning [4], e.g., DCT coefficient splitting, multiple description quantization [45], and multiple description transform coding [46]. In [47] different combinations of video segments, code blocks, coded at high and low rates have been employed for wavelet-based flexible MD encoders. Moreover, motion compensated temporal prediction based MD coding technique is proposed in [53]. More optimized schemes in terms of introduced redundancy and robustness to packet losses can be achieved by custom designed MD coders that are specifically designed according to channel characteristics and utilized transmission mechanism, while standard based approaches promise compatibility. The techniques based on standard codecs, e.g., H.264/AVC, H.264/SVC, etc., require either pre-processing of the raw sequence or post-processing of the encoded bitstream as depicted in Figure 2-4-b-c. Temporal splitting of raw frames in [48][51] and generating descriptions by temporal splitting and periodic redundant picture insertion in [50] are examples of standard codec based MD generation. Forward-error-correction approach is adopted for generation of MD bitstreams in [52] and [54]. In addition, [49] follows the approach in Figure 2-4-c by employing flexible MD encoders compatible with H.264/AVC/SVC where post processing is performed for distributing enhancement layers between descriptions that generates GOPs at high and low bitrates.

2.2.2 Determination of MDC Scheme for P2P Streaming

H.264/AVC would be the best coding choice due to its superior compression efficiency, if the video service was guaranteed to avoid congestion and packet losses. However, most video services today require scalability of the bandwidth and a mechanism for robustness against packet losses/delays. Video coding solutions, SVC and MDC, offer different trade-offs between compression efficiency, adaptation to bandwidth variations, and robustness to packet losses. As discussed in above, SVC is a well-defined solution for adaptive video delivery over congested networks, while MDC is basically designed in order to cope with packet losses/delays that can be caused due to ungraceful peer exits in P2P networks. Figure 2-5 depicts a type of categorization of video codecs based on their proposed usage under different combinations of network dynamics, which are congestion and peer exits. In our evaluations of SVC and MDC over P2P networks, we consider the conditions where both congestion and ungraceful peer exits occur. Therefore, the selection of the video coding should take both adaptation to congestion and robustness to peer exits in to account at the same time. To this end, this approach corresponds to the third quadrant indicating use of SVC-based MDC scheme. In literature, there are various studies on the generation of SVC-based MDC schemes. Taal et al. [70] and Chou et al. [71] form scalable descriptions based on FEC in which each layer of a description is protected by corresponding erasure codes. Adedoyin et al. [72] proposes scalable MDC architecture that creates an even and odd streams with residual information regarding the other stream, so that the residual data can improve the quality of interpolated frames. Abanoz et al. [49] proposes distribution of enhancement layers between MDC descriptions, while Kondi [73] achieves hybrid scalable/multiple description codec by partitioning the DCT coefficient for SNR scalability. Further studies combining SVC and MDC could be found in [47][74][75][76].

Since the transmission mechanism is chunk based in receiver-driven P2P networks as we will elaborate more in section 2.4, adaptation to bandwidth conditions or recovering a lost/delayed packet can only be performed in chunk basis; and fine-grained recovery is not likely. For example, redundant picture based MDC [50] has capability of compensating loss of single P-frame, while [77] can even go further by allocating bitrates of the descriptions in macro-block scale. However, employing such approaches in pull-based P2P networks does not offer capability of recovery in frame or macro-block level. Therefore, the employed MDC scheme does not have to be fine-grained and complex.

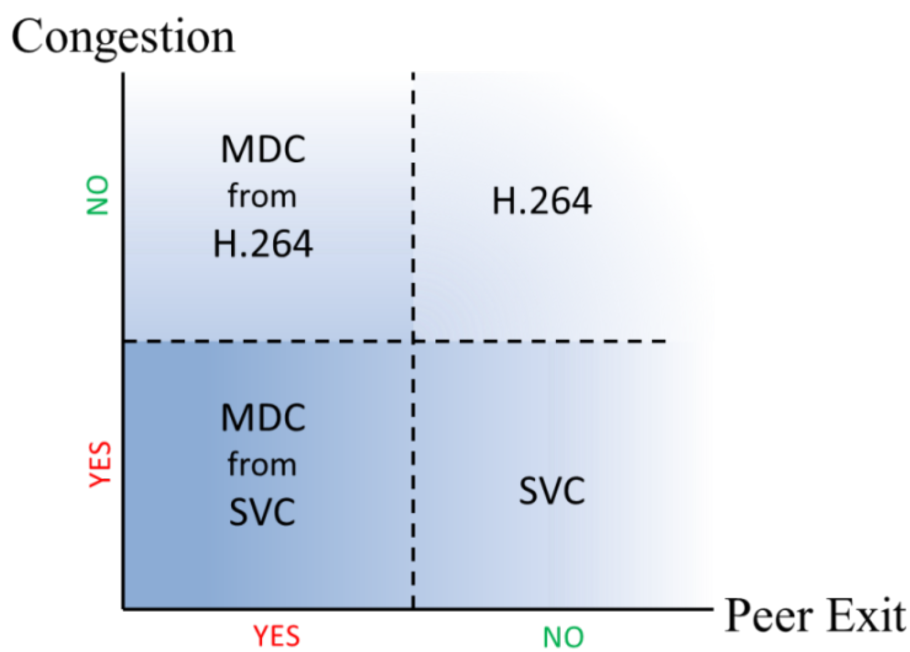

Figure 2-5 Selection of Video Coding

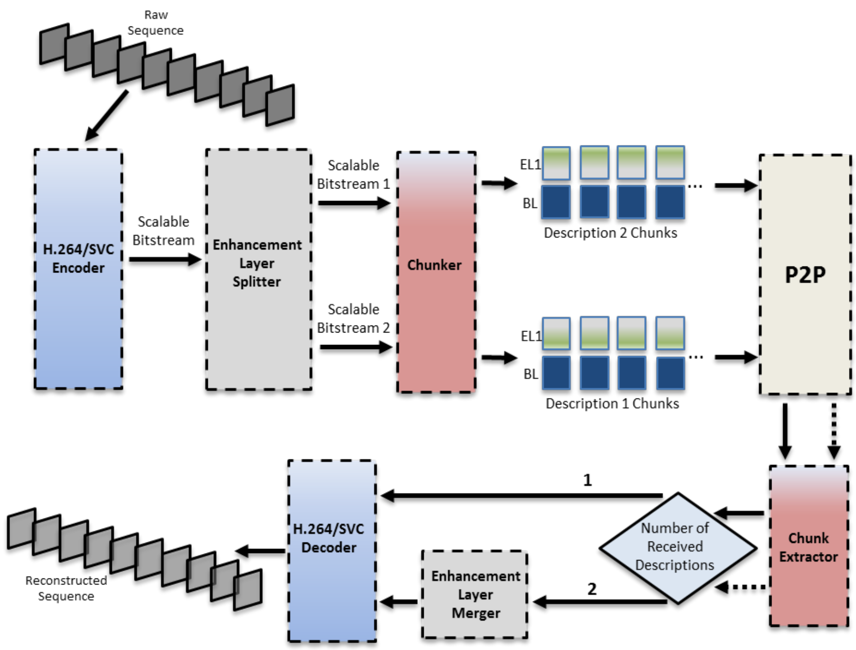

Figure 2-6 Employed SVC-based MDC Scheme 1

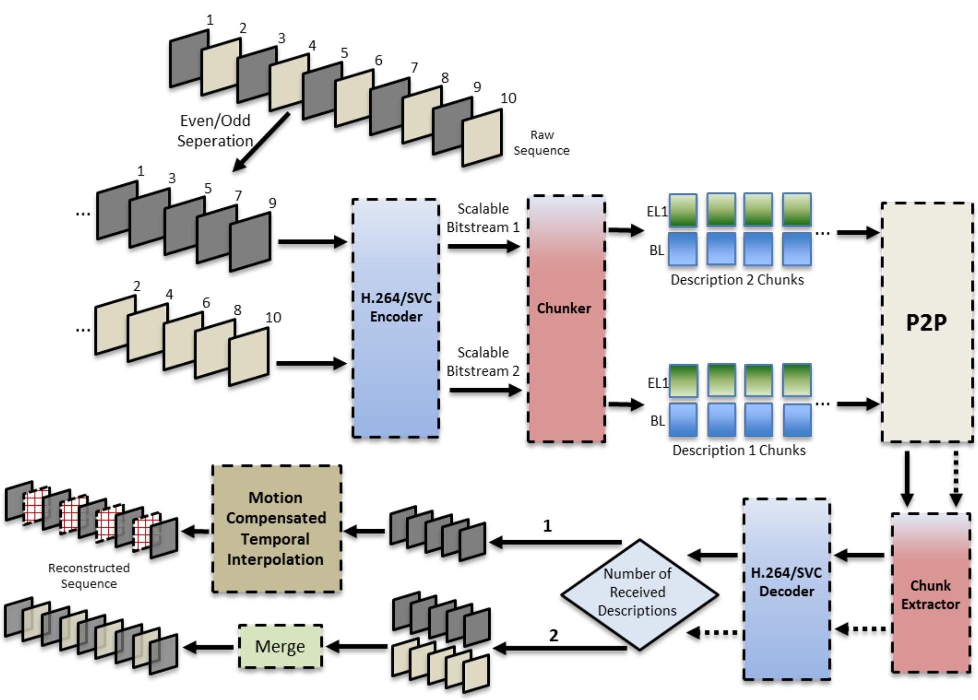

Figure 2-7 Employed SVC-based MDC Scheme 2

Taking the advantage of both SVC and MDC, and having the intuition on chunk based MDC generation; we generate SVC-based MDC chunks as described in Figure 2-6 by adopting the MDC scheme in [49]. In addition, with lower redundancy, we also employ even/odd frame splitting based approach as depicted in Figure 2-7 similar to [72] without side information usage, which consequently reduces the reconstructed quality when only one of the descriptions is received and decreases redundancy. For motion compensated temporal interpolation, we utilize the software [78] that uses an MPEG-style motion compensation algorithm.

2.3 Related Work

2.3.1 MDC over P2P Networks

P2P networks are promising mediums to utilize MDC for video streaming applications since each description can be sent and requested over different/independent paths or links. In literature, there exist quite efforts on deploying MDC over P2P networks. Tree-based approaches can easily exploit use of MDC in order to distribute live content with push-based mechanism. CoopNet [15], Splitstream [7] and [61] use multiple diverse distribution trees to provide redundancy in network and MDC to provide redundancy in data. Pouwelse et al. [62] extends MDC with scalability feature and uses over multicast tree structure. In addition, TURINstream [65] employs multi-tree-based push solution where tree nodes are represented by cluster of peers, which is formed by a small set of fully connected collaborating peers. Unlike in tree-based solutions, the usage of MDC in mesh-pull based P2P networks is not straightforward. Liu et al. [66] adopts MDC for building an incentive mechanism based on service differentiation in which peers contributing with more uplink bandwidths receive more description and consequently better video quality. Magharei et al. [67] analyzes the fundamental tradeoffs and limitations in design of a mesh-based P2P streaming for live content using MDC. In addition, Lu et al. [68] proposes partnership formation mechanism over mesh-based overlay where spatial-temporal hybrid interpolation based MDC scheme is employed for adjusting streaming rate to bandwidth and device capacities of each peer. Xu et al. in [63], a VoD system by using MDC and streaming different descriptions of a requested video from separate peers is described for pull-based P2P approach in which they show that increasing number of descriptions can improve the system performance significantly due to greater path diversity and flexibility in redundancy allocation by FEC-based MDC. Moreover, Zink et al. [64] discusses the feasibility of MDC for P2P video streaming. Both studies in [63] and [64] assume that some peers are dedicated for streaming of the specific description(s) without considering the type of the overlay whether it is tree-based or mesh-based. However, such assumption may not be realistic for Torrent-like streaming protocols. Summing up these studies, we might conclude that MDC mostly employed over tree-based overlays, while studies over mesh-pull-based P2P networks is inconclusive and require more comprehensive analysis.

2.3.2 **Layered Coding vs. Multiple Description Coding**

Layered coding (LC) (H.264/SVC in this work) and MDC have similar characteristics as they both generate multiple sub-bitstreams, which are exploited in order to adapt the network conditions or be resistant to packet losses. However, the main difference between the two coding approaches is that SVC requires base layers for decoding enhancement layers while descriptions of MDC are independently decodable. Therefore unlike SVC, MDC does not require any prioritization for the transmission of packets containing base or enhancement layers. In literature, various studies on the comparison of the two coding approaches for transmission have been performed. However, there is not an ultimate result for the performance comparison of both codecs and there are contradictory conclusions about the superiority of one codec over the other.

Reibman et al. compares FEC-coded LC with balanced MDC over a binary symmetric and a random erasure channel [55]. They concluded that MDC performs better than LC only in very high error prone networks. Moreover, Singh et al. concluded that MDC outperforms LC in most of the scenarios over networks with long RTTs in which base layers are transmitted with TCP [56]. In [57] comparisons are performed over wireless networks with multipath routing in which they also considered LC with Automatic Repeat Request (ARQ) for base layers. Their observations were similar to above studies where MDC is better for delay constraint applications with long RTTs. However, they also concluded that if retransmission of the base-layer with ARQ mechanism is allowed, LC outperforms MDC. In addition, Zhou et al. showed that MDC performs better for large packet loss rates and delays, while LC is preferred for small loss rates and acceptable delays [58]. On the other hand, Chakareski et al. concludes layered coding outperforms MDC when rate-distortion optimized scheduling of the packet transmission is employed [59]. Lee et al. in [60] performs the most comprehensive evaluations considering FEC and ARQ based protection methods and they observe similar results. However, in [56] it is remarked that their results contradicts with the ones in [55]. Moreover, a very recent work by Xu et al. [69] proposes a rate-distortion optimized packet scheduling scheme for MDC, and evaluates MDC vs. SVC over mesh-pull-based P2P networks. Interestingly they end up with that MDC exhibits stronger robustness and achieves better performance even without peer churn. Due to all these varying results, it is not trivial to conclude on the performance comparisons of layered coding and multiple description coding. The possible reason for these different conclusions and contradictions is that the experiments may not be comprehensive and performed under different assumptions and conditions. In any case, it can be summarized as the general view of the results as MDC has advantages over LC for networks with no feedback mechanism and a long RTT while LC is preferred if prioritized transmission with error control is employed.

2.4 Chunk Generation for P2P Streaming

How to packetize the elementary bitstream of multimedia content has critical importance, since packetization requires the consideration of the network structure and properties of the employed streaming protocol. In P2P delivery, the main unit of data exchange is called as chunks. In BitTorrent, the most widely used P2P protocol for file distribution, chunks are fixed sized, which is determined according to the total size of the shared content. In NextShare P2P video streaming platform, which is developed over BitTorrent, same fixed size chunking approach is adopted [30][43]. However, this approach is not feasible for video delivery since video may probably have varying bitrate and fixed size chunking may require padding of redundant data. Therefore, we utilize variable sized chunking approach in which each chunk contains a GOP. The effect of using variable size chunking is analyzed in [28]. By doing so, each chunk contains independently usable data. If the payload size is low for utilizing the available bandwidth due to TCP's slow start policy, multiple numbers of GOPs can also be packed in a chunk. However, this may reduce the adaptation capability since adaptation is performed on chunk basis.

In conventional client-server adaptive video streaming solutions using SVC, adaptation is performed by dropping the enhancement layer NAL units from senders' buffer in sender driven approaches. While in receiver driven solutions, it is done by not requesting the enhancement layer NAL units. Therefore, we separate chunks as base chunk and enhancement chunks. Base chunks contain base layer and Non-VCL NAL units in order to ensure video decoding, while enhancement chunks contain only enhancement layer NAL units as depicted in Figure 2-8 for the bitstream with base layer and one enhancement layer where GOP size is set as 4. In this study we consider only SNR scalability, and therefore enhancement NAL units contain data needed for higher quality. Since our packetization is GOP based and fine granularity of packetization is not needed, slice mode is disabled while encoding the content with H.264/SVC. Besides, it brings additional overhead. Note that first base and enhancement chunks contain NAL units for two I-frames due to hierarchical coding in H.264/SVC. In our simulations, the header NAL units are also packed into first base chunk.

Chunk generation for SVC-based MDC is similar to SVC chunking as depicted in Figure 2-9. As discussed in section 2.2.2, we consider scalable MDC schemes where each description is independently decodable and has a layered architecture.

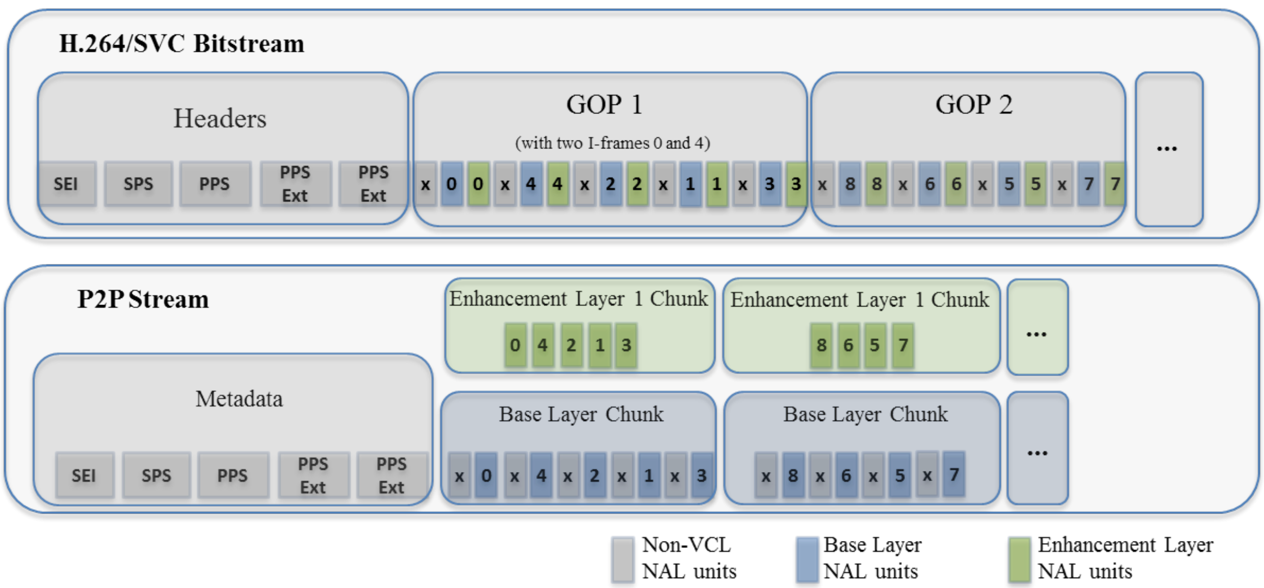

Figure 2-8 Sample Chunk Generation Using one GOP per Chunk for 2-Layer SVC

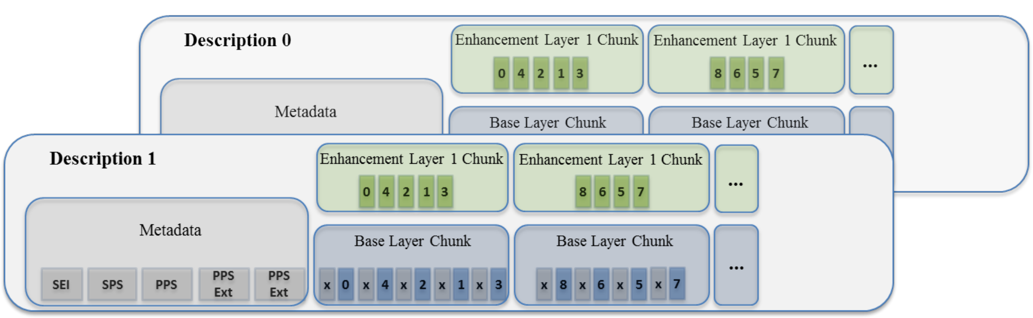

Figure 2-9 Chunk Structure of an SVC-based MDC

Chapter 3

ADAPTIVE P2P VIDEO STREAMING PROTOCOL

This chapter describes the features of the BitTorrent-like protocol for adaptive video streaming over P2P networks. We developed the protocol by using Peersim [1], which enables great scalability by offering extendable and pluggable components of the network. The streaming protocol is based on an event-based engine, since it allows us to manage and observe the events in packet level and in a timely fashion. Following, we discuss the main features of the pull-based P2P video streaming solution that we implemented.

The streaming protocol employs mainly two groups of messages: 1) overlay related messages and 2) data (e.g. chunk, bitmap) related messages. Overlay related messages include connection request-accept-refuse and graceful exit messages. The latter group contains chunk get-have-send-ack and busy messages. Our video streaming protocol is a mesh-pull based solution. Unlike push-based systems, in pull-based solutions the intelligence is at the receiving peer side. In other words, peers decide which chunk to request, when to request and from which source to request. The streaming protocol must ensure continuous play-out and provide as much quality as possible by efficiently utilizing the network. Therefore, it should be robust to network dynamics such as peer exit and congestion. Peer exits occur in two ways: they may either leave the network with notification by sending graceful exit message or they simply quit ungracefully without any notification. The effects of these two types of quits are different. By informing the neighbors for the exit event, the exiting peer makes its neighbors save time by avoiding possible chunk requests to itself. However, instantaneous exits without notification may cause discontinuous play-out, since peers lose time until they notice the exit event and make a new request to other neighbors for the same chunk. On the other, for the cases where congestion occurs, the receiver should adapt its future requests for layered video chunks. Consequently, in order to efficiently utilize the network and obtain continuous play-out with highest quality possible, three important features needs to be implemented within the adaptive video streaming protocol: 1) peer selection, 2) sliding windowing and 3) chunk scheduling.

3.1 Peer Selection

In pull-based solutions for video streaming, the choice of which neighbor peer to request a chunk has an effect on the overall performance of the system. The fastest way to get a chunk can be to request it directly from the main seed server; however it is contradictory with the concept of P2P. On the other hand, a peer can end up with discontinuous viewing experience if chunk is requested from a neighbor having insufficient upload capacity. Therefore, a robust P2P protocol requires experiencing as much quality as possible while minimizing the load at the main seed server.

In this work, we do not focus on efficient peer selection algorithm for enhancing the performance of the system from the peers' perspective. Instead, from the perspective of P2P activity, in order to diminish the server load peers request chunks firstly from its leecher neighbors; and if all requested leecher neighbors responds with busy messages, chunks are requested from seeder neighbors. If peers are still unable to find an available peer, they eventually request from the main seed server.

In literature, solutions for optimum peer selection have been extensively studied and there are highly cited solutions. Further information could be found in [22][23][24][25][26][27].

3.2 Dynamic Sliding Windowing Mechanism

Due to timely delivery requirement of the P2P video streaming applications, sliding windowing mechanism is implemented so that chunk requests are performed within the sliding window. Unlike P2P file distribution applications with policies such as rarest-first, windowing mechanism enables peers to request chunks with closer play-out deadlines.

The window slides from its current position t_{WIN} to next position t'_{WIN} for two different cases: 1) when all chunks within the window have been just received, and 2) when the current play-out time reaches the window's current position. The first case indicates that peer is able to get chunks of the current window before their play-out deadlines. Once all chunks of the current window position are received, window slides by its size s_{WIN} (window length based on chunk quantity) and new requests for the following s_{WIN} chunks can be made as depicted in Figure 3-1. However, in the latter case, there are no buffered chunks and the play-out deadlines of the windowed chunks are just about to come up. Such situation arises if that peer is not able to get the chunks on time due to either its own insufficient downloading rate or uploading rates of the sending peers. Since this study only focuses on upload constrained P2P networks, this case may only occur due to slow uploading peers. If the window slides by its size, s_{WIN}, like in the first case, it may result in skipping of the not downloaded chunks within the window. Therefore in order to decrease the chunk skipping ratio, window slides by only one chunk as depicted in Figure 3-2. The outcome of this approach is that the peer starts getting chunks sequentially and therefore cannot contribute to P2P. It is natural that if a peer is getting the chunks barely on time, it cannot present those chunks to other peers on time and consequently decreases the P2P activity. The choice of the request windows size s_W is an important parameter for the performance of the P2P system. With very short window size, the chunk requests become sequential and consequently the availability and the diversity of chunks among peers substantially decrease.

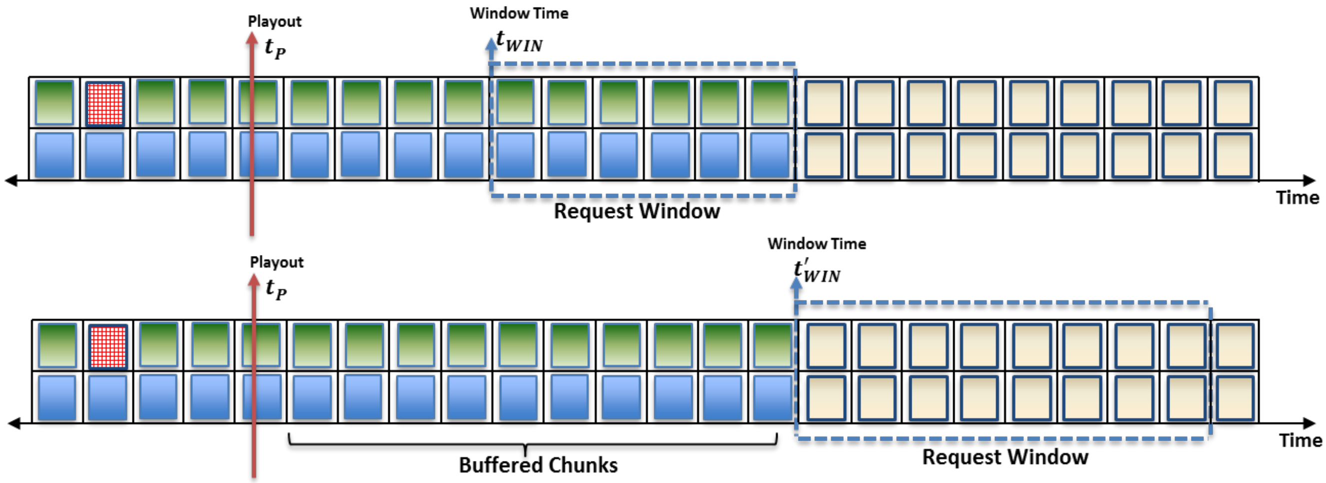

Figure 3-1 Sliding window status before and after it slides for case 1

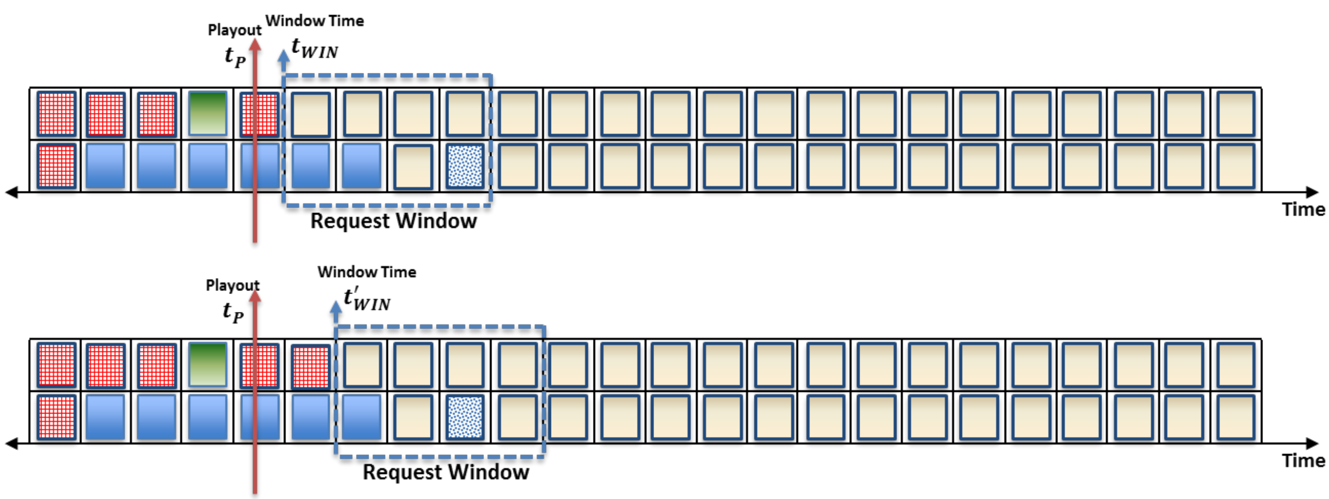

Figure 3-2 Sliding windows status before and after it slides for case 2

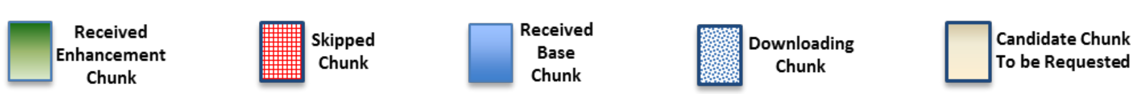

Figure 3-3 Chunk states

Such situation increases the load at the main seed server, which is not favorable for P2P networks. On the other hand, with very long request window size, the play-out may become intermittent and it is not acceptable for real time applications such as video streaming. Instead of choosing s_{WIN} as a constant value, we implement adaptive windowing mechanism, which is proposed by [28]. The approach is based on modifying s_{WIN} according to the buffer durations of the peers where buffer duration is the interval between the current window time t'_{WIN} and the current play-out time t_P. Number of buffered base chunks, which indicates the buffer duration, expresses the instantaneous play-out state of the peer. If a peer has adequate number of buffered chunks as in Figure 3-1, timely constraints on requesting chunks with closer play-out deadlines become relieved and consequently it can perform its future chunk requests in a more comfortable manner as long as it has enough buffers. Originating from this fact s_{WIN} is adaptively adjusted after each slide throughout streaming. Figure 3-1 presents an example for adaptive windowing. After window slides, s_{WIN} is increased from 6 to 8 according to equation (1) so that peer can make requests for the chunks with further play-out deadlines.

$$s_{WIN} = (t'_{WIN} - t_P) * c_{WIN} \qquad 0 < c_{WIN} < 1 \qquad (1)$$

For the other case, where peers are unable to buffer chunks s_{WIN} is not adaptively adjusted but it is set to a constant value indicated as 4 in Figure 3-2. It can be set to 1. However in order to keep P2P activity still possible, we set a higher value for s_{WIN}. Figure 3-3 presents possible states of chunks during streaming.

3.3 Chunk Scheduling

The chunk scheduling mechanism is an essential component of high performance scalable video streaming over P2P networks. By taking advantage of using scalable coding, the employed scheduling method should consider the four dimensions of high quality P2P video streaming, which are first to ensure the continuity of the play-out even for peers having very low buffer durations, second to maximize the bandwidth utilization, third to minimize the bandwidth waste, and fourth to increase the P2P activity.

3.3.1 **Design Goals**

Video streaming applications requires continuous viewing experience by the end users. In scalable video streaming over P2P networks, the intermittent viewing experience occurs when a base layer chunk is not received before its play-out deadline. Such a case may come up due to two reasons. First of all, a peer may be unable to receive the base chunk because of either its insufficient download rate or slowly uploading peer. This reasoning is related to capacities of the peers' Internet connections. Secondly, a peer may miss a base layer chunk even if the available bandwidth is higher than the bitrate required for downloading the chunk. Such a situation arises if the chunk scheduling algorithm is not designed efficiently considering all possible scenarios for the network. If an enhancement layer chunk of the closer play-out deadlines is scheduled simply before base layer chunks with later deadlines within the window, discontinuous play-out may occur since some of the available bitrate is being used for downloading the enhancement chunk and the remaining bandwidth is not enough for downloading the future base chunk on time as illustrated in Figure 3-4. Another example case resulting in discontinuous play-out due to improper chunk scheduling can be the situation where all base chunks are received within the current window and the peer requests an enhancement chunk. If the transmission time of the requested enhancement chunk takes too much time due to slow uploading peer, it is possible that the peer cannot receive the future base chunks after sliding the window and skips them. Such situations cannot be acceptable since the peer has capability of receiving base chunks on time, but due to poor scheduling approach it is not able to receive them before deadlines. Therefore, the employed chunk scheduling algorithm should intelligently consider this problem and perform scheduling accordingly.

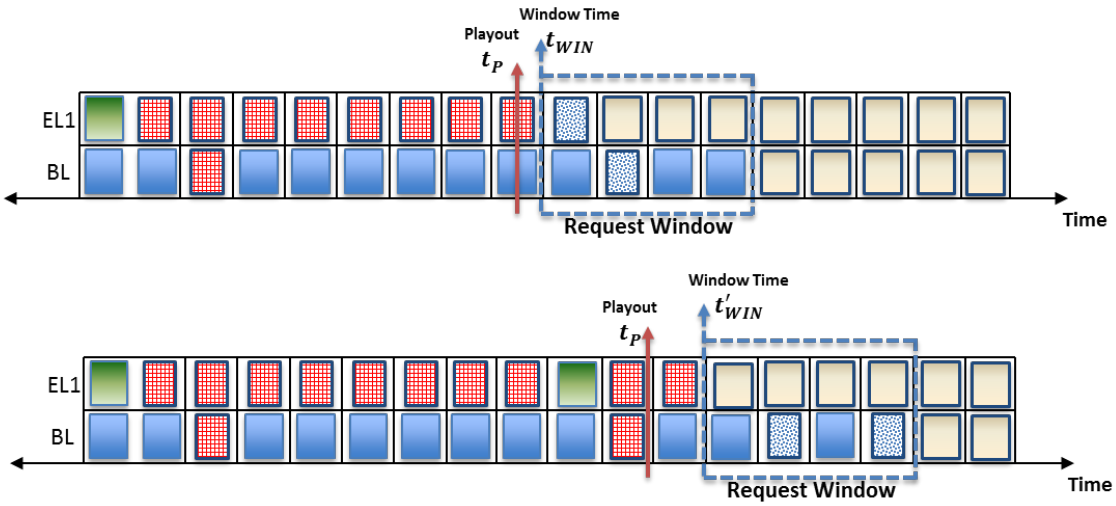

Figure 3-4 Discontinuous play-out due to inefficient scheduling

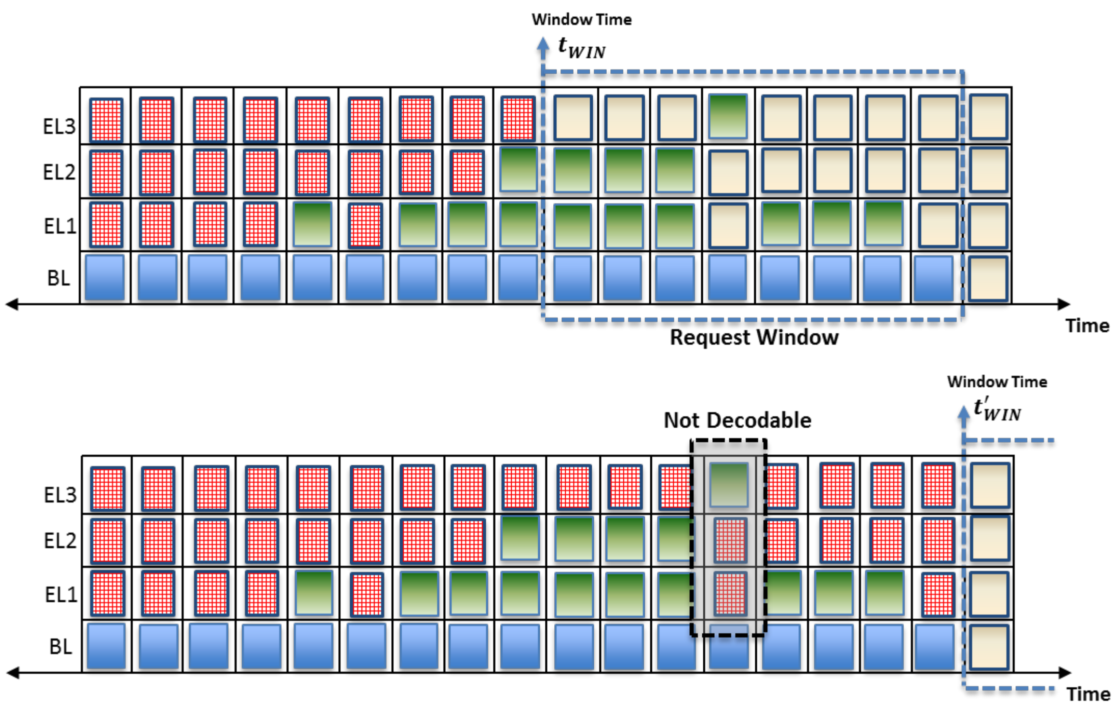

Figure 3-5 Bandwidth wasting due to not decodable enhancement layer

Efficient bandwidth utilization is another design goal of a proper chunk scheduling mechanism. The received quality should be maximized given the available bandwidth. Windowing mechanism affects the chunk scheduling, since it constraints chunk requests with the closer play-out deadlines. Once all chunks within the current window are received, it slides. Consider the case where all chunks except one due to slow uploading peer have been received and the buffer duration is high. If peer's window waits for receiving the remaining chunk and does not slide, it cannot use its download capacity to request and download future chunks. Consequently, the peer may run out of buffer. In order to deal with this problem, in our implementation a timeout value t_{OUT} is set and after t_{OUT} window slides by its size. New chunk requests within new window position can be made along with the continuing chunk download scheduled previously.

In order to avoid wasting bandwidth, chunks should be scheduled with a proper prioritization approach. Since higher enhancement layer chunks are not decodable without existence of lower layer chunks, scheduling needs to be performed in such a way that the number of enhancement chunks that are received but cannot be played is minimized. Figure 3-5 depicts a case, where only base layer and third enhancement layer chunks for the same play-out deadline have been received. Since only base layer chunk can be decodable for that time segment, receiving the third enhancement layer wastes the bandwidth instead of requesting lower layer chunks. The basic scheduling approach for dealing with this condition is to first request the lowest layer (base layer) chunks within the window, and once all chunks for the current layer are received, to continue requesting chunks from one upper layer until all chunks are received or all buffered chunks are played. This approach also minimizes the frame skipping since base layer chunks have always the highest priority but at the same time decreases the chunk diversity among peers since each peer probably requests the same chunks. This scheduling method can be called as *baseline* approach as depicted in Figure 3-6 for base layer and three enhancement layers. Another method for the chunk scheduling considering the bandwidth waste problem is the *zigzag* approach [29] as depicted in Figure 3-7. The idea is to prioritize the chunks with closer play-out deadlines within the current window by assigning highest priority to base chunks of the same play-out deadline. On the other hand, the enhancement chunks with closer play-out deadlines have also higher priority than the base chunks with later deadlines. Although this approach increases the instantaneous quality received and nullifies the bandwidth wastage problem, if the current buffer duration of a peer is very low, it may end up with missing a base layer chunk when downloading enhancement chunks of previous play-out deadlines as already mentioned in Figure 3-4.

3.3.2 **Related Work on Layered Chunk Scheduling**

Since scalable video streaming over P2P networks has been an attractive field for the researchers, several solutions have been proposed for scheduling or picking the chunks. Unlike our receiver-driven approach, work in [35] proposes a well-defined supplier side scheduling method for self-organized mesh overlays distributing layered chunks. The work in [36] is one of the first studies defining layered chunk scheduling and it suggests a receiver driven chunk scheduling method similar to *baseline* approach. In [29] the *zigzag* algorithm is proposed to achieve the best possible quality. However these approaches cannot meet all requirements of high quality P2P streaming, defined in 3.3.1 above, at the same time. To this end, partners of P2P-Next project [33] proposes Knapsack problem based piece picking algorithm for layered scheduling in [30]. In [31] they evaluate and compare their algorithm with the *baseline* and *zigzag* approaches for different scenarios considering networks dynamics such as available upload rate and peer churn rate. Although their solution works well in all scenarios compared to other approaches, the algorithm is based on a utility function using distortion reduction capability of each individual layer

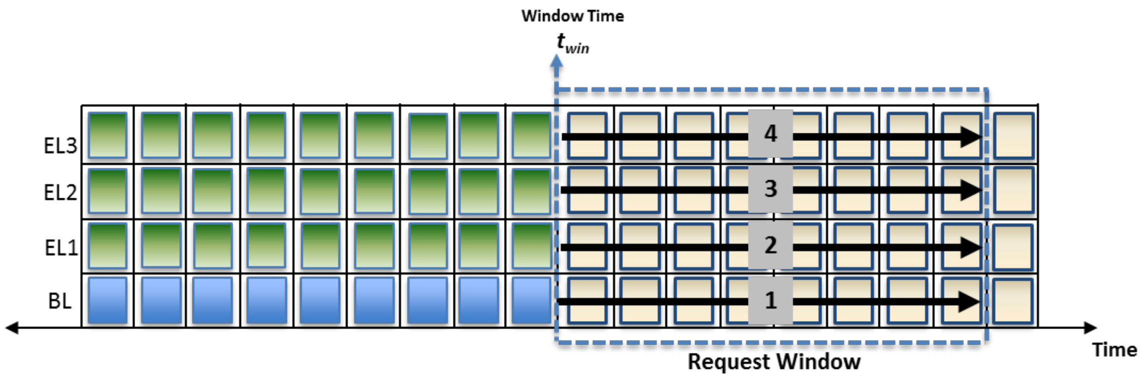

Figure 3-6 Baseline approach for chunk scheduling

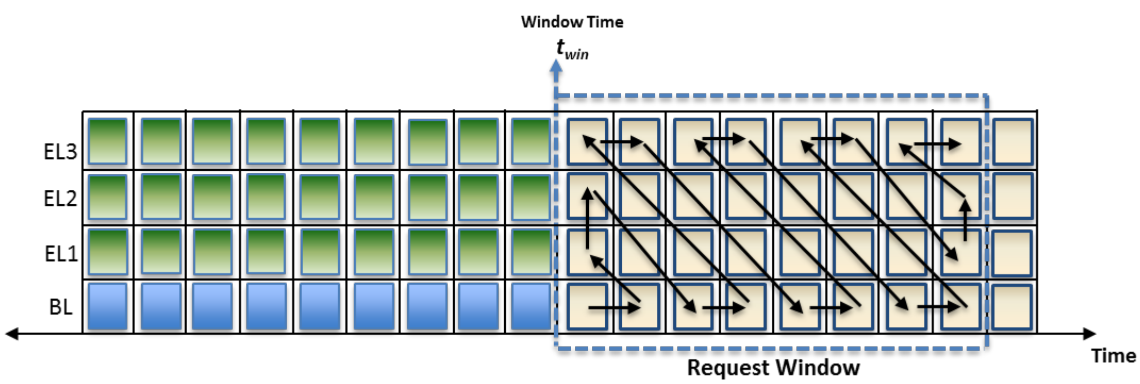

Figure 3-7 Zigzag approach for chunk scheduling

chunks and it makes their solution content dependent since each peer needs to have information about distortion reduction values of each piece before proper scheduling. In addition to their Knapsack problem based solution, they propose another scheduling method with different utility function called Deftpack [32], which performs even better than their first approach.

Even though good solutions meeting the design features of robust and efficient layered chunk scheduling exist, they do not consider or present results related to the fourth dimension properly, that is, to increase the P2P activity and decrease the load on main seed server(s) along with the peers' individual gains by ensuring the first three design goals. To this end we propose a highly dynamic and buffer driven layered chunk scheduling method meeting all design goals of the efficient chunk scheduling by ensuring continuous play-out, maximizing bandwidth utilization, minimizing bandwidth waste, and increasing the P2P activity.

3.3.3 **Proposed Method and Adaptation**

The idea behind our model is to prioritize the chunks based on mainlys_{WIN}, the size of the current request window, which is updated at every slide according to current buffer duration. Along with the current buffer duration, the number of layers N_L has also effect on how to prioritize the chunk scheduling. As mentioned in section 3.2, s_{WIN} is a sufficient parameter for deciding on how comfortably a peer can request future chunks. With this motivation, if a peer has very high buffer size and consequently long s_{WIN}, it would have the right to freely request chunks without constraining the play-out deadlines and prioritization of chunks among layers. This approach increases the diversity of the chunks from all layers over the network, thus it helps relieving the load on the main seed server. On the other hand, peer having difficulties retaining continuous play-out has to prioritize the base chunks with closer deadlines. In this regard, we model the effect of the play-out deadlines on prioritization of the chunks with half-normal distribution f_T,

$$f_T(t;\, \sigma_1) = \frac{\sqrt{2}}{\sigma_1\sqrt{\pi}}\, e^{\frac{-t^2}{2\sigma_1{}^2}} \qquad t > 0 \qquad (2)$$

where the standard deviation is set as the current window size.

$$\sigma_1 = s_{WIN} \qquad (3)$$

In addition to that, the prioritization of chunks among layers is modeled with exponential distribution f_L with standard deviation λ_1 instead of half-normal distribution in order to put emphasis on the importance of the layers efficiently.

$$f_L(l;\, \lambda_1) = \begin{cases} \lambda_1{}^{-1}\, e^{\lambda_1 l} & l \geq 0 \\ 0 & l < 0 \end{cases} \qquad (4)$$

We set λ_1 as the linearly mapped version of σ_1 according the shortest and longest window sizes allowed, s_{WIN_MIN} and s_{WIN_MAX} respectively.

$$\lambda_1 = \frac{(\sigma_1 - s_{WIN\_MIN} + \epsilon) * s_{WIN\_MAX}}{s_{WIN\_MAX} - s_{WIN\_MIN}} \qquad (5)$$

Figure 3-8 depicts how f_T spreads over time axis by varying the current window size. As the window size increases due to current buffer state, the areas of the regions coinciding with the consecutive time segments approach to each other. On the other hand, decreasing s_{WIN}, areas of regions under f_T are differentiated where most of the areas correspond to closer play-out deadlines.

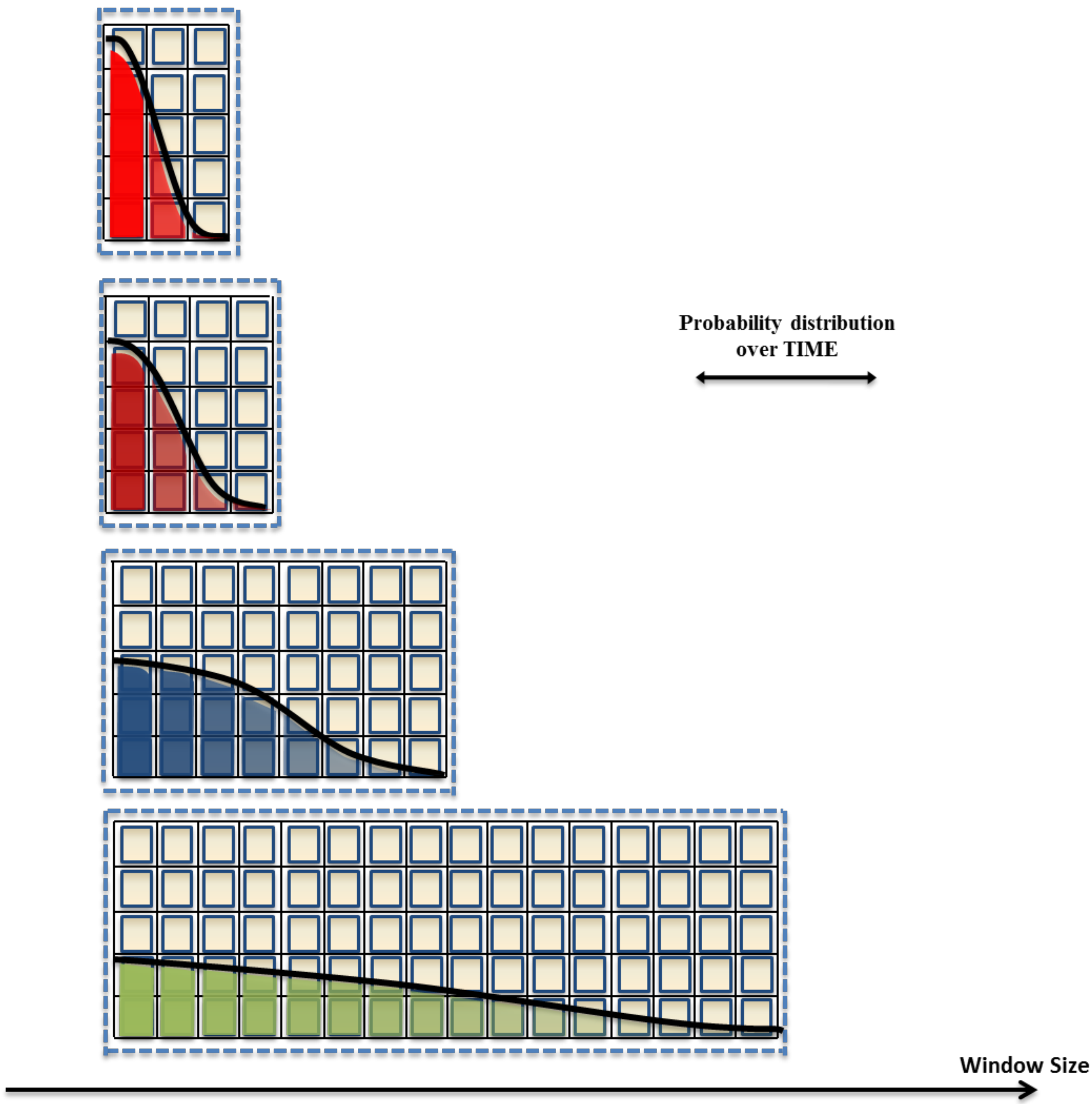

Figure 3-8 Half-Normal priority distribution of chunks for different window size

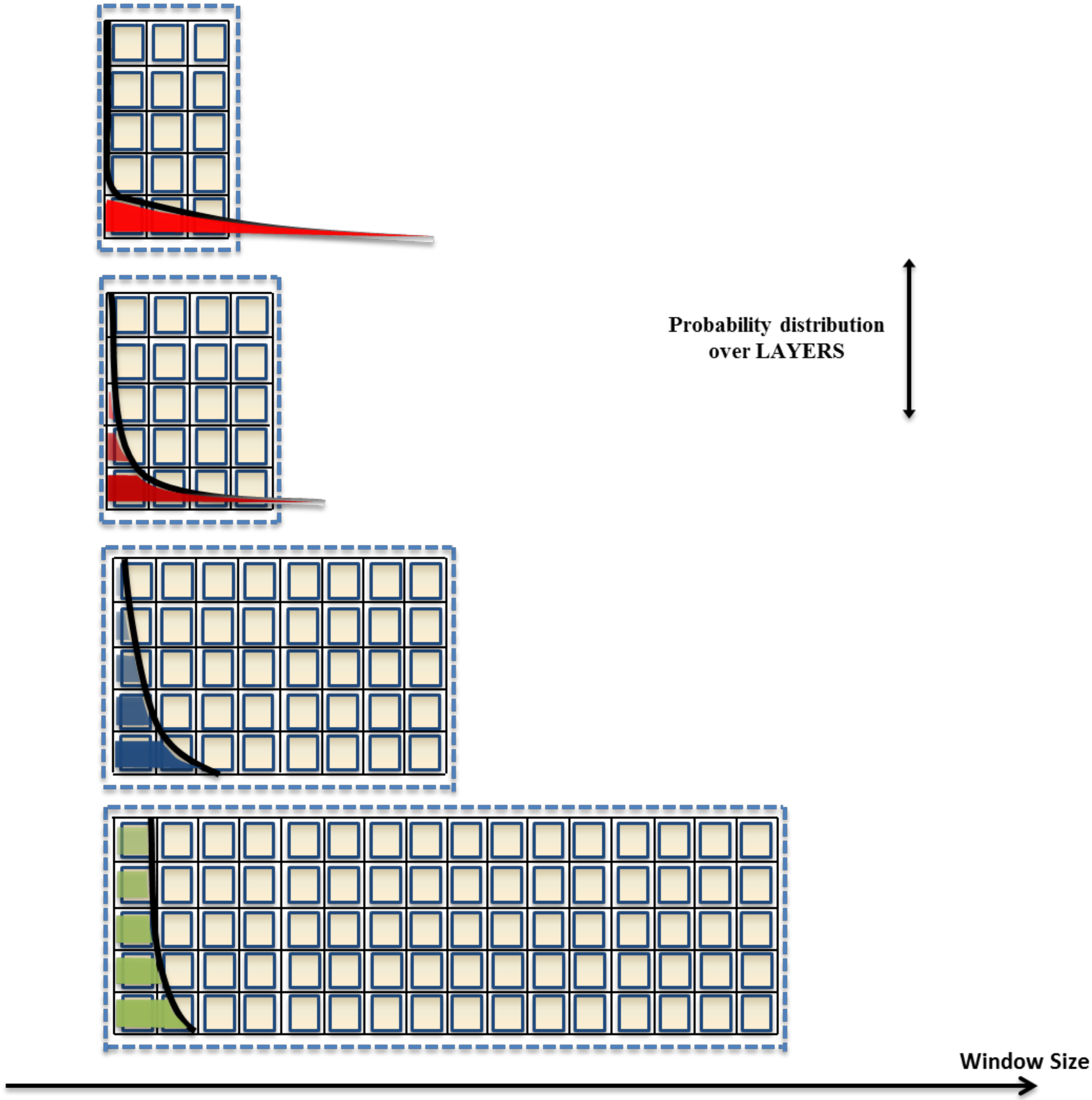

Figure 3-9 Exponential priority distribution of layers for different window sizes

In Figure 3-9 same behavior can be observed in layer dimension but with more drastic changes due to exponential distribution. As s_{WIN} increases, the areas of regions corresponding to different levels of layers become closer. On the contrary, no area is located for enhancement layers when s_{WIN} is minimum.

The exact values for prioritization of chunks scheduling are determined by calculating the dynamics weights of the chunks according to amount of areas corresponding to specific time segments and layer ids as depicted in Figure 3-8 and Figure 3-9. The adaptive prioritization coefficients are modeled with multinomial distribution $f_{T,L}$ where the probability mass function is calculated based on cumulative distribution functions over time and layer, F_T and F_L respectively. The step sizes, $step_{Time}$ and $step_{Layer}$, indicates the increments per chunk duration and per layer jump respectively; and are calculated as the following;

$$step_{Time} = F_T^{-1}(0.99)/s_{WIN} \tag{6}$$

$$step_{Layer} = F_{L_{BASE}}^{-1}(0.99)/N_L \tag{7}$$

where $F_{L_{BASE}}$ is the cdf of exponential distribution $f_{L_{BASE}}$ with standard deviation is set to minimum window length, s_{WIN_MIN}. The probability components along time axis is calculated as

$$\begin{aligned} p_t &= P(step_{Time} * (t-1) < T \leq step_{Time} * t) \\ &= F_T(step_{Time} * t) - F_T\big(step_{Time} * (t-1)\big) \end{aligned} \tag{8}$$

where p_t specifies the coefficient of selecting a chunk from time t. With same approach the components along layer axis is calculate as

$$\begin{aligned} p_l &= P\big(step_{Layer} * (l-1) < L \leq step_{Layer} * l\big) \\ &= F_L\big(step_{Layer} * l\big) - F_L\big(step_{Layer} * (l-1)\big) \end{aligned} \tag{9}$$

where p_l defines the coefficient of selecting a chunk from layer *l*. Consequently the probability of selecting a chunk from time *t* and layer *l*, $\mathrm{p}_{t,l}$ is calculated as

$$f_{T,L}(t,l) = P(T = t \text{ and } L = l) = \mathrm{p}_{t,l} = p_t * p_l \tag{10}$$

In Table 3-1 through Table 3-8 the possible chunk scheduling probabilities of two layered and five layered content for different states of the play-out are presented. When peer has low buffer, it needs to drop stream, that is, it should not request enhancement layer chunks in order to maintain continuous play-out. Table 3-1 and Table 3-5 indicate such situation where zero probabilities are assigned for all selections of enhancement layers requests. This behavior verifies that our method meets the first design goal, ensuring the play-out continuity. If due to better buffer conditions windows size begins to increase, less number of enhancement layers receives zero probability of selection as presented in Table 3-2. Note that the probabilities in both time axis and layer axis get closer to each other in Table 3-4 and Table 3-8 as compared to tables featuring shorter window lengths. The reason for this is that, in our model, as the current buffer conditions gets better and better, the probability distribution of chunk selection approaches to uniform distribution, which helps to increase the diversity of chunks among peers and accordingly to decrease the server load as suggested by the fourth design goal. The bandwidth wasting problem, which is the reason for the third design goal, is also minimized in our model since the probability of receiving higher layer chunks without receiving lower layers are exponentially minimized with very low variance in case of low buffer duration. On the other hand, for large window sizes, it is possible to receive layered chunks without the hierarchical order. However, peer would have sufficient buffer duration to request and receive the intermediate layer chunks since long window lengths are the reflections of high buffer durations. The bandwidth utilization, the second design goal of a chunk scheduling, is actually met by the decision of when to slide the window in our model rather than dynamically changing the probability distributions.

After each chunk request, the probability distribution function $f_{T,L}$ is updated. The probability of the scheduled chunk is distributed among the unscheduled ones within the current window according to their current probabilities.

Table 3-1 Sample chunk selection probabilities for 5 layer content when window size is 3

	t	t+1	t+2
EL 4	0.000	0.000	0.000
EL 3	0.000	0.000	0.000
EL 2	0.000	0.000	0.000
EL 1	0.000	0.000	0.000
BL	0.574	0.324	0.102

Table 3-2 Sample chunk selection probabilities for 5 layer content when window size is 4

	t	t+1	t+2	t+3
EL 4	0.000	0.000	0.000	0.000
EL 3	0.000	0.000	0.000	0.000
EL 2	0.003	0.002	0.001	0.000
EL 1	0.035	0.025	0.013	0.005
BL	0.411	0.296	0.153	0.057

Table 3-3 Sample chunk selection probabilities for 5 layer content when window size is 8

	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7
EL 4	0.014	0.012	0.011	0.008	0.006	0.004	0.002	0.001
EL 3	0.023	0.021	0.017	0.014	0.010	0.006	0.004	0.002
EL 2	0.037	0.034	0.029	0.022	0.016	0.010	0.006	0.003
EL 1	0.061	0.056	0.047	0.037	0.026	0.017	0.010	0.006
BL	0.100	0.092	0.078	0.061	0.043	0.028	0.017	0.009

Table 3-4 Sample chunk selection probabilities for 5 layer content when window size is 16

	t	t+1	t+2	t+3	t+4	t+5	...	t+12	t+13	t+14	t+15
EL 4	0.015	0.015	0.015	0.014	0.012	0.011	...	0.003	0.002	0.002	0.001
EL 3	0.019	0.019	0.018	0.017	0.015	0.014	...	0.004	0.003	0.002	0.001
EL 2	0.023	0.022	0.021	0.020	0.018	0.017	...	0.004	0.003	0.002	0.002
EL 1	0.028	0.027	0.026	0.024	0.022	0.020	...	0.005	0.004	0.003	0.002
BL	0.033	0.033	0.031	0.029	0.027	0.024	...	0.006	0.005	0.003	0.003

Table 3-5 Sample chunk selection probabilities for 2 layer content when window size is 3

	t	t+1	t+2
EL 1	0.000	0.000	0.000
BL	0.574	0.324	0.102

Table 3-6 Sample chunk selection probabilities for 2 layer content when window size is 4

	t	t+1	t+2	t+3
EL 1	0.0009	0.0006	0.0003	0.0001
BL	0.4475	0.3221	0.1662	0.0622

Table 3-7 Sample chunk selection probabilities for 2 layer content when window size is 8

	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7
EL 1	0.052	0.048	0.041	0.032	0.023	0.015	0.009	0.005
BL	0.181	0.167	0.141	0.110	0.078	0.051	0.031	0.017

Table 3-8 Sample chunk selection probabilities for 2 layer content when window size is 16

	t	t+1	t+2	t+3	t+4	t+5	...	t+12	t+13	t+14	t+15
EL 1	0.045	0.044	0.042	0.040	0.037	0.033	...	0.009	0.006	0.005	0.004
BL	0.073	0.072	0.069	0.064	0.059	0.053	...	0.014	0.010	0.008	0.006

Chapter 4

P2P VIDEO STREAMING EVALUATIONS

Video coding solutions SVC and MDC for P2P video streaming have been employed for better adaptation to varying channel conditions, robustness to packet losses and peer instability. Most results reported in the literature either have unrealistic P2P protocols such as centralized tree-based solutions, or they employ very basic encoding approaches with limited results from both network utilization and end-user video quality perspectives. In this chapter, we extensively compare the performances of both codecs under changing network dynamics such as congestion and unpredictable peer exits. In the following section, we initially describe the details of the overlay network where we performed the simulations. For the evaluations of the codecs, firstly we evaluate P2P streaming of SVC using high number of enhancement layers vs. single enhancement layer. Secondly, we evaluate SVC vs. MDC usage for P2P video streaming. Moreover, we also consider different P2P streaming approaches for the evaluations, which are basically push-based (sender-driven) and pull-based (receiver-driven) streaming solutions, even though the major focus of this study is pull-based P2P streaming over mesh networks.

4.1 Overlay Network Description

The streaming tests are performed over a partially connected mesh network that consists of n_L leecher peers, n_S seeder peers and n_{MS} main seed servers. While seeders and servers surely possess the whole chunks of the content, each leecher peer has direct connection to one of the seeders and the servers. These direct connections enable access to all chunks through a seeder or server at the worst case. Such a situation may occur when a leecher peer is not able to get a chunk from its leecher neighbors. There are bidirectional connections among leecher peers, where each leecher has n_N neighbors (including a server and a seeder) where n_N is a uniform random number between $n_{N\text{-}Max}$ and $n_{N\text{-}Min}$. Limitation in number of neighbors allows lessening of the messaging overhead, e.g., bit-map exchange messages among neighbors, during the streaming.

In order to evaluate the effect of changing network conditions on the performance of the system, two network dynamics are considered: 1) congestion on the links (edges) and 2) ungraceful peer (node) exits. For varying bandwidth on the links, the Internet traffic modeling is taken into account. At every T_{CROSS} seconds each uplink is introduced with cross traffic that is modeled as an independent Poisson distribution with mean λ, which is a convenient model for bursty nature of the Internet [21]. The maximum upload capacities that can be achieved by the seeder and leecher peers are chosen between U_{MIN} and U_{MAX}, and these are set to different values for different scenarios, e.g., mixed rate peer pool where there is a remarkable difference on uplink capacities of peers. Since this study assumes that peers have asymmetric Internet connections and limitations are only due to the upload rates, dynamics related to downlinks are not considered. As a second network dynamics, the unpredictable peer exits, a probability p_L for ungraceful leave events is set, and periodically executed. The value for p_L is decided according to scenarios' requirements, for instance, it is set so that 25% of the peers leave the network by the end of the streaming session.

4.2 Evaluation of Video Streaming Using SVC over P2P Networks

In this section, we analyze the effect of using higher number of layers on mesh-pull based P2P streaming protocol using SVC, which is discussed in Chapter 3. Increasing number of layers reveals finer adaptation capability, while with two layer coding coarse quality variances may occur in case of bandwidth oscillations.

4.2.1 Test Parameters and Scenarios

P2P tests are performed for 6 different test conditions to cover a wide range of possibilities. In first four groups of these scenarios, the average upload rates vary up to 25% for different regions of available bandwidth. These regions are selected as adjacent to bitrates of enhancement layers in Table 4-1, while they are selected for higher bitrates than corresponding enhancement layers in Table 4-2. In these tables, the rate-distortion values of different number of layered SVC approaches for approximately 50 seconds length two contents with GOP size 16, *In to Tree* at 50fps and *Soccer* at 30fps, are presented. Moreover, for the last two scenarios, the bitrate variation can go up to 60% for the purpose of testing over highly dynamic networks. For instance, if average upload rate is set to 2Mbps, it varies between 1.5Mbps to 2.5Mbps for the first four scenarios and between 0.8Mbps to 3.2Mbps for the latter scenarios due to cross traffic that is updated at every T_{CROSS}. The traffic is updated more frequently in last two scenarios. All of the six scenarios are simulated for peers with both homogeneous and heterogeneous upload capacities. The dynamic windowing mechanism is enabled with initial window size $S_{WIN\text{-}INIT}$ set to 5 chunk duration and the initiation delay set to 7 chunk duration. RTT has also great effect on the performance of pull-based system and it is randomly chosen between RTT_{MAX} and RTT_{MIN} for each pair of peers. The rest of the simulation parameters and descriptions are presented in Table 4-3.

Table 4-1 Bitrate vs. PSNR of *Soccer* content

Soccer		Base	EL 1	EL 2	EL 3	EL 4
2 Layer	Bitrate (Mbps)	0.637	1.514			
	PSNR (dB)	32.61	35.51			
5 Layer	Bitrate (Mbps)	0.505	0.814	1.279	1.766	2.535
	PSNR (dB)	31.48	32.92	34.18	35.05	36.14

Table 4-2 Bitrate vs. PSNR of *In to Tree* content

In To Tree		Base	EL 1	EL 2	EL 3	EL 4
2 Layer	Bitrate (Mbps)	0.676	1.358			
	PSNR (dB)	31.36	33.33			
5 Layer	Bitrate (Mbps)	0.508	0.827	1.320	2.118	3.738
	PSNR (dB)	30.49	31.68	32.72	33.68	34.8

Table 4-3 Simulation Parameters

n_{MS}	1	# of main seed server
n_S	4	# of main seeding peer
n_L	100	# of leecher peer
$n_{N\text{-}Max}$	10	maximum # of neighbors
$n_{N\text{-}Min}$	5	minimum # of neighbors
C_{MS}	15	main seed server constant
T_{CROSS}	5 and 2	cross traffic update periods (in seconds)
p_L	0	peer exit probability
$S_{WIN\text{-}MIN}$	3	maximum size of the window (in chunks)
$S_{WIN\text{-}MAX}$	30	minimum size of the window (in chunks)
$S_{WIN\text{-}INIT}$	5	initial size of the window (in chunks)
C_{WIN}	0.66	window constant defining reliance on buffer
L_{PLAY}	~50 sec	Content durations
t_B	7	pre-buffering delay (in chunks)
n_{CU}	2	# of maximum concurrent uploads
n_{CD}	2	# of maximum concurrent downloads
λ	10	mean of Poisson distribution for cross traffic
RTT_{MIN}	50 ms	minimum latency
RTT_{MAX}	100 ms	maximum latency

4.2.2 Results and Discussion

Table 4-4 and Table 4-6 present the P2P streaming results that are computed by averaging the PSNR values of the decoded stream for each one of the receiving n_L leecher peers. Each table contains results for each scenario when 2-layer and 5-layer SVC coding is employed. Moreover, in Table 4-5, results for the case in which heterogeneous peers are considered are presented for *Soccer* content. In all experiments 100% base layer chunk delivery has been achieved making each peer experience continuous play-out. The P2P activity, which is determined based on the source of the received chunks, is varied between 65% and 80% for different scenarios where 100 leecher peers assisted to each one of their neighbor leechers.

Table 4-4 Average PSNR Results for *Soccer* with Homogeneous Upload Capacities

		%20-25 Bandwidth Variation				%50-60 Bandwidth Variation	
Scenario		1	2	3	4	5	6
Average Bitrate (Mbps)		1.25	1.5	2	3.5	1.5	2
Average PSNR (dB)	2 Layer	33.22	33.8	34.89	35.51	34.31	35.51
	5 Layer	32.73	33.36	34.28	36.13	34.01	34.62

Table 4-5 Average PSNR Results for *Soccer* with Heterogeneous Upload Capacities

		%20-25 Bandwidth Variation				%50-60 Bandwidth Variation	
Scenario		1	2	3	4	5	6
Average Bitrate (Mbps)		1.25	1.5	2	3.5	1.5	2
Average PSNR (dB)	2 Layer	33.66	33.96	34.63	35.51	33.92	35.37
	5 Layer	33.27	33.37	34.37	35.89	33.13	34.36

Table 4-6 Average PSNR Results for *In to Tree* with Homogeneous Upload Capacities

		%20-25 Bandwidth Variation				%50-60 Bandwidth Variation	
Scenario		1	2	3	4	5	6
Average Bitrate (Mbps)		1.8	3	4	5	3	4
Average PSNR (dB)	2 Layer	33.05	33.33	33.33	33.33	33.33	33.33
	5 Layer	32.21	33.25	33.35	33.69	33.63	34.05

Results indicate that the average PSNR values with higher number of layers are below the values when 2-layer SVC is used. Although adaptation points increases with higher number of layers, it brings two kinds of overhead for streaming in pull-based P2P: 1) coding overhead as presented in Figure 2-3, Table 4-1 and Table 4-2, and 2) enhancement chunk request messaging delays in P2P. The former is discussed in Section 3.1.2. The latter kind of overhead, which is due to P2P messaging, depends on the overlay and bandwidth conditions, such as *RTT* and number of available neighbors. For each one of the upper layer chunks, peers perform a new request that makes peer loose time. At the best case, it costs one *RTT* per chunk. Moreover, if a peer is unable to find an available neighbor for receiving current chunk, the messaging delay linearly increases as it asks for further availability of its remaining neighbors. In order to achieve the highest quality with 5-layer scalable content, the probability of losing time due to messaging and consequently having a decreasing play-out buffer increases. On the other hand, in P2P streaming of 2-layer SVC content, overheads due to both coding and P2P messaging are diminished. According to the first scenario in Table 4-4 for *Soccer* sequence, the available bandwidth varies between the base quality rate and the full quality rate for 2-layer SVC, while it is around third enhancement quality rate for 5-layer case. The overall received quality is 33.22 dB and 32.73 dB, for 2 and 5-layer coding, respectively. The utilization of the available bandwidth is higher when 2-layer SVC is employed, since with 5-layer the received quality remains slightly under second enhancement quality even the available bandwidth is adjacent to rate of third enhancement layer rate. This situation arises due to messaging overhead introduced by higher layers as discussed above. Same observations apply for scenarios 2-4. Comparing scenarios 2 with 5, and 3 with 6, it is obtained that the overall quality is higher in 5 and 6 even though the cross traffic variation is higher. Since in these highly varying bandwidth scenarios the upload capacities may go beyond high bitrates, the buffering capability of peers increase and consequently compensate the effects of very low bitrates. On the other hand, the load on main seed server, in which the server is determined as if it is C_{MS} (set as a simulation parameter) times powerful in terms of bandwidth and capability of concurrent service than regular peers, increases and higher number of chunks are received from the server, which is not much desired for successful P2P systems.

In Table 4-5 the results for the same scenarios, but with heterogeneous peers in terms of upload capacities are presented. In the heterogeneous case, U_{MIN} is set to approximately half of U_{MAX} while for the homogeneous case they were equal. Comparing 2-layer and 5-layer SVC usage, same observations related to messaging overhead can be performed. In addition, if we compare results between homogeneous and heterogeneous overlays, it is not easy to conclude for scenarios with low bandwidth variation. However, with highly dynamic upload capacities in the last two scenarios, higher quality is achieved in homogeneous networks.

Experiments are repeated for another sequence *In to Tree*, which has higher frame rate compared to *Soccer*. In the same way with chunking of *Soccer* bitstream, each chunk of *In to Tree* contains a GOP. Since frame rate is higher in *In to Tree* sequence, higher number of chunks needs to be received compared to chunks of *Soccer* for identical play-out duration. Table 4-6 presents the results where the effect of higher number of layering can be observed similar to Table 4-4. However, even with 5Mbps average bitrate in the 4th scenario, full quality cannot be achieved in 5-layer scenario. Along with the effect of higher number layers, packing a GOP with size 16 into single chunks decreases the bandwidth utilization for *In to Tree*, whose frame rate is 50fps, since higher number of messaging is required in time axis also. As a solution, higher number of GOPs can be packed into a chunk and consequently peers can request fewer chunks for a given play-out interval so that higher utilization of the available bandwidth when streaming contents with high frame rates can be achieved. Nevertheless, this approach slightly reduces the adaptation granularity.

Average upload rates less than the ones in the first scenarios are also employed in our initial experiments. However, in those cases, some peers could not receive certain base chunks especially during the initial phases of the streaming session. Chunk availability at the leecher peers is not frequent in these initial phases. Therefore chunks need to be transferred over multiple peers, which can be called as multi-hopping, until they reach to requesting peer. On the other hand, *RTT* affects the delivery time of the chunks from one peer to another and the total transmission time from first peer to last peer of hopping linearly increases. Although main seed server highly assists at these stages, high number of leecher peers may request from the server; and consequently the transmission time increases. Due to either the effect of multi-hopping or congested server, if the transmission time of a chunk is longer than the buffer duration at the request moment, peers may miss the base chunk and experience discontinuous play-out.

Comparison of 2-layer and 5-layer SVC streaming from overall peers' perspective has been done previously. Figure 4-1 through Figure 4-6 depict the received quality histograms of peers for each scenario for which the overall received quality is presented in Table 4-4. Each one of the bars in these figures represents timely averaged received quality per peer; and as observed in the histograms, these received PSNRs slightly vary from one peer to another. Moreover, a few peers using 2-layer SVC receive lower quality than the ones with 5-layer SVC for the first, second, and fifth scenarios. Even in dynamically changing bandwidth scenarios, 5th and 6th, the overall received quality was higher with 2-layer SVC. For the 6th scenario, such case is natural since the upper bound of the changing average upload rate was beyond the full quality rate of the 2-layer SVC. On the other hand, 5th scenario requires more detailed analysis and comparison of 2-layer vs. 5-layer along the time axis considering possible instantaneous quality switches, which are not favorable for end users [44], experienced by peers. To this end, additional two experiments for the analysis of PSNR change through time have been performed for a sample peer.

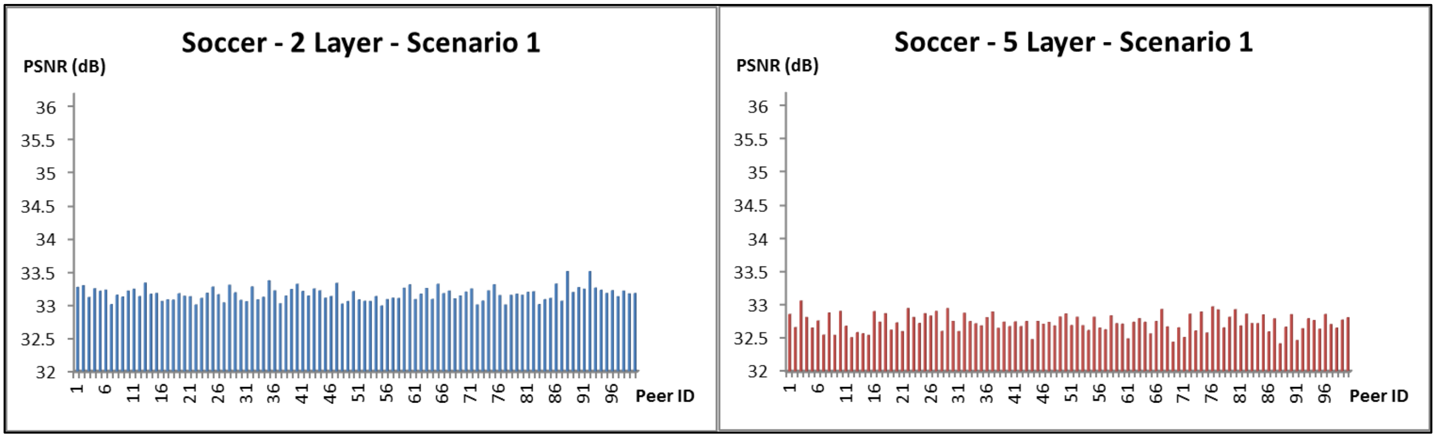

Figure 4-1 Soccer PSNR comparison for scenario 1

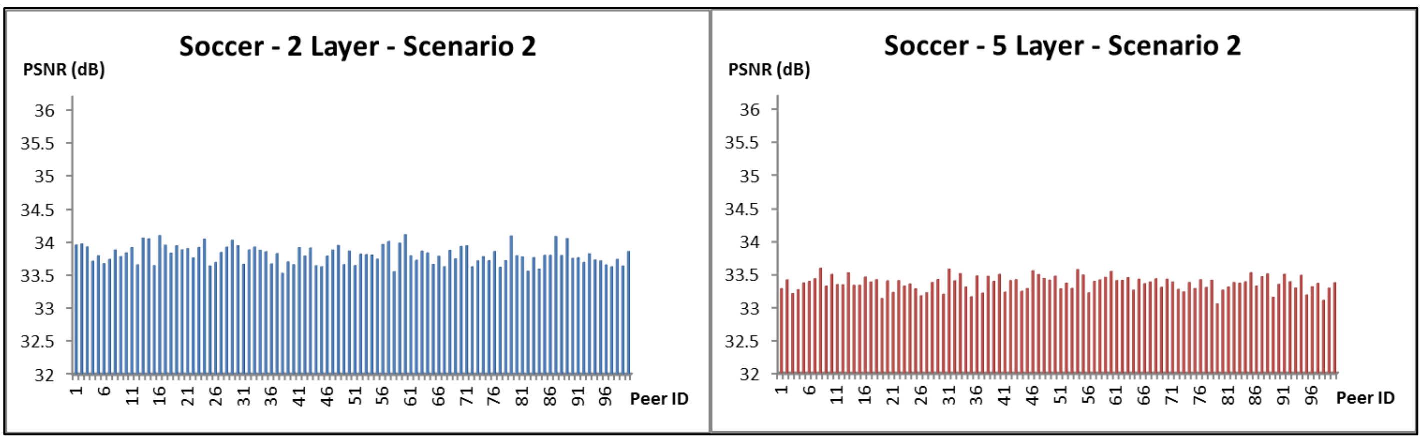

Figure 4-2 Soccer PSNR comparison for scenario 2

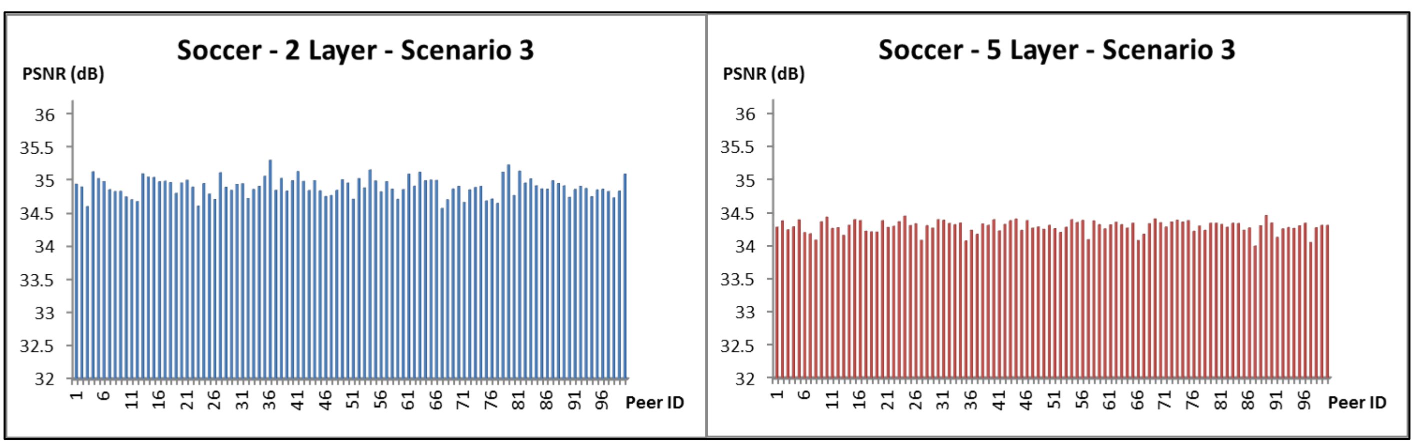

Figure 4-3 Soccer PSNR comparison for scenario 3

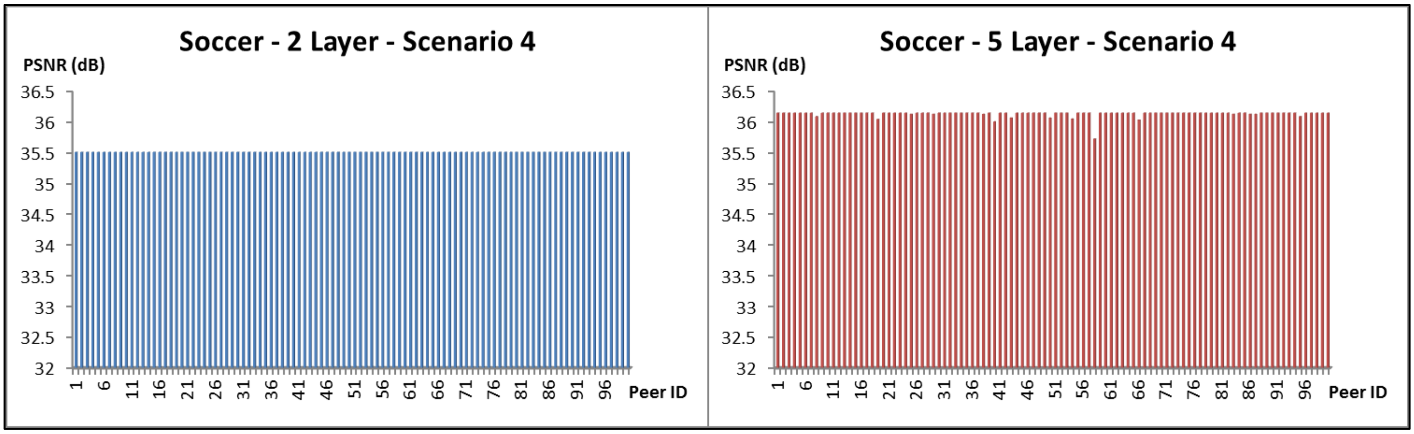

Figure 4-4 Soccer PSNR comparison for scenario 4

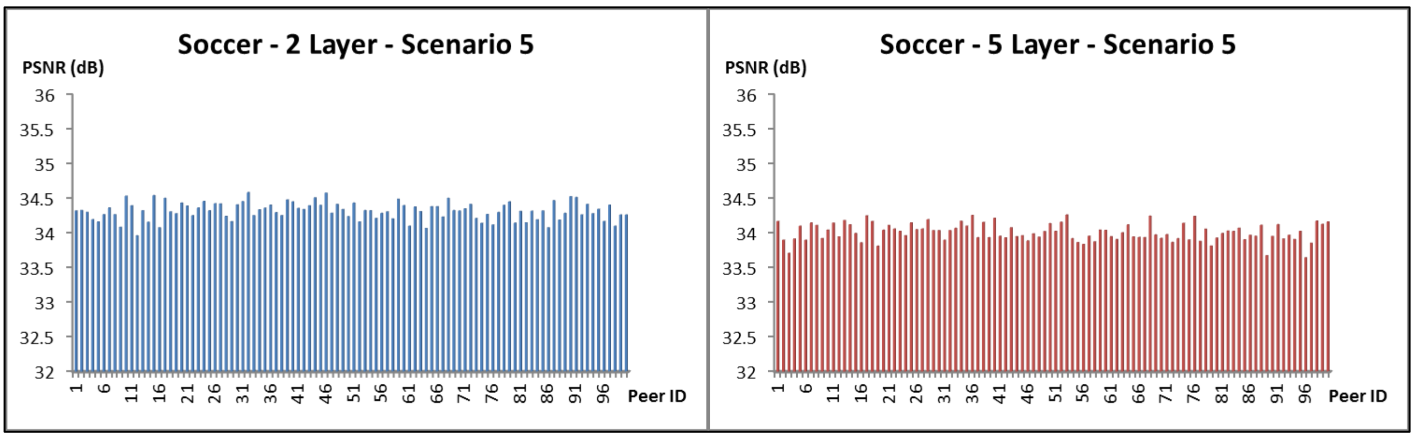

Figure 4-5 Soccer PSNR comparison for scenario 5

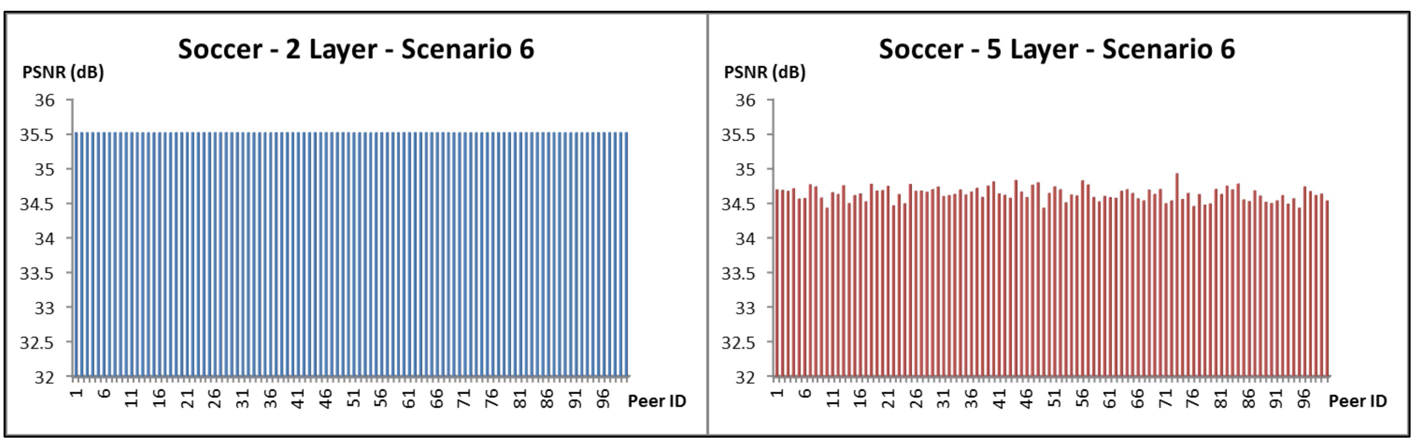

Figure 4-6 Soccer PSNR comparison for scenario 6

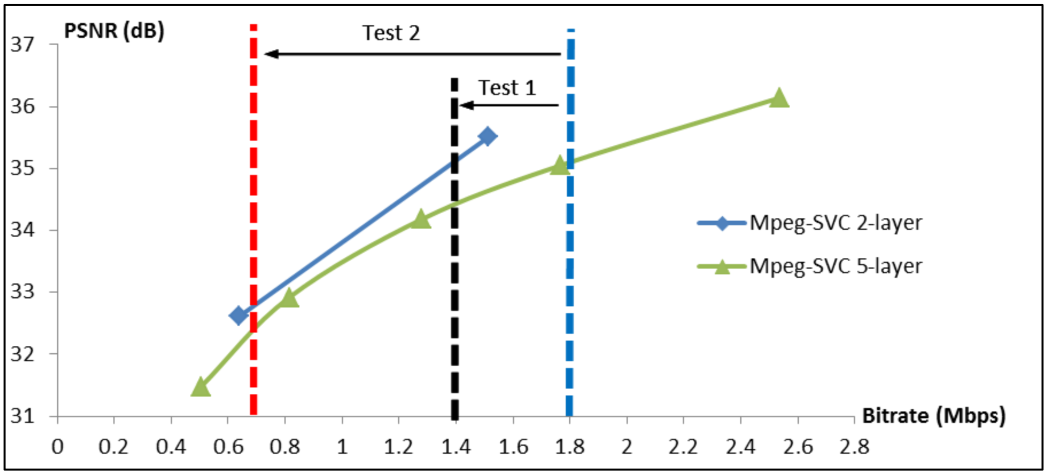

Figure 4-7 the Effect of Available Bandwidth Drop on Quality Change

The goal of these additional two tests is to evaluate the usage of 2-layer and 5-layer SVC in case of bandwidth drops as described in Figure 4-7. Test 1 covers the situation in which the available bandwidth of a peer drops from 1.8 Mbps to 1.4 Mbps. In such a case, the available bandwidth stays slightly below the full quality bitrate for 2-layer SVC while it stands above the third quality layer bitrate for 5-layer SVC. In addition, for Test 2 higher bandwidth drop is simulated for both coding. In both tests the drops occur at the 15th second of the play-out while streaming ~50 seconds *Soccer* content, as depicted in Figure 4-8. Figure 4-9 presents the PSNR changes for Test 1. It has been observed that when 5-layer SVC is employed, the bandwidth drop does not cause sharp PSNR drop, which is expected naturally for 5-layer SVC since it offers more adaptation points. On the other hand, sharp drop is also not observed with 2-layer SVC even though the instantaneous available bitrate after the bandwidth drop is not sufficient to receive the full quality. In Test 2, the available bitrate after the 15th second is only sufficient to receive base layer chunks for both 2-layer and 5-layer. Therefore sharp drops occur in both coding approaches, but high quality reception lasts longer around +6 seconds with 2-layer SVC as presented in Figure 4-10.

Figure 4-8 Available Bandwidth Drop for Quality Change Test 1 and Test 2

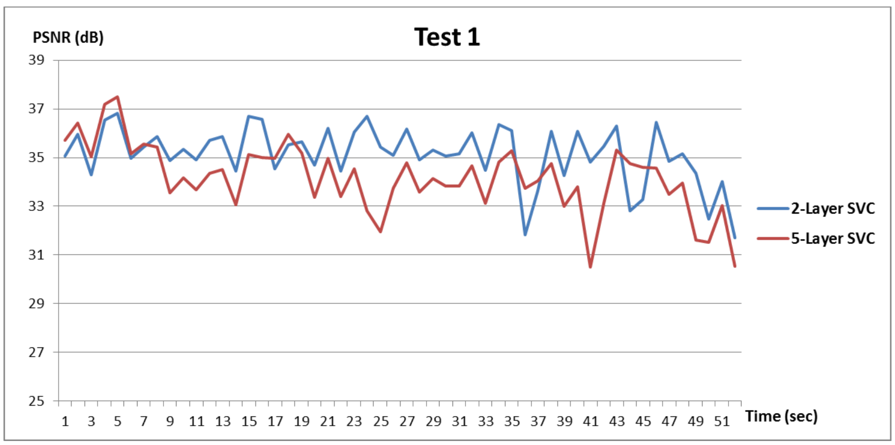

Figure 4-9 PSNR Change of a Peer for Quality Change Test 1 with 2 and 5-Layer SVC

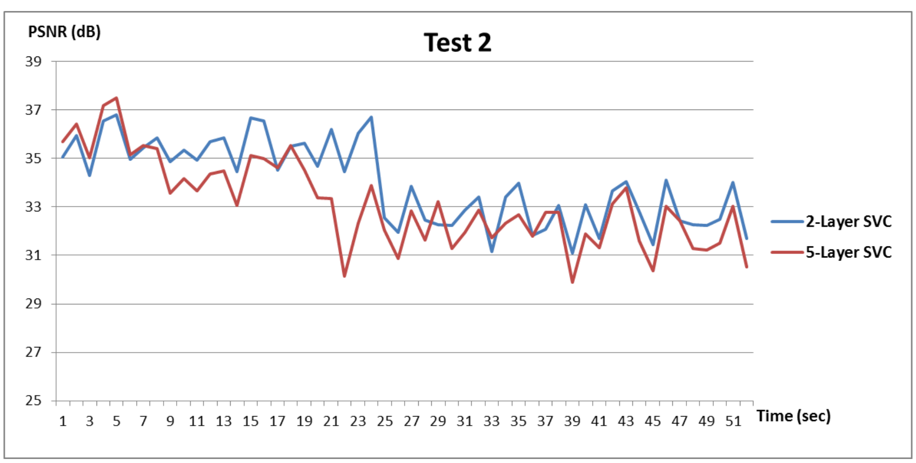

Figure 4-10 PSNR Change of a Peer for Quality Change Test 2 with 2 and 5-Layer SVC

The reason behind this situation is that since RD performance of 2-layer SVC is better as in Figure 4-7, it has higher buffer duration at the 15th second compared to buffer duration when streaming 5-layer SVC.

Summarizing the results from these P2P streaming experiments of the scalable video, we might conclude at two important points. Firstly, considering the overall received quality of all peers, employing 2-layer SVC coding yields higher overall quality compared to higher number of layers due to both coding efficiency and less messaging. Secondly, although SVC with higher number of enhancement layers promises more scalability and adaptation points, even for situations in which the bandwidth varies around the adjacent bitrates of the enhancement layers, employing 2-layer SVC is sufficient for experiencing continuous play-out with negligible quality changes and it provides higher quality almost at every time instant for pull-based P2P streaming.

4.3 Comparison of SVC vs. MDC over P2P Networks

In this section, we investigate the performances of SVC and SVC-based MDC over P2P networks with packet level simulations. First, we perform streaming experiments over sender-driven (push-based) overlays. Secondly and most importantly, evaluations are performed over torrent-like receiver-driven (pull-based) networks, for which the studies in the literature stays unsatisfactory as discussed in 2.3.2. The tests are performed considering mainly two types of problem in P2P streaming: congestion in uplinks and ungraceful peer exits.

4.3.1 **Evaluation over Sender-Driven P2P Networks**

In addition to our main focus, BitTorrent-like adaptive P2P streaming solution, discussed in Chapter 3, we also developed a basic sender-driven adaptive video streaming solution using Peersim. The simulations are performed over single source partially connected mesh network in which each peer is responsible for pushing chunks within its buffer. Buffers may congest rapidly in case of having inadequate upload bandwidths. For this reason, peers discard sufficient number of enhancement chunks within their buffers starting from the ones buffered earlier, so that they can keep up with the play-out time by keeping base layer chunks and remaining enhancement layer chunks. However, if a peer is about to fail serving its neighbors at the playback rate even it has already discarded all of the enhancement layer chunks from its buffer; it may start dropping the base layer chunks in the same manner. The outcome of this condition is observed as it depends on the encoding scheme of the video.

4.3.1.1 Evaluation Criteria and Simulation Parameters

The evaluation criteria for the performance of a P2P streaming solution are not well defined and can be very complex. In most of the studies on video streaming over P2P networks, researchers have used pure networking criteria such as packet loss and delay, neglecting the effect of video. On the other hand, regarding only the video aspects such as the average received PSNR value for all peers does not provide accurate insight. Error concealing decoders may cover the decrease in the quality of the video due to frame skipping. Inefficient utilization of the network also affects the throughput and corresponding evaluation. If a full quality receiving peer consumes its bandwidth with redundancy, this also does not provide reliable quality evaluation due to low utilization of the network.

IPTV subscribers expect to receive high quality service to have satisfaction on the quality of viewing experience where frequent pixellization (blocky artifacts) or frame skipping is unacceptable. Based on this understanding, we classified the peers into two categories: acceptable and non-acceptable quality receiving peers. If a peer is not able to receive at least two base layer chunks on time or it receives the video at lower quality than the base quality, then it is marked as unacceptable. Therefore, we provide two results, first the ratio of the acceptable peers and then their average peak signal to noise ratio (PSNR) value. Therefore, high PSNR with a low number of satisfying peers is not a good result evaluating the overall end-user quality of experience.

All simulations are performed with 100 peers connected to a maximum number of 10 neighbors which are randomly chosen. The simulated overlay network consists of homogenous peers in terms of the upload bandwidths with more or less ten percent variation.

Table 4-7 Content Bitrates for SVC and SVC-based MDC Descriptions (Kbps)

	Adile			Train		
	SVC	**Desc1**	**Desc2**	**SVC**	**Desc1**	**Desc2**
Base Rate	296	296	296	569	569	569
Enhc Rate	543	275	267	935	472	463
Total Rate	839	571	563	1504	1041	1032

Table 4-8 PSNR (dB) Values of SVC and SVC-Based MDC Streams

`	Adile			Train		
	SVC	**Desc1**	**Desc2**	**SVC**	**Desc1**	**Desc2**
Base PSNR	32,24	32,24	32,24	30,62	30,62	30,62
Full PSNR	38,55	35,27	35,34	35,42	32,87	32,92

Two different 30-second sequences, *Adile* and *Train*, are encoded with SVC and SVC-based MDC as in [49], where bitrates and PSNR values are given respectively in Table 4-7 and Table 4-8. The buffer lengths of the peers are set to 2 seconds which is reasonable for achieving smooth streaming. In our simulations, we have performed several tests using SVC and SVC-based MDC with varying parameters, such as peer exit rate and average upload capacity.

4.3.1.2 Results and Discussion

We have seen that when peers leave the network more frequently, SVC-based MDC provides higher number of satisfied peers, although the received quality is lower due to redundancy introduced. Table 4-9 and Table 4-10 present the P2P simulation results for contents *Adile* and *Train*, respectively. In each table, the first sub-table presents the results for using SVC based-MDC while the second one presents for using SVC. Each result cell either represents the percentage of acceptable quality receiving peers (AQRP) or the overall quality (PSNR) received (OQR) over all peers in acceptable quality receiving group.

Table 4-9 Results for *Adile*

	Average Upload Bandwidth(Kbps)					
	512		1024		2048	
Peer exit rate	AQRP	OQR	AQRP	OQR	AQRP	OQR
1	99%	32,8	99%	38,3	99%	38,5
5	95%	32,9	95%	38,4	94%	38,5
10	91%	32,8	89%	38,5	90%	38,5

a)MDC

	Average Upload Bandwidth(Kbps)					
	512		1024		2048	
Peer exit rate	AQRP	OQR	AQRP	OQR	AQRP	OQR
1	99%	34,5	99%	38,3	99%	38,5
5	93%	35,7	95%	38,3	94%	38,5
10	72%	33,6	90%	36,6	90%	38,5

b)SVC

Table 4-10 Results for *Train*

Peer exit rate	Average Upload Bandwidth(Kbps) 512		1024		2048	
	AQRP	OQR	AQRP	OQR	AQRP	OQR
1	11%	32,9	96%	31,1	99%	35,3
5	10%	32,9	94%	31,2	95%	35,2
10	10%	32,9	92%	31,4	91%	35,4

a)MDC

Peer exit rate	Average Upload Bandwidth(Kbps) 512		1024		2048	
	AQRP	OQR	AQRP	OQR	AQRP	OQR
1	10%	35,4	99%	31,9	100%	34,5
5	10%	35,4	93%	32,5	95%	34,4
10	10%	35,4	75%	34,6	90%	34,7

b)SVC

AQRP: Acceptable Quality Receiving Peer
OQR: Overall Quality Received (dB)

The effect of the peer exit rate on the quality received by the end-users can be best studied when the average upload rate of the P2P network is low. Streaming *Adile* with 512 Kbps average upload rate is sufficient for SVC. The bitrate for non-discardable layers of Adile is at 296 Kbps. For 1% peer exit rate, the overall quality received by the peers is 1.7 dB higher with 99% peer satisfaction when SVC is utilized. As peer exit rate increases, the number of peers satisfied using SVC decreases. At the 10% peer exit rate, 91% of the peers received acceptable quality with MDC, while it decreases down to 72% with the usage of SVC. Although the overall quality is higher in SVC for 10% peer exit rate, the satisfaction rate is higher with MDC and we might say that the number of peers receiving acceptable quality has higher importance than the average quality received for P2P streaming. For upload rates higher than the total of MDC descriptions' rates, no differences are observed whether MDC or SVC is employed. Similar results are achieved with the other test content. Streaming Train with 512 Kbps average upload rate is unacceptable for both coding approaches since the rate of the base layers is at 569 Kbps. Accordingly, the satisfaction percentage stays around 10% for each case, which is mainly due to the peers that have accesses the main server directly. When the average rate is at 1024 Kbps, the received quality is 1-2 dB higher with SVC. Moreover, satisfaction decreases similarly as peer exit rate increases for using SVC compared to MDC.

Our comprehensive analysis of experimental results indicates that MDC should be preferred if peer exit rate is more than 5% and the average upload rate is between base and full rate of the content, so that the number of satisfied peers is high. However if average upload capacity is only sufficient for streaming base chunks, SVC needs to be employed. On the other hand, for high upload capacity P2P networks, where upload rate average is higher than the total rate of the MDC descriptions, it does not matter whether MDC or SVC is preferred.

4.3.2 **Evaluation over Torrent-like Receiver-Driven P2P Networks**

4.3.2.1 Test Parameters and Scenarios

For the experiments of this section, we employ the MDC scheme depicted in Figure 2-7 which is based on temporal splitting of the raw sequence. In Table 4-11, RD performances of SVC and employed MDC scheme is presented for three contents featuring different characteristics in temporal and spatial domain. In the same table resultant redundancy of MDC scheme also presented. The redundancy is only due to decreasing coding efficiency of temporal splitting and we do not add residual information enhancing the quality in case of receiving single description. Among three contents, *Soccer* is the highest motion featuring content, while *Phone Call* has the least motion. Consequently, MDC redundancy in *Soccer*, which is %26, is higher than the others.

Table 4-11 Bitrates (Mbps) of Contents for MDC based on Temporal Split

		Full Sequence	Odd Frames	Even Frames	Merged	Redundancy
Phone Call	Base	0.077	0.045	0.046	0.091	0.14
	Full	0.177	0.101	0.101	0.201	
Soccer	Base	0.635	0.392	0.395	0.787	0.26
	Full	1.511	0.950	0.952	1.903	
Foreman	Base	0.228	0.139	0.139	0.279	0.24
	Full	0.452	0.281	0.281	0.562	

Table 4-12 PSNRs (dB) of Contents with Motion Compensated Temporal Interpolation

		Full Sequence	MCTI of Odd Frames	MCTI of Even Frames	Merged
Phone Call	Base	34.38	33.74	33.74	34.08
	Full	37.02	35.94	35.93	36.62
Soccer	Base	32.6	27.08	27.11	32.07
	Full	35.51	28.59	28.6	35.09
Foreman	Base	34.25	31.5	31.5	33.81
	Full	36.93	33.24	33.24	36.6

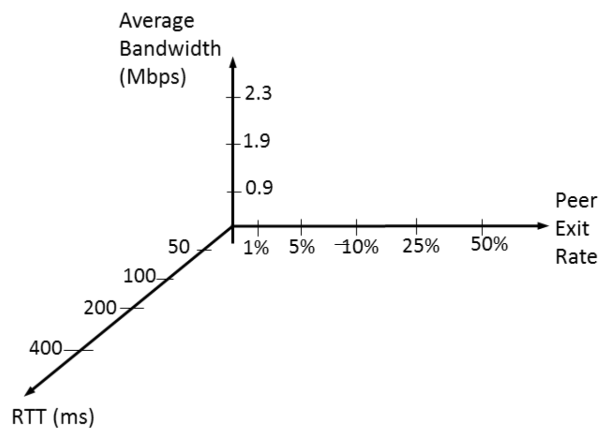

Figure 4-11 Test Parameters' Combinations

Table 4-13 Simulation Parameters 2

n_{MS}	1	# of main seed server
n_S	4	# of main seeding peer
n_L	100	# of leecher peer
$n_{N\text{-}Max}$	10	maximum # of neighbors
$n_{N\text{-}Min}$	5	minimum # of neighbors
C_{MS}	20	main seed server constant
T_{CROSS}	5	cross traffic update periods (in seconds)
p_L	Figure 4-11	peer exit probability
$S_{WIN\text{-}MIN}$	2-3	maximum size of the window (in chunks)
$S_{WIN\text{-}MAX}$	30	minimum size of the window (in chunks)
$S_{WIN\text{-}INIT}$	3-5	initial size of the window (in chunks)
C_{WIN}	0.66	window constant defining reliance on buffer
L_{PLAY}	~75 sec	Content durations
t_B	4-7	pre-buffering delay (in chunks)
n_{CU}	2	# of maximum concurrent uploads
n_{CD}	2	# of maximum concurrent downloads
λ	10	mean of Poisson distribution for cross traffic
RTT_{MIN}	Figure 4-11	minimum latency
RTT_{MAX}	Figure 4-11	maximum latency

The average PSNRs of SVC and MDC scheme is presented in Table 4-12. First column represents encoding of single sequence, while second and third column represents the MDC case when single description is received and motion compensated temporal interpolation is performed. Finally in fourth column PSNR values are presented for the case when both descriptions are received and interleaved. Since, *Soccer* has too much motion; the motion compensation performance is not as good as of *Foreman* or *Phone Call* contents and therefore it has low PSNR values when single description is received.

The extensive experiments are performed for different combinations of P2P network dynamics, which are average upload capacity, ungraceful peer exits and additionally RTT, since the tests are based on receiver-driven P2P streaming. In Figure 4-11, the values of these network dynamics have been presented, while rest of the parameters is presented in Table 4-13. Note that ungraceful peer exit notice duration is set to 2*RTT*. We use *Soccer* content for the streaming evaluation of the coding approaches, where content duration L_{PLAY} is approximately 75 seconds. Due to employed MDC scheme, number of chunks of SVC in time axis is twice the chunks of a single description of MDC as the encodings are performed with GOP size 16 for both SVC and for a description of MDC. Hence, the duration of chunk of the employed MDC scheme is twice the chunks of SVC. In order to perform a fair evaluation of the codecs, initial buffering durations must be as close as possible; therefore t_B (in chunks) is set 3 and 5 for MDC and SVC, respectively.

4.3.2.2 Results and Discussion

Evaluation results are presented in Table 4-14, Table 4-15 and Table 4-16. Each table contains results for different average upload rates that are varied 10% during a streaming session. Columns stand for different RTT values, while rows have different peer exit rates. Note that the peer exit rates are approximate and may vary by 1%. Each macro-cell in a table consists of eight cells and contains result of a single simulation experiment.

Table 4-14 SVC vs. MDC Comparison for *Soccer* in Pull-based P2P Streaming for Different Combinations of RTT and Peer Exit Rate when Average Upload is 0.9 Mbps

			RTT (ms)							
			50		100		200		400	
			Full	Half	Full	Half	Full	Half	Full	Half
~ Peer Exit Rate 1%	SVC	Base + Enh	1052 - 7.48%	-	1023 - 7.27%	-	979 - %6.96	-	937 - 6.66%	-
		Base	8980 - 63.8%	-	8022 - 57.06%	-	6638 - 47.21%	-	5254 - 37.37%	-
	MDC	Base + Enh	702 - 9.98%	98 - 1.39%	699 - 9.94%	91 - 1.29%	694 - 9.87%	66 - 0.93%	694 - 9.87%	33 - 0.46%
		Base	3082 - 43.84%	2892 - 41.14%	2637 - 37.51%	3020 - 42.96%	1763 - 25.08%	3090 - 43.96%	1118 - 15.90%	2727 - 38.79%
5%	SVC	Base + Enh	1018 - 7.54%	-	982 - 7.27%	-	939 - 6.96%	-	890 - 6.59%	-
		Base	8619 - 64.89%	-	7972 - 59.09%	-	6404 - 47.47%	-	5138 - 38.08%	-
	MDC	Base + Enh	380 - 5.63%	10 - 0.14%	380 - 5.63%	12 - 0.17%	380 - 5.63%	5 - 0.07%	380 - 5.63%	1 - 0.01%
		Base	2968 - 44%	2780 - 41.21%	2471 - 36.63%	2971 - 44.04%	1783 - 26.43%	2987 - 44.28%	1082 - 16.04%	2630 - 38.99%
10%	SVC	Base + Enh	964 - 7.54%	-	936 - 7.32%	-	898 - 7.02%	-	842 - 6.58%	-
		Base	8402 - 65.74%	-	7552 - 59.09%	-	6186 - 48.4%	-	4935 - 38.61%	-
	MDC	Base + Enh	360 - 5.63%	9 - 0.14%	360 - 5.63%	8 - 0.13%	634 - 9.92%	61 - 0.95%	360 - 5.63%	2 - 0.03%
		Base	2840 - 44.44%	2704 - 42.31%	2342 - 36.65%	2765 - 43.27%	1678 - 26.25%	2789 - 43.64%	1072 - 16.77%	2556 - 40%
25%	SVC	Base + Enh	814 - 7.64%	-	786 - 7.38%	-	749 - 7.03%	-	721 - 6.68%	-
		Base	7411 - 69.58%	-	6601 - 61.98%	-	5498 - 51.53%	-	4457 - 41.29%	-
	MDC	Base + Enh	300 - 5.63%	6 - 0.11%	300 - 5.63%	6 - 0.11%	300 - 5.63%	3 - 0.06%	300 - 5.63%	1 - 0.02%
		Base	2685 - 50.42%	2066 - 38.79%	2169 - 40.73%	2269 - 42.61%	1486 - 27.9%	2471 - 46.40%	960 - 18.02%	2166 - 40.67%
50%	SVC	Base + Enh	542 - 7.48%	-	512 - 7.21%	-	499 - 6.89%	-	472 - 6.64%	-
		Base	57.11 - 78.85%	-	5014 - 70.61%	-	4242 - 58.57%	-	3451 - 48.60%	-
	MDC	Base + Enh	204 - 5.63%	5 - 0.14%	200 - 5.63%	5 - 0.14%	204 - 5.63%	2 - 0.06%	200 - 5.63%	2 - 0.06%
		Base	1984 - 54.79%	1317 - 36.37%	1815 - 51.12%	1335 - 37.60%	1309 - 36.15%	1549 - 42.77%	817 - 23.01%	1526 - 42.98%

Table 4-15 SVC vs. MDC Comparison for *Soccer* in Pull-based P2P Streaming for Different Combinations of RTT and Peer Exit Rate when Average Upload is 1.9 Mbps

			RTT (ms)							
			50		100		200		400	
~ Peer Exit Rate			Full	Half	Full	Half	Full	Half	Full	Half
1%	SVC	Base + Enh	9954 - 79.8%	-	7804 - 55.51%	-	4933 - 35.09%	-	1135 - 8.07%	-
		Base	14058 - 100%	-	14058 - 100%	-	14058 - 100%	-	9891 - 70.35%	-
	MDC	Base + Enh	2344 - 33.35%	3006 - 42.77%	1539 - 21.9%	2996 - 42.62%	766 - 10.9%	2378 - 33.83%	405 - 5.76%	635 - 9.03%
		Base	7029 - 99.86%	10 - 0.14%	7016 - 99.82%	13 - 0.18%	6992 - 99.48%	35 - 0.50%	6842 - 97.34%	171 - 2.43%
5%	SVC	Base + Enh	9634 - 71.41%	-	7532 - 55.83%	-	4655 - 34.50%	-	1136 - 8.42%	-
		Base	13490 - 100%	-	13490 - 100%	-	13486 - 99.97%	-	9394 - 69.63%	-
	MDC	Base + Enh	2004 - 29.71%	2878 - 42.66%	1538 - 22.80%	2887 - 42.80%	766 - 11.36%	2244 - 33.27%	395 - 5.85%	670 - 9.93%
		Base	6733 - 99.82%	12 - 0.18%	6732 - 99.81%	13 - 0.19%	6713 - 99.53%	32 - 0.47%	6568 - 97.38%	158 - 2.34%
10%	SVC	Base + Enh	9113 - 71.3%	-	7238 - 56.63%	-	4493 - 35.15%	-	1048 - 8.2%	-
		Base	12780 - 100%	-	12780 - 100%	-	12780 - 100%	-	9268 - 72.51%	-
	MDC	Base + Enh	2020 - 31.26%	2775 - 42.95%	1526 - 23.61%	2778 - 43%	728 - 11.27%	2208 - 34.17%	394 - 6.17%	747 - 11.69%
		Base	6453 - 99.87%	8 - 0.12%	6447 - 99.78%	14 - 0.22%	6430 - 99.52%	31 - 0.48%	6228 - 97.46%	146 - 2.28%
25%	SVC	Base + Enh	8085 - 74.91%	-	6566 - 60.84%	-	4381 - 40.59%	-	892 - 8.26%	-
		Base	10792 - 100%	-	10792 - 100%	-	10792 - 100%	-	8236 - 76.31%	-
	MDC	Base + Enh	1844 - 33.73%	2291 - 41.91%	1295 - 23.69%	2235 - 40.88%	753 - 13.77%	1853 - 33.89%	389 - 7.12%	930 - 17.01%
		Base	5462 - 99.91%	5 - 0.09%	5453 - 99.74%	14 - 0.26%	5448 - 99.65%	19 - 0.35%	5377 - 98.35%	82 - 1.50%
50%	SVC	Base + Enh	6237 - 82.87%	-	4082 - 55.28%	-	3623 - 49.06%	-	630 - 8.69%	-
		Base	7384 - 100%	-	7384 - 100%	-	7384 - 100%	-	6148 - 84.89%	-
	MDC	Base + Enh	1638 - 42.72%	1455 - 37.95%	1211 - 34.11%	1288 - 36.28%	711 - 18.89%	1505 - 39.99%	349 - 9.27%	866 - 23.01%
		Base	3830 - 99.9%	4 - 0.1%	3830 - 99.9%	4 - 0.1%	3742 - 99.44%	21 - 0.56%	3686 - 97.95%	67 - 1.78%

Table 4-16 SVC vs. MDC Comparison for *Soccer* in Pull-based P2P Streaming for Different Combinations of RTT and Peer Exit Rate when Average Upload is 2.3 Mbps

			RTT (ms)							
			50		100		200		400	
			Full	Half	Full	Half	Full	Half	Full	Half
~ Peer Exit Rate 1%	SVC	Base + Enh	14056 - 99.98%	-	11837 - 84.20%	-	8065 - 57.36%	-	2756 - 19.60%	-
		Base	14058 - 100%	-	14058 - 100%	-	14058 - 100%	-	13988 - 99.50%	-
	MDC	Base + Enh	5808 - 82.62%	772 - 10.98%	3930 - 55.91%	2267 - 32.25%	1923 - 27.35%	3038 - 43.22%	691 - 9.83%	2000 - 28.45%
		Base	7029 - 100%	0 - 0%	7024 - 99.93%	5 - 0.07%	7013 - 99.77%	16 - 0.23%	6963 - 99.06%	62 - 0.88%
5%	SVC	Base + Enh	13632 - 100%	-	11493 - 85.19%	-	7756 - 57.49%	-	2777 - 20.58%	-
		Base	13632 - 100%	-	13490 - 100%	-	13490 - 100%	-	13440 - 99.62%	-
	MDC	Base + Enh	5282 - 78.30%	1013 - 15.01%	3783 - 56.08%	2145 - 31.80%	1865 - 27.65%	3028 - 44.89%	685 - 10.15%	1945 - 28.83%
		Base	6672 - 98.92	73 - 1.08%	6743 - 99.97%	2 - 0.03%	6734 - 99.84%	11 - 0.16%	6669 - 98.87%	73 - 1.08%
10%	SVC	Base + Enh	13064 - 100%	-	11490 - 89.90%	-	7336 - 57.40%	-	2775 - 21.71%	-
		Base	13064 - 100%	-	12780 - 100%	-	12780 - 100%	-	12742 - 99.70%	-
	MDC	Base + Enh	5202 - 80.51%	838 - 12.97%	3588 - 55.53%	2018 - 31.23%	1661 - 25.99%	2733 - 42.77%	671 - 10.5%	1816 - 28.41%
		Base	6459 - 99.97%	2 - 0.03%	6458 - 99.95%	3 - 0.05%	6315 - 98.83%	75 - 1.17%	6321 - 98.92%	62 - 0.97%
25%	SVC	Base + Enh	11502 - 100%	-	10013 - 91.57%	-	7085 - 65.65%	-	2843 - 26.34%	-
		Base	11502 - 100%	-	10934 - 100%	-	10792 - 100%	-	10761 - 99.71%	-
	MDC	Base + Enh	4408 - 80.62%	726 - 13.28%	3594 - 65.74%	1334 - 24.40%	1781 - 32.58%	2316 - 42.36%	615 - 11.24%	1744 - 31.90%
		Base	5419 - 99.12%	48 - 0.88%	5466 - 99.98%	1 - 0.02%	5457 - 99.82%	10 - 0.18%	5401 - 98.79%	57 - 1.04%
50%	SVC	Base + Enh	8662 - 100%	-	7268 - 94.78%	-	5598 - 75.81%	-	2449 - 33.81%	-
		Base	8662 - 100%	-	7668 - 100%	-	7384 - 100%	-	7231 - 99.84%	-
	MDC	Base + Enh	3398 - 85.46%	354 - 8.90%	2601 - 67.84%	867 - 22.61%	1571 - 40.98	1427 - 37.21%	739 - 19.64%	1316 - 34.97%
		Base	3974 -99.95%	2 - 0.05%	3830 - 99.9%	4 - 0.1%	3830 - 99.9%	4 - 0.1%	3655 - 97.13%	35 - 0.93%

Moreover, each cell contains two values: 1) number of chunks delivered to alive peers and 2) average percentage of the delivered chunks.

In Table 4-14 results when the average upload rate is 0.9 Mbps are presented. We observed that this upload rate is not sufficient to receive base layers. Although the bitrate of base layers in SVC and MDC lower than 0.9 Mbps, peers could not receive all base layers chunks. There exists couple of reasons for this situation. First, the effective upload capacities are under the set value 0.9 Mbps due to RTT. Second, peer exits and busy reply messages of requested chunks linearly multiplies the effect of RTT. Third and most importantly, the multi-hopping effect degrades the availability of chunks for a time instant. Consequently, available upload capacity cannot be utilized completely. Increasing the peer exit rate and having constant RTT value (going downwards within a table), the chunk delivery ratio is increasing. At first glance, it seems irrational. However, we observed that as peer exit rate increases the chunk delivery ratio through main seed server increases approximately 10% even though the total number of chunks delivered decreases, which is not favorable. Therefore, the server accessibility of a peer increases. Comparing performances of SVC and MDC for a macro-cell representing a single experiment, we observed that higher number of base chunks allocating higher time duration is received with MDC, however some base chunks are motion compensated temporally interpolated chunks with lower PSNR. For instance, when peer exit rate is ~10% and RTT is 100 ms, 59.09% of base chunks are delivered if SVC is chosen, while 36.65% of base chunks are delivered for both descriptions and 43.27% delivered for one description. Consequently, by utilizing MDC, 20.83% more time duration is successfully covered. The main fact of this is that low bitrate decodable single description bitstreams can be possible by using MDC, and therefore more peers having less bandwidth capacity can receive them. On the other hand, it is not required to compare the enhancement chunk delivery performances since the bitrate is not sufficient even for base chunks, although there exit some received enhancement chunks mostly due to pre-buffering duration of a streaming session. Increasing the RTT and keeping peer exit rate constant (going left to right within a table), the chunk delivery ratio decreases with both codecs since the effective bandwidth decreases.

In Table 4-15, the average upload capacity is increased to 1.9 Mbps. For SVC, except the cases in which RTT is 400 ms, all base chunks are received and enhancement chunk delivery performance is varied between 83% and 35% due to increasing RTT. Considering the base chunk delivery, by using MDC, whole content duration is covered with MDC base layer chunks, while up to 0.5% of the base chunks, which is insignificant, are received with a single description. Increasing RTT for MDC usage, it is observed that only one description chunk reception ratio increases. When RTT is 400 ms, MDC performs better than SVC by covering longer base chunk duration. Higher durations with enhancement layers can be achieved with MDC and has analogous interpretation with the case of base layer chunks when average upload capacity is 0.9 Mbps. However, the received quality is less than SVC. Same observation applies for Table 4-16 with higher enhancement chunk delivery ratios.

Even increasing the peer exits rate in Table 4-15 and Table 4-16, it is observed that no base chunk is missed due to exiting uploading peer when SVC is utilized, which is not actually designed for packet loss recovery. This is mainly due to peers has capability to understand the peer exit events within a time based on RTT and consequently to re-request a chunk intelligently in receiver-driven P2P solutions. The time duration needed to download the redundant part of MDC can be utilized for recovering a peer exit event. Moreover, when peer exits do not occur, higher quality can be achieved since encoding efficiency of SVC is higher than MDC.

Chapter 5

CONCLUSIONS

In this thesis, we have first presented an adaptive buffer-driven chunk request prioritization method. Secondly, the usage of two layer enhancement vs. higher number of enhancement layers of SVC in pull-based P2P video streaming has been analyzed. Thirdly, the performance of SVC and MDC video coding has been evaluated over both push-based and pull-based torrent-like P2P networks with comprehensive experiments. The main contributions and findings of this thesis can be summarized as follows:

- The proposed adaptive buffer-driven chunk scheduling method fulfills design requirements such as maximizing bandwidth utilization, minimizing bandwidth wasting, ensuring play-out continuity and increasing P2P activity at the same time.
- Most of the studies in literature employs high number of layers for SVC streaming over P2P networks without considering encoding overhead of high number of layers. To this end, we have evaluated usage of two and higher number layering in receiver-driven torrent-like P2P networks. Exploiting coding efficiency and less messaging overhead of two layer SVC, higher buffers is achieved compared to higher number of layered SVC. Consequently, we have concluded that two layer SVC performs better than SVC with higher number layers for both networks utilization and individual peer's viewing experience even though higher number of layers increases the adaptation levels.

- In literature, studies comparing performances of SVC and MDC, which are respectively proposed for better adaptation to bandwidth conditions and packet loss situations, utilizes unrealistic overlays and protocols. To this end, we have evaluated video streaming performances of SVC and SVC-based MDC over sender-driven and receiver-driven torrent-like P2P networks. We have mainly concluded that in push-based P2P solutions peer exits considerably affects the performance; therefore MDC should be preferred when peer exit rate is higher than 5%. On the other hand, in torrent-like pull-based P2P solutions, when using SVC, peer exit event can be recovered without loss of a base layer chunk for reasonable RTT values, even though SVC is not designed for packet loss recovery. Moreover, higher qualities can be achieved with SVC since there is no redundancy introduced in SVC. However, for fairly high RTT values, the effective available bandwidth decreases and MDC performs better since bitstreams with low bitrates is more likely to be received by peers having insufficient bandwidths.

BIBLIOGRAPHY

[1] H. Walk, “Oops Pow Surprise… 24 hours of video all up in your eyes!,” *Broadcasting Ourselves: The Official YouTube Blog*, March 17, 2010, [Online]. Available: http://youtube-global.blogspot.com/2010/03/oops-pow-surprise24-hours-of-video-all.html

[2] “Cisco Visual Networking Index (VNI): Forecast and methodology, 2010-2015” Cisco, June 2, 2010, [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[3] H. Schwarz, D. Marpe, T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[4] V. K. Goyal, “Multiple description coding: Compression meets the network,” *IEEE Signal Processing Magazine*, pp. 74-93, Sep. 2001.

[5] J. H. Jeon, S. C. Son, and J. S. Nam, “Overlay multicast tree recovery scheme using a proactive approach,” *Computer Communications*, vol. 31, pp. 3163-3168, 2008.

[6] M. Fesci, E. T. Tunali, and A. M. Tekalp, “Bandwidth-aware multiple multicast tree formation for P2P scalable video streaming using hierarchical clusters,” in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, Cairo, Egypt, Nov. 2009.

[7] M. Castro, P. Druschel, A. M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, “Split-stream: High-bandwidth content distribution in cooperative environments,” in *Proc. Int. Workshop on Peer-to-Peer Systems (IPTPS)*, Feb. 2003.

[8] P. Baccichet, J. Noh, E. Setton, and B. Girod, “Content-aware P2P video streaming with low latency,” in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, Beijing, China, Jul. 2007.

[9] J. Noh, P. Baccichet, F. Hartung, A. Mavlankar, and B. Girod, “Stanford peer-to-peer multicast (SPPM) overview and recent extensions,” in *Proc. Picture Coding Symposium (PCM)*, invited paper, May 2009.

[10] P. P. Baccichet, T. Schierl, T. Wiegand, and B. Girod, “Low-delay peer-to-peer streaming using scalable video coding,” in *Proc. International Packet Video Workshop*, Nov. 2007.

[11] BitTorrent Protocol Specification, [Online]. Available: http://wiki.theory.org/BitTorrentSpecification

[12] A. Vlavianos, M. Iliofotou, and M. Faloutsos, “BiToS: enhancing BitTorrent for supporting streaming applications,” in *Proc. IEEE Global Internet*, Apr. 2006.

[13] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. van Steen, and H. Sips, “Tribler: a social-based peer-to-peer system,” in *Proc. of IPTPS*, 2006.

[14] E. Setton, P. Baccichet, and B. Girod, “Peer-to-peer live multicast: a video perspective,” *Proc. IEEE,* vol. 96, no. 1, pp. 25-38, Jan. 2008.

[15] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, “Resilient peer-to-peer streaming,” in *Proc. IEEE Int. Conf. on Network Protocols (ICNP)*, 2003.

[16] D. Jurca, J. Chakareski, J. P. Wagner, and P. Frossard, “Enabling adaptive video streaming in P2P systems,” *IEEE Comm. Mag.*, vol. 45, no. 6, pp. 108-114, Jun 2007.

[17] A. Magnetto, R. Gaeta, M. Grangetto, M. Sereno, “TURINstream: a totally push, robust, and efficient P2P video streaming architecture,” *IEEE Transactions on Multimedia*, vol. 12, no. 8, pp. 901-914, 2010.

[18] R. Birke, C. Kiraly, E. Leonardi, M. Mellia, M. Meo, and S. Traverso, “A delay-based aggregate rate control for p2p streaming systems,” *Computer Communications*, 2012, to appear.

[19] T. Bonald, L. Massoulie, F. Mathieu, D. Perino, and A. Twigg, “Epidemic live streaming: Optimal performance trade-offs,” in *Proc. SIGMETRICS*, pp. 325-336, Jun. 2008.

[20] M. Jelasity, A. Montresor, G. P. Jesi, and S. Voulgaris, The Peersim simulator. http://peersim.sourceforge.net

[21] V. S. Frost, and B. Melamed, “Traffic modeling for telecommunication networks,” *IEEE Communications Magazine*, vol. 32, pp. 70-81, Mar 1994.

[22] Z. Wen, N. W. Liu, K. L. Yeung, and Z. B. Lei, “Closest playback-point first: a new peer selection algorithm for p2p vod systems,” in *Proceedings of IEEE GLOBECOM*, Houston, TX, Dec. 2011.

[23] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, and D. Yao, “Optimal peer selection for P2P downloading and streaming,” in *Proc. of IEEE INFOCOM*, March 2005.

[24] Y. Wang, T. Z. J. Fu, and D. M. Chiu, “Analysis of load balancing algorithms in p2p streaming,” in *Proc. of SIAM*, 2008.

[25] X. Zhang, J. Liu, B. Li, and T. S. P. Yum, “Coolstreaming/donet: a data-driven overlay network for efficient live media streaming,” in *Proc. of IEEE* INFOCOM, March 2005.

[26] L. D’Acunto, N. Andrade, J. Pouwelse and H. Sips, “Peer selection strategies for improved QoS in heterogeneous bittorrent-like vod systems,” in *IEEE International Symposium on Multimedia,* 2010.

[27] M. Heffeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, “PROMISE: peer-to-peer media streaming using CollectCast,” in *Proceedings of ACM Multimedia (MM’03)*, Berkeley, CA, Nov. 2003.

[28] C. G. Gurler, S. S. Savas, and A. M. Tekalp, "Variable chunk size and adaptive scheduling window for P2P streaming of scalable video," in *Proc. of IEEE Int. Conf. on Image Process. (ICIP)*, Orlando, Florida, Oct. 2012.

[29] Y. Ding et al. "Peer-to-peer video-on-demand with scalable video coding," *Computer* Communications, 33(14), pp. 1589-1597, September 2010.

[30] M. Eberhard et. al, "Knapsack problem-based piece picking algorithms for layered content in peer-to-peer networks," in *ACM Workshop on Advanced Video Streaming Techniques for P2P Networks and Social Networking*, pp. 71-76, 2010.

[31] M. Eberhard, R. Petrocco, H. Hellwagner, and C. Timmerer, "Comparison of piece-picking algorithms for layered video content in peer-to-peer networks," in *Proc. of IEEE Consumer Communications and Networking Conference (CCNC)* , Las Vegas, Nevada, Jan. 2012.

[32] R. Petrocco, M. Eberhard, J. Pouwelse, and D. Epema, "Deftpack: a robust piece-picking algorithm for scalable video coding in P2P systems," in *Proceedings of the International Symposium on Multimedia*, Dana Point, California, Dec. 2011.

[33] The P2P-Next Project. http://www.p2p-next.org

[34] NextShare [Online]. Available: http://www.livinglab.eu

[35] Z. Liu et al., "LayerP2P: Using layered video chunks in P2P live streaming," *IEEE Transactions on* Multimedia, 11(7), pp. 1340-1352, August, 2009.

[36] R. Rejaie, and A. Ortega, "PALS: peer-to-peer adaptive layered streaming," in *Proceedings of the 13th ACM NOSSDAV*, pp. 162-171, Monterey, CA, June 2003.

[37] T. Stockhammer, M. Walter, and G. Liebl, "Optimized H.264-based bitstream switching for wireless video streaming," in *Proc. of Video Streaming, Multimedia and Expo (ICME)*, pp. 1396-1399, 2005.

[38] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[39] A. Segall, and G. J. Sullivan, "Spatial scalability within the H.264/AVC scalable video coding extension," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1121-1135, Sep. 2007.

[40] H. Kirchhoffer, D. Marpe, H. Schwarz, and T. Wiegand, "A low-complexity approach for increasing the granularity of packet-based fidelity scalability in scalable video coding," in *Proc. of Picture Coding Symposium (PCS)*, Nov. 2007.

[41] N. Ramzan, E. Quacchio, T. Zgaljic, S. Asioli, L. Celetto, E. Izquierdo, and F. Rovati, "Peer-to-peer streaming of scalable video in future Internet applications," *IEEE Communications Magazine*, vol. 49, Issue 3, pp. 128-135, March 2011.

[42] N. Ramzan, S. Wan, and E. Izquierdo, "Joint source-channel coding for wavelet-based scalable video transmission using an adaptive turbo code," *EURASIP J. Image Video Processing*, vol. 2007, 2007.

[43] Capovilla, et al., "An architecture for distributing scalable content over peer-to-peer networks," in *Proc. of Int. Conf. on Advances in Mult. (MMEDIA'10)*, Athens, 2010.

[44] M. Zink, O. Knzel, J. Schmitt, and R. Steinmetz, "Subjective impression of variations in layer encoded videos," in *International Workshop on Quality of Service*, pp. 137-154, 2003.

[45] V. Vaishampayan, N. Sloane, and S. Servetto, "Multiple description vector quantization with lattice codebooks: design and analysis," *IEEE Trans. on Information Theory*, vol. 47, no. 5, pp. 1718-1734, July 2001.

[46] Y. Wang, A. Reibman, M. Orchard, and H. Jafarkhani, "An improvement to multiple description transform coding," *IEEE Trans. on Image Process.*, vol. 50, no. 11, pp. 2843-2854, Nov. 2002.

[47] E. Akyol, A. M. Tekalp, M. R. Civanlar, "A flexible multiple description coding framework for adaptive peer-to-peer video streaming," *IEEE J. Selected Topics in Signal Process.*, vol. 1, no. 2, Aug. 2007.

[48] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *VCIP: Visual Communication and Image Processing*, pp. 392-409, 2001.

[49] T. B. Abanoz, and A. M. Tekalp, "Svc-based scalable multiple description coding and optimization of encoding configuration," *Signal Processing: Image Communications*, vol. 24, no. 9, pp. 691-701, Oct. 2009.

[50] I. Radulovic, P. Frossard, Y. K. Wang, M. Hannuksela, and A. Hallapuro, "Multiple description video coding with H.264/AVC redundant pictures," *IEEE Trans. Circuits Syst. Video Techno.*, vol. 20, no. 1, pp. 144-148, Jan. 2010.

[51] C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, "H.264-based multiple description coding using motion compensated temporal interpolation," in *IEEE MMSP'10*, France, Oct. 2010.

[52] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, vol. 1. Asilomar, CA, pp. 342-346, Oct. 1999.

[53] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri, "Multiple description coding for video using motion compensated prediction," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 837-841, Kobe, Japan, Oct. 1999.

[54] W. Xiang, C. Zhu, C. Siew, Y. Xu, M. Liu, "Forward error correction-based 2-D layered multiple description coding for Internet video streaming and multicast," *Signal Processing: Image Communication*, vol. 19, no. 12, pp. 1730-1738, 2009.

[55] A. R. Reibman, H. Jafarkhani, M. T. Orchard, and Y. Wang, "Performance of multiple description coders on a real channel," in *Proceedings of IEEE International*

Conference on Acoustics Speech and Signal Processing (ICASSP'99), vol. 5, pp. 2415-2418, 1999.

[56] R. Singh, A. Ortega, L. Perret, W. Jiang, "Comparison of multiple description coding and layered coding based on network simulations," in *Proceedings of the SPIE Conference on Visual Communication Image Processing*, 2000.

[57] Y. Wang, S. Panwar, S. Lin, and S. Mao, "Wireless video transport using path diversity: multiple description vs. layered coding," in *Proceedings of IEEE International Conf. on Image Processing (ICIP'02)*, 2002.

[58] Y. Zhou and W. Y. Chan, "Performance comparison of layered coding and multiple description coding in packet networks," in *Proceedings of the IEEE GLOBECOM*, pp. 2155-2159, Dec. 2005.

[59] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," in *Proceedings of ACM*, pp. 422-431, 2003.

[60] C. Lee, J. Kim, Y. Altunbasak, and R. M. Mersereau, "Layered coded vs. multiple description coded video over error-prone networks," *Signal Processing: Image Communication*, May 2003.

[61] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zang, "The feasibility of supporting large-scale live streaming applications with dynamic application end-points," in *Proc. SIGCOMM'04*, Portland, USA, Aug. 2004.

[62] J. A. Pouwelse, J. R. Taal, R. L. Lagendijk, D. H. J. Epema, H.J. Sips, "Real-time video delivery using peer-to-peer bartering networks and multiple description coding," in *Proc. of the IEEE International Conf. on Systems, Man and Cybernetics*, vol. 5, pp. 4599-4605, Oct. 2004.

[63] X. Xu, Y. Wang, S. S. Panwar, and K. W. Ross, "A peer-to-peer video-on-demand system using multiple description coding and server diversity," in *Proc. Int. Conf. of Image Processing (ICIP'04)*, vol.3, pp. 1759-1762, Singapore, Oct. 2004.

[64] M. Zink, and A. Mauthe, "P2P streaming using multiple description coded video," in *Proc. Euromicro Conf.*, pp. 240-247, Rennes, France, Sep. 2004.

[65] A. Magnetto, R. Gaeta, M. Grangetto, M. Sereno, "TURINstream: a totally push, robust, and efficient P2P video streaming architecture," *IEEE Transactions on Multimedia*, vol. 12, no. 8, pp. 1340-1352, 2009.

[66] Z. Liu, Y. Shen, S. Panwar, K. Ross, and Y. Wang, "P2P video live streaming with MDC: providing incentives for redistribution," in *Proc. ICME*, pp. 48-51, Jul. 2007.

[67] N. Magharei, and R. Rejaie, "Prime: Peer-to-peer receiver-driven mesh-based streaming," *IEEE/ACM Trans. Networking*, vol. 17, no. 4, pp. 1052-1065, 2009.

[68] M. Lu, J. Wu, K. Peng, P. Huang, J. Yao, H. Chen, "Design and evaluation of a P2P IPTV system for heterogeneous networks," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1568-1579, 2007.

[69] Y. Xu, C. Zhu, W. Zeng, and X. J. Li, "Multiple description coded video streaming in peer-to-peer networks," *Signal Processing: Image Communication*, Feb. 2012.

[70] J. Taal, and R. Lagendijk, "Fair rate allocation of scalable multiple description video for many clients," in *Proc. Visual Communication Image Process. (VCIP)*, pp. 2172-2183, 2005.

[71] P. A. Chou, H. J. Wang, V. N. Padmanabhan, "Layered multiple description coding," in *Proc. International Workshop on Packet Video*, Nantes, France, 2003.

[72] S. Adedoyin, W. Fernando, H. Karim, C. Hewage, A. Kondoz, "Scalable multiple description coding with side information using motion interpolation," *IEEE Trans. Consumer Electron.*, vol. 54, no. 4, pp. 2045-2052, Nov. 2008.

[73] L. P. Kondi, "A rate-distortion optimal hybrid scalable/multiple description codec," *IEEE Trans. Circ. and Sys. for Video Tech.*, vol. 15, no. 7, pp. 921-927, July 2005.

[74] H. Wang, and A. Ortega, “Robust video communication by combining scalability and multiple description coding techniques,” in *Proc. SPIE Symp. Electronic Imaging*, San Jose, CA, Jan. 2003.

[75] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, “Multiple description video coding for scalable and robust transmission over IP,” *IEEE Trans. Circ. and Sys. for Video Tech.*, vol. 15, no. 3, pp. 321-334, March 2005.

[76] M. Yu, X. Yu, G. Jiang, R. Wang, F. Xiao, and Y. Kim, “New multiple description layered coding method for video communication,” in *Proc. of Int. Conf. on Parallel and Distributed Computing Applications and Tech.*, 2005.

[77] L. Chunyu, T. Tillo, Z. Yao, and J. Byeungwoo, “Multiple description coding for H.264/AVC with redundancy allocation at macro block level,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 589-600, 2011.

[78] J. Cornet, Motion Compensated Temporal Interpolation Software, Available. [Online] http://jcornet.free.fr/linux/yuvmotionfps.html