Efficient Delivery of Scalable Media Streaming over Lossy Networks

Jason Quinlan
BSc

Thesis submitted for the degree of
Doctor of Philosophy

NATIONAL UNIVERSITY OF IRELAND, CORK

School of Computer Science
& Information Technology

November 2014

Head of School: Prof Barry O'Sullivan

Supervisors: Prof Cormac Sreenan
Dr. Ahmed Zahran

Contents

List of Figures

List of Tables

I, Jason Quinlan, certify that this thesis is my own work and has not been submitted for another degree at University College Cork or elsewhere.

Jason Quinlan

Abstract

Recent years have witnessed a rapid growth in the demand for streaming video over the Internet, exposing challenges in coping with heterogeneous device capabilities and varying network throughput. When we couple this rise in streaming with the growing number of portable devices (smart phones, tablets, laptops) we see an ever-increasing demand for high-definition videos online while on the move. Wireless networks are inherently characterised by restricted shared bandwidth and relatively high error loss rates, thus presenting a challenge for the efficient delivery of high quality video. Additionally, mobile devices can support/demand a range of video resolutions and qualities. This demand for mobile streaming highlights the need for adaptive video streaming schemes that can adjust to available bandwidth and heterogeneity, and can provide us with graceful changes in video quality, all while respecting our viewing satisfaction. In this context the use of well-known scalable media streaming techniques, commonly known as scalable coding, is an attractive solution and the focus of this thesis.

In this thesis we investigate the transmission of existing scalable video models over a lossy network and determine how the variation in viewable quality is affected by packet loss. This work focuses on leveraging the benefits of scalable media, while reducing the effects of data loss on achievable video quality. The overall approach is focused on the strategic packetisation of the underlying scalable video and how to best utilise error resiliency to maximise viewable quality. In particular, we examine the manner in which scalable video is packetised for transmission over lossy networks and propose new techniques that reduce the impact of packet loss on scalable video by selectively choosing how to packetise the data and which data to transmit. We also exploit redundancy techniques, such as error resiliency, to enhance the stream quality by ensuring a smooth play-out with fewer changes in achievable video quality.

The contributions of this thesis are in the creation of new segmentation and encapsulation techniques which increase the viewable quality of existing scalable models by fragmenting and re-allocating the video sub-streams based on user requirements, available bandwidth and variations in loss rates. We offer new packetisation techniques which reduce the effects of packet loss on viewable quality by leveraging the increase in the number of frames per group of pictures (GOP) and by providing equality of data in every packet transmitted per GOP. These provide novel mechanisms for packetising and error resiliency, as well as providing new applications for existing techniques such as Interleaving and Priority Encoded Transmission. We also introduce three new scalable coding models, which offer a balance between transmission cost and the consistency of viewable quality.

First and foremost this thesis is dedicated to my wife, Teresa, and my son, Jack, for all I am I owe to them. Teresa you are my life, my light and my love. “mancheeruns”, need I say anymore. Jack you are my faith in humanity reborn and all I have is yours.

This thesis is also dedicated to my family, for their love, their support and for just plain putting up with me for the past forty odd years. I especially mention those that I have lost but who are forever in my thoughts: Edward (Pop), Margaret, Mary (Mother), Neil and Harold. For everyone else, I do not mention you by name, for you know how much you are loved, plus there are too many of you to name.

Finally it is dedicated to a friend who was lost during my journey. Tony O’Donovan taught me that you are not limited by the hand you are dealt and that life it is not meant to be a struggle but it is an opportunity to live, to love and to enjoy what ever comes your way.

“There’s no one I’d rather be than me” - Wreck-it Ralph

Acknowledgements

I am forever indebted to my supervisors, Professor Cormac Sreenan and Doctor Ahmed Zahran. Their guidance, their support and their ongoing belief in my research made it possible for me to achieve all that I could. A simple "thank you" will never be enough.

I wish to thank my MISL colleagues past and present, Tony, Nashid, Lau, Paul, Mary, Lanny, Dapong, Hazzaa, Thuy, Xiuchao, Neil, to name but a few, for their help and support during my journey. Thank you all for making me feel welcome and for being there when I needed a helping hand.

A special thank you is reserved for Ilias Tsompanidis. Every student hits a point early in their research, where they wonder if a PhD is the correct path for them. Without Ilias I would not have found the light at the end of that particular question. He was a friend when I needed a friend the most. Ilias is straightforward, honest and direct, and there are points in your life when that is what you need, and that is what I like the most about him. *I have been and always shall be your friend.*

I would also like to thank everyone who made my journey through UCC so memorable. From catering staff, to support staff, from my fellow undergraduate students (2006 to 2010) to the current undergraduate students, from friends made at conferences to friends made in UCC. You have all helped to create such an amazing and outstanding memory for me. I will care for it always and remember you fondly.

I also want to thank the employees and customers of my electrical company, Quinlan Development. Thank you for your understanding and for bearing in mind my college needs when considering your electrical requirements.

Finally, I would like to acknowledge the support provided by the Science Foundation Ireland (SFI) and by the National Telecommunication Regulation Authority (NTRA) of Egypt.

Chapter 1

Introduction

Thesis Statement: Strategic packetisation and error resiliency can improve the efficiency and quality of scalable media delivery over lossy networks.

Before the advent of the Internet, the option to view our favourite TV show was limited by two factors, namely: *location*, *i.e.* the position of the television, which was predominately in the main living room of our home and *time*, *i.e.* when the TV show was broadcast. The introduction of the Internet and the option to access video online, known as *Internet Video*, have removed both of these limiting factors. Increases in broadband speeds for both cellular and fixed line networks coupled with a surge in portable devices (smart phones, tablets, laptops) have provided Internet video the opportunity to grow at an astonishing rate. According to the 2012 Cisco Visual Networking Index [1], by 2017, 69% of all Internet traffic will be video. Over 14% will be Internet video directly to TV. When P2P file sharing of video is included, then video traffic will range from between 80% to 90% of all Internet traffic.

When we couple this rise in Internet video with the growing number of portable devices we see an ever-increasing demand for high-definition online videos while on the move, *i.e.* provided by wireless networks such as Cellular, Satellite and Wi-Fi. In the 2013 Cisco global data traffic forecast [2], currently over 53% of mobile data is video and this will rise to 69% by 2018. This growth in demand is alarming for mobile network operators, who are already struggling to keep up with data backhaul demands and the management of different wireless networks with diverse data rates. At the same time, users expect to view high quality video on devices with vastly different configurations, ranging from tiny smartphones to large-screen high-definition TVs. Before we examine the strain this demand in transmitted video data will place on existing networks, we first present a brief overview of Internet video.

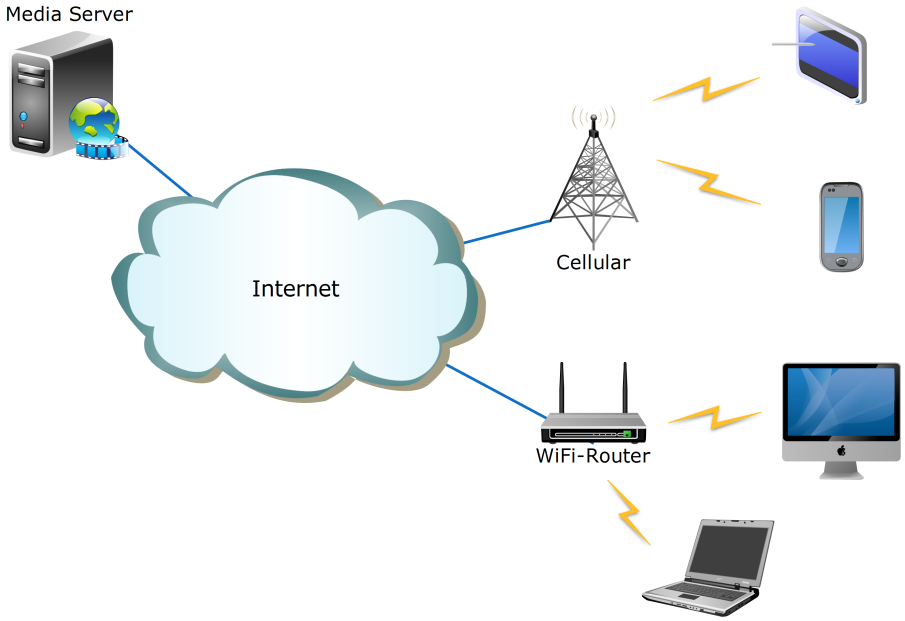


Figure 1.1: Two wireless networks, cellular and Wi-Fi, connected via the Internet to a single media server

1.1 Streaming Video over the Internet

Internet video involves the delivery of a video file from a Server located in one area of the Internet to a Client in a different location on the Internet. Figure 1.1 represents this geographical separation of Server and Client. In this image we illustrate four different devices, of contrasting physical size, which are connected via two wireless technologies, *i.e.* Cellular and Wi-Fi, to a single Server located in the Internet. If we use the example of the laptop; the video file is transmitted from the Server over physical links on the Internet until the data arrive at the Wi-Fi router and are transferred wirelessly to the laptop. Two distinct transmission models are used to transmit the video data from the server to the client and these are "Download" and "Streaming". "Download" requires that all of the file is received at the client prior to reaintuction, or *decoding*, and subsequent viewing of the video contents. Dependent on bandwidth speeds, video file size and losses in the transmission network, "Download" can take considerable time before the video file can be viewed. Thus it is beneficial only when a video file is to be viewed at a later time. "Streaming" on the other hand provides a means of transmitting video which can be viewed nearly immediately upon initial receipt, *i.e.* known as real-time viewing. Hence streaming is widely utilised for sports events and news broadcasts. "Streaming" video data are structured so that once a portion of the data are received at the client, decoding can occur. As long as video data can arrive at the client marginally faster than can be decoded and viewed, then no disruption in viewable quality is perceived. It is important to understand that the path from Server to Client is not influenced by the transmission model, *i.e.* download or streaming.

Traditionally, the structure, or encoding, of the video file is based on three options and these are:

1. *Resolution* - the number of pixels in width and height. A pixel is a single physical point on a screen and is traditionally composed of varying intensities of the colours red, green and blue. A standard HD compatible TV has a resolution of 1920 pixels wide and 1080 pixels high. As the transmission cost of each pixel is equal irrespective of displayed colour, varying the resolution provides a means of adapting transmission cost, also known as varying the bitrate.
2. *Frame rate* - the number of frames per second. Each video is composed of images, and it is the number of images, or frames, displayed per second that provide the illusion of movement. Each frame has a encoding cost and decreasing the number of frames transmitted, reduces the transmission cost.
3. *Fidelity* - the quality level for each frame. Quality can be defined as the *sharpness* of the video. By reducing the sharpness, or blurring, the content of a video frame, the variation in colour between adjacent pixels is reduced. This increases the number of pixels with the same colour intensity and reduces the number of pixels that need to be transmitted.

1.2 Adaptation of Video based on User and Network Constraints

Now that we understand the underlying structure of the video being transmitted over the network, we need to consider why a user may select a specific encoding. The viewable quality of a video is governed by restrictions mandated by both the viewable device itself and the transmission network over which the video is delivered. We begin by considering the constraints imposed by the network.

1. *Network Constraints* - Each link on the network has a limited amount of available bandwidth. Each network is also limited by the underlying technology of the network. Wireless networks tend to have low available bandwidth and a relatively low number of users in comparison to wired physical networks. The level of available bandwidth determines the amount, or quantity, of data that can traverse the specific link on the network. Once the amount of data traversing the network exceeds the level of available bandwidth, data will be lost. In physical networks this loss, called *packet loss*, is typically executed at the routers and switches, where the packets traversing the network are simply dropped. While in wireless networks, the packet loss rate is commonly caused by weak signals, by mobility of the user, and by bit errors, *i.e.* corruption in the data received by the user. Another source of packet loss is bad channel conditions which naturally occur in

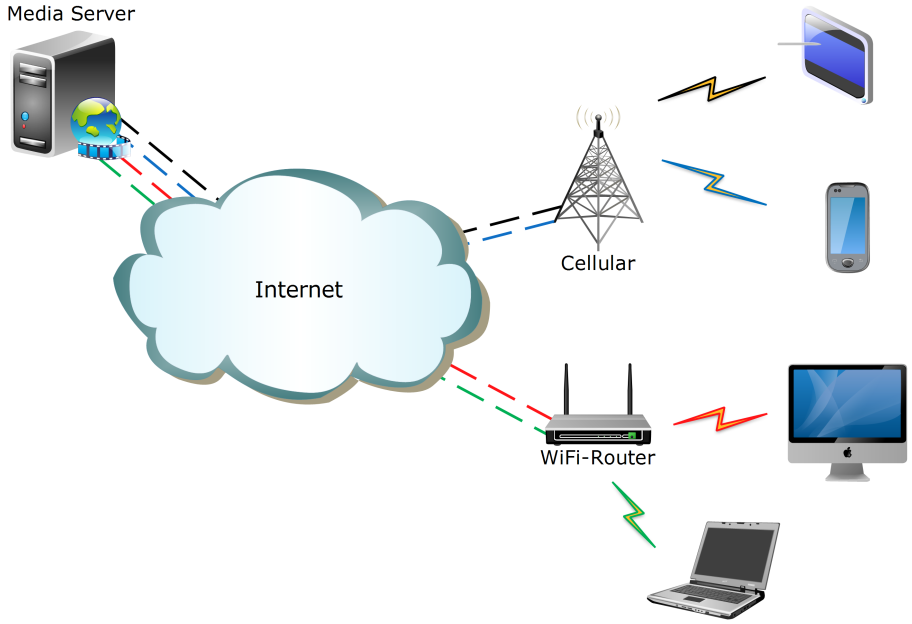


Figure 1.2: Four portable devices streaming video data over two wireless networks, cellular and Wi-Fi, connected via the Internet to a single media server

every wireless channel at a different loss rate, predisposing wireless networks to a higher packet loss rate than physical networks. Typically the viewable quality of a specific encoding of the stream is determined by the level of packet loss incurred during transmission. In general, the level of packet loss tends to vary over the duration of the stream.

2. *Device Limitations* - As seen in Figure 1.1, we show the transmission path from a Server to four different devices, while also illustrating the transmission of a single stream when four devices are viewing a single clip with the same encoding, with each device receiving the same data. In Figure 1.2 we extend this example and illustrate where each device is streaming a different video encoding from the Server. Let us assume that each device is viewing a different video clip and not just a different encoding of the same clip. Each stream is colour coded: black for the tablet, blue for the phone, red for the personal computer and green for the laptop. The selection of video encoding for each device is based on the capabilities, or the limitations, of each device, examples of these constraints are physical size or resolution, computational complexity and battery limitations. By varying the video encoding options, the user can receive the correct video file for their respective device.

   If we now assume that each user is watching the same video, albeit with different encodings, we begin to note why video streaming is consuming so much of the available bandwidth. In our example, four versions of the stream traverse the same link from server to Internet, but more worryingly there is an increase in the data being transmitted over the limited resources of the wireless networks, as

two encodings of the same stream are being transmitted over each of the wireless networks. As the number of users streaming the same data in different encodings increases, this reduces the available bandwidth for other users, which in turn causes higher packet loss rates and reduces the perceived quality of the user.

To counteract the encoding variability that can occur due to device and networks requirements, servers tend to offer a selection of predefined *static* encodings from which to choose, sometimes known as Multi-bitrate streaming or Simulcast [3]. A single encoding may result in unsuitable video quality for some devices or networks. Maintaining multiple copies of the same video, but with different encodings, elevates storage requirements on the Server and increases overall transmission cost. To illustrate the level of encoding variations that can occur due to the number of different devices as well as the difference in transmission networks: for each video available on Netflix there are 120 different encodings available to select [4]. More importantly, this form of *static* streaming, where each stream is a fixed encoding, lacks the flexibility of adapting to changes in network conditions and may result in inefficient usage of network bandwidth as the number of users increase.

To counteract this duplication of data, adaptive, or scalable, modes of video streaming are proposed in the literature. These modes change the streaming encoding characteristics according to changes in the transmission context, *e.g.* device capabilities, service cost, and available resources, or in user requirements. Adaptation is typically based on two options:

i. Re-selection of the stream but at a different quality level. This is commonly achieved by including exchange points within the stream at predefined time periods. At each of these points in the stream, the user can *adapt* the quality of the stream by re-selecting a change in fidelity, frame rate, resolution, or any combination of same, this is commonly know as "progressive download". As only a *segment* of the stream at the required quality level is required, this reduces overall transmission cost as the entire stream does not need to be re-downloaded. An example of this option is Dynamic Adaptive Streaming over HTTP (*DASH*) [5]. Adaptive streaming is susceptible to overhead transmission costs from the client, where each individual segment is requested from the server. Adaptive streaming is also prone to increased storage costs as multiple quality levels of each segment are available. As this form of selective adaptation is user-centric, *i.e.* the quality of the stream is based on the requirements of an individual user, there is no immediate savings in either storage cost or transmission cost, even when multiple users are viewing the same stream. Each of these multiple users may select a different encoding to suit their respective device and network constraints. Only when multiple users are watching the same encoding can there be a reduction in transmission cost.

ii. Selection of a subset of the media data being received, so as to *scale* the stream to suit the required fidelity and network conditions. Generally, a video is identified as a scalable stream when an original high quality version of the video can be encoded into a set of sub-streams such that a combination of one or more of these sub-streams can be used to replay the video at varying quality levels. An example of this option is Scalable Video Coding (*SVC*) [6]. SVC reduces the storage cost as only one encoding is required, albeit at a higher bitrate. But the structure of SVC is detrimentally affected by network loss, such that relatively low level of network loss can cause noticeable deterioration in viewable quality. This form of selective adaptation is also user-centric, *i.e.* the sub-stream(s) selected are based on the requirements of an individual user. The structure of SVC provides immediate reductions in transmission cost when multiple users are viewing the same stream, such as occurs with IPTV [7]. As a single SVC stream is composed of multiple sub-streams, each individual user need only select the sub-streams required to decode their respective viewable quality, thus the sub-streams common to all users are transmitted once, reducing overall transmission cost.

   One additional technology is required to transmit one single stream to multiple clients. Multicast [8] permits a reduction in the number of streams by replicating the stream during transmission at the network routers and switches. Thus only one stream is transmitted over any single link on the network. To determine which stream data belong to which client, clients join Multicast groups and accepted stream data based on the group IP address. Thus sub-streams common to all users can be combined to form a single Multicast group. This approach can cause network loss for some users where the bandwidth available in the network is lower than the transmission rate of the sub-streams common to all users. Which can lead to congestion induced packet loss and subsequent deterioration in viewable quality for some users. In the literature, Receiver-driven Layered Multicast [9, 10] is proposed as a means of reducing this form of loss by permitting each user to specify the level of quality required based on device and network limitation but can lead to an increase in the number of multicast groups required due to the granularity demanded by some users.

Video conferencing is an example where both adaptive and scalable techniques are now being researched as a means to maintain viewable quality for users with diverse requirements. Cisco offers a white paper [11] which provides an overview of the codecs, or encoding processes, currently available for video conferencing.

Next, we consider the motivation for the research undertaken in this thesis. From smart devices to set-top boxes and from YouTube to Netflix, the demand for media is increasing but transmission of media data over lossy networks such as Cellular, Satellite and Wi-Fi can cause intermittent connections, network congestion and mitigate

the achievable viewable quality of the user. We have seen that the transmission networks are limited in available bandwidth, experience data loss and are susceptible to duplication of data over the network. While static streaming provides encodings which better reflect user requirements, this form of streaming increases overall transmission cost and storage requirements. Adaptive streaming provides a fine grained approach to video quality selection, but with higher overhead and storage costs. Multicast provides the transmission mechanism required to reduce overall transmission cost for multiple users viewing the same stream, but quality can deteriorate when limitations in available bandwidth occur. Scalable streaming provides the promise of independent stream qualities with efficient cumulative stream transmission, while minimising storage and transmission costs for multiple users but is sensitive to network loss. The increasing demand for media streaming highlights the need for scalable video streaming schemes that can adjust to available bandwidth, where the transmission network can limit the quality of the video streaming, and can provide the user with graceful changes in video quality as loss rates increase, all while increasing viewing satisfaction.

Finally, we clarify the scenario in which this research can be utilised relative to existing techniques currently deployed in an overall media streaming architecture.

i. *Feedback Mechanisms* - Typically feedback mechanisms are used by existing streaming models, *i.e.* DASH, so as to provide a means of changing the choice of resolution/quality level so as to adapt to variations in the transmission packet loss rate. Our scalable streaming models offer techniques which consider static levels of error correction, irrespective of network loss rates, so as to minimise overall transmission cost, *i.e.* Scalable Description Coding (SDC) and Adaptive Layer Distribution (ALD), while also offering adaptive level of error correction, based on determined underlying network loss rates, so as to provide a balance between overall transmission cost and maximised viewable quality, *i.e.* Streaming Classes (SC).

ii. *Transmission Models* - Unicast and Multicast are normally used for the transmission of data over the Internet. The packetisation techniques we propose, *i.e.* Section-Based Description Packetisation (SDP) and Section Distribution (SD) can be utilised to maximise viewable quality irrespective of the underlying transmission model.

iii. *Deployment* - The optimal environment for using our streaming models and optimisation techniques is primarily pre-transmission. Our segmentation and encapsulation techniques define the structure of the underlying media content post-encoding, while our packetisation techniques define the composition of the packet data prior to transmission.

In contrast, one of our streaming models, *i.e.* Streaming Classes (SC), provides a novel transmission framework which considers all aspects of encoding, pack-

etisation, transmission, decoding and subsequent viewing and provides adaptive mechanisms for determining and supporting consistent high levels of viewable quality for all users.

1.3 Summary of Thesis Contributions

The following is a list of the contributions contained within this Thesis:

i. New segmentation and encapsulation techniques which increase the viewable quality of existing scalable models by fragmenting and re-allocating the video sub-streams based on user requirements, available bandwidth and variations in loss rates, *i.e.* grouping subsets of scalable data and allocating selective levels of error resiliency so as to provide high levels of viewable quality for all users irrespective of user and network issues.

ii. New Packetisation techniques which reduce the effects of packet loss on viewable quality by leveraging the increase in the number of frames per group of pictures (*GOP*) and by providing equality of data in every packet transmitted per GOP, *i.e.* each packet is composed of the same levels of prioritised video data.

iii. New scalable streaming models which increase the consistency of viewable quality over time while minimising the level of error correction required to recover from varying levels of network loss.

1.4 Thesis Structure

The remainder of the Thesis is organised as follows.

- Chapter 2, "Background and Related Work", presents relevant background and related work, as well as an overview of the topics considered and the goals of this work. [1]

- Chapter 3, "Scalable Description Coding (SDC)", we describe our initial approach to increasing quality in scalable video by introducing scalability to description-based streaming. This chapter is based on SDC: Scalable Description Coding for Adaptive Streaming Media [14] which was presented at the 19th IEEE International Packet Video Workshop (*PV 2012*), Munich, Germany in May 2012.

[1] An article was published in a (non peer-reviewed) publication that was written for "outreach" purposes. The article was entitled "TV on the move: How the growth in Internet streaming influences the video quality on your mobile device"[12] and was published in the Boolean [13] magazine for UCC. The Boolean is an annual collection of short papers in which doctoral students describe their area of research and some of their main findings. These articles are journalistic in nature and are written so as to be accessible to a non-specialist audience.

- Chapter 4, "Adaptive Layer Distribution (ALD)", presents our design to increase viewable quality by increasing the number of descriptions transmitted per GOP while decreasing transmission cost by leveraging equality at the packet level. This chapter is based on ALD: Adaptive Layer Distribution for Scalable Video [15] which was presented at the Multimedia Systems Conference (*MMSys 2013*), Oslo, Norway in February 2013 and on a journal version of ALD: Adaptive Layer Distribution for Scalable Video [16] which has been accepted for published by Multimedia Systems Journal (*MMSJ*).
- Chapter 6, "Subjective Testing", illustrates our subjective testing of SVC, MDC, SDC and ALD and their respective variants proposed in Section 3.3 and Section 4.2.
- Chapter 5, "Streaming Classes (SC)", introduces a scalable framework which increases the viewable quality of scalable and description-based models by selectively allocating data to prioritised classes. A journal version of the contents of this chapter, to be titled "Efficient Delivery of Scalable Video using a Streaming Class Model" is currently under development and will be submitted to Transactions on Multimedia Computing, Communications and Applications (*TOMM*) at a later date. [2]
- Our conclusions are presented in Chapter 7.

[2]The contents of chapters 3 to 5 were used as the basis for a patent application. The patent "Method and System for Scalable Description Coding for Adaptive Streaming of Data" (PCT/EP2013/059686) [17] was submitted in 2013 and is currently under PCT review.

Chapter 2

Background and Related Work

As we have seen in Chapter 1, recent years have featured a dramatic rise in the volume of video streaming traffic over the Internet and mobile networks. This increase contributes to a widely acknowledged bandwidth "crunch" at the network edge and is enabled by new devices that feature a large diversity in their capabilities. However, the increase escalates many transmission issues faced by media streaming applications. The current model of transmitting multiple versions of the same video to different devices is over burdening the transmission network and is causing data to be lost during transmission. The effects of this duplication of data is being viewing in the reduction of achievable quality for each of the received streams. Hence, using adaptive video streaming schemes [18] that can adjust, or scale, the achievable quality of the media stream to the available bandwidth evolves as a crucial need for both transmission networks and streaming applications. Multi-bitrate streaming (*adapting*) and layered coding (*scaling*) are the two primary streaming models that support video scalability. The following provides a brief overview of each technique:

i. *Multi-bitrate (MBR) streaming* is a mechanism by which a media clip is encoded as several streams each with a different bitrate and a distinct quality version of the original media clip, sometimes known as Simulcast. Simulcast permits the Server to simultaneously stream multiple versions, or encodings, of the same video file. Early adoption of this mechanism limited the video choice of the user to only one of the available bit-rates. Subsequent change in quality required user video re-selection. A highly efficient [19] and widely used implementation of this concept is the H.264/MPEG-4 Part 10 or AVC (*Advanced Video Coding*) compression standard [20, 21].

   Recent multi-bitrate technologies such as HTTP streaming enable stream quality adaptation by separating the stream into video sections, commonly known as segments. Each segment corresponds to a URL and selecting the required URL permits the decoder to switch between the different qualities based on a num-

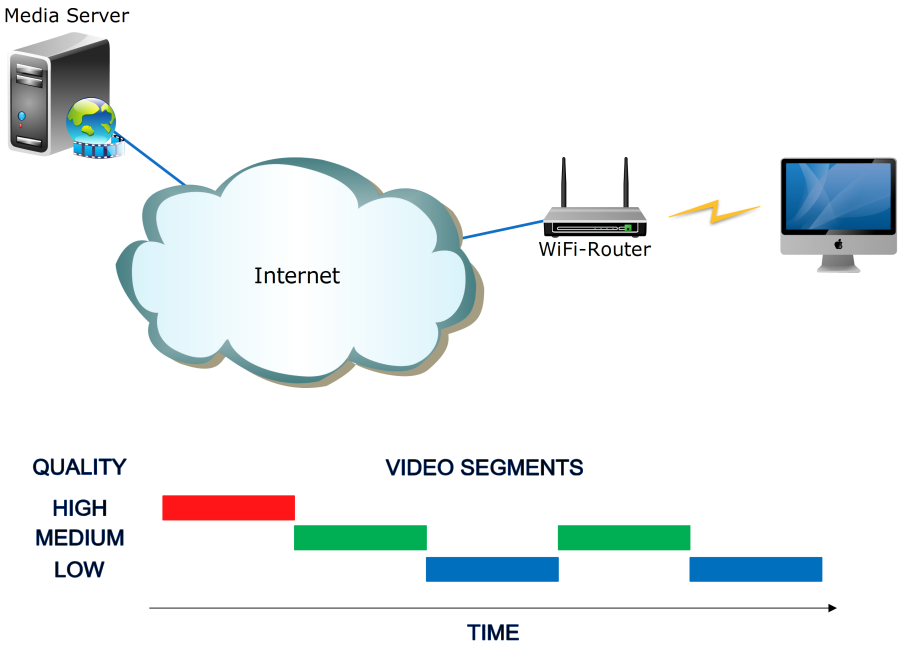


Figure 2.1: The adaptation of quality for a HTTP stream being transmitted to the iMac from Figure 1.2

ber of settings, such as user requirements or network conditions. Examples of this technique include 3GPP Packet-switched Streaming Service (PSS) [22] and Dynamic Adaptive Streaming over HTTP (DASH) [5]. PSS provides for an RTP-based implementation of MBR streaming. RTP (Real-Time Transport Protocol) is a standardised packet format for transmitting video over IP networks. DASH, while a component of the specification of PSS, has developed independently from PSS. DASH provides HTTP-based progressive download and adaptive streaming. DASH alters quality by selecting the segments that best suit device requirements and network conditions. The main limitations of a multi-bitrate scheme include large storage and bandwidth requirements, as multiple encodings of the same video clip are stored and transmitted over the network. Content delivery networks (*CDN*) [23] can be used to reduce the large storage costs of a single server. CDN is a mechanism by which frequently used data can be stored closer to the edge of the network, thus reducing the time taken to transmit data from server to client. DASH is compatible with CDN, due in part to the creation of stream segments, *i.e.* partitioning each of the individual quality streams into smaller portions. CDN utilises DASH to reduce the amount of data being stored and streamed, as only popular segments and specific bit-rates need to be stored close to the edge.

Figure 2.1 illustrates the adaptation of quality for a HTTP stream being transmitted to the personal computer from Figure 1.2. The same colour coding from Figure 1.2 is used to explain the adaptation of stream quality as different video segments are chosen as time progresses. Changes in network loss is an example

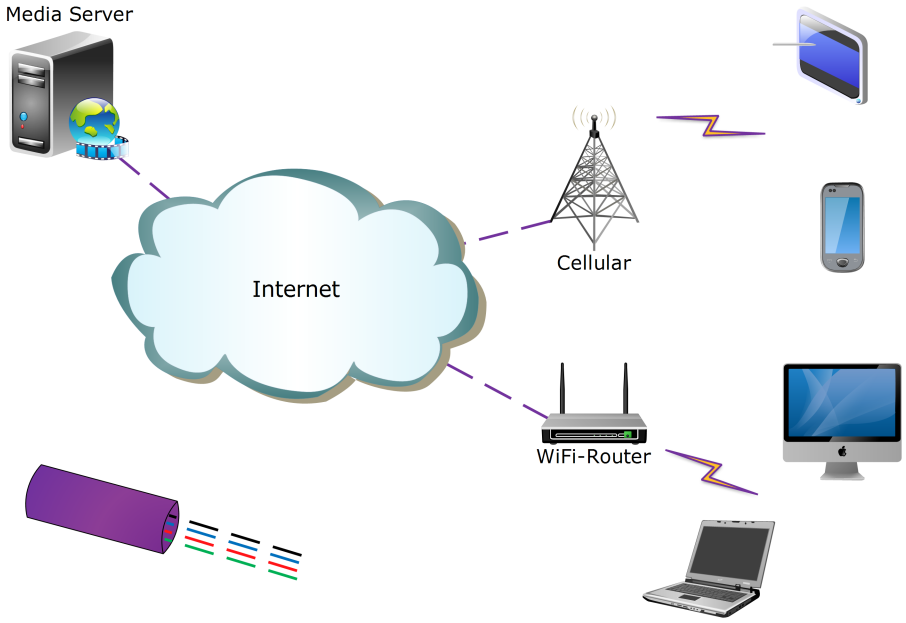


Figure 2.2: Four portable devices streaming a single layered video stream over two wireless networks, cellular and Wi-Fi, connected via the Internet to a single media server. The single purple container illustrates how the layered stream consist of the individual streams depicted in Figure 1.2

of the cause of the variation in viewable quality presents in Figure 2.1.

ii. To counter the increase in bandwidth utilised by MBR streaming, an adaptive streaming technique, commonly known as *layered coding* [24], provides a means of adapting stream quality by adjusting the stream bitrate. Layered coding provides a means of encoding numerous fidelity (quality) levels as one stream. In layered coding, a high quality media clip is fragmented into N layers, which consist of a single base layer and $N - 1$ enhancement, or enhanced, layers. The base layer generally supports coarse minimal quality. The reception of the subsequent enhancement layers increase the viewable quality by providing an increase in temporal, spatial or quality dimensionality. Thus stream quality adaption of layered coding is provided by means of layer selection.

   Figure 2.2 illustrates an example of a single layered stream, composed of four layers, being transmitted to four devices with differing requirements. It can be seen that only one stream is transmitted over the Internet, and in comparison to Figure 1.2 only one stream is transmitted over each of the wireless networks. Thus reducing the number of channels being utilised to transmit the video stream. As illustrated in Figure 2.2 it can be useful to think of layered coding as a single high quality container which consists all of the individual streams depicted in Figure 1.2.

   An example of layered coding is Scalable Video Coding (*SVC*) [6, 25], an extension to the H.264/MPEG-4 Part 10 or AVC (Advanced Video Coding) compression

standard. A major limitation to layered coding is that the technique implements a prioritised encoding hierarchy such that the increase in quality delivered by an enhancement layer is subject to the availability at the decoder of all lower layers that the enhancement layer is dependent upon [26]. In this manner, the loss of a lower layer prohibits the decoding of all higher enhancement layers. More seriously, the loss of the base layer invalidates the video decoding.

A major difference between MBR and Layered Coding is the mechanism for stream adaptation. Both mechanisms communicate with the Server for the initial stream quality level. With MBR a feedback mechanism to the Server is required for selection of the next quality level. The quality of the stream will only adapt once a request for the next segment is sent to the Server and successfully received by the Client. While for Layered Coding the composition of the stream permits adaptation of the stream in real-time at the Client-side. Adaptation of stream quality can be achieved by dropping, or deleting, specific layers within the bitstream.

The remainder of this chapter shall use the encoding extensions of H.264 as a means of explaining the variation that can occur in transmission cost and viewable quality based on the encoding utilised. We shall also illustrate in detail the composition and structure of the various elements and present one additional streaming model: Multiple Description Coding (*MDC*), well-known in the literature, which attempts to reduce the limitation of layered coding [27]. [28] provides a comprehensive performance comparison between layered coding and MDC.

2.1 Advanced Video Coding

The Advanced Video Coding Standard (*H.264/AVC*) [29] is widely used in hardware and software products, such as mobile phones, Blu-ray players, satellite systems, and for videoconferencing. AVC provides the same stream quality for approximately half the data rate of MPEG-2 [29], thus effectively doubling the coding efficiency. AVC only standardises the bit-stream format and the central decoding process, *i.e.* converting the encoded syntax into a viewable stream. All other options, such as transmission, error loss concealment, encoding algorithms, are vendor specific; thus maximising industry deployment of the AVC standard by permitting manufacturers to package AVC to suit their needs.

The bitstream organisation of an AVC stream is provided by encapsulating the bitstream components into Network Abstraction Layer (*NAL*) units and by using a Video Coding Layer (Video Coding Layer) to carry data associated decoding information, such as temporal and spatial prediction. The NAL was created to provide

network-friendly encapsulation and provides this by including header information which can be used by packet and bitstream based transmission.

By gathering a number of continuous frames in to a collection known as a group of frames (*GOF*) or group of pictures (*GOP*), AVC provides an efficient mechanism for creating frame interdependency based on intra- and inter-frames, such as I, P and B frames. Intra-frames (I frames) are fixed points in the stream, and are independent of other frames, while inter-frames (P and B frames) provide a means of bitrate reduction by relying on adjacent frames for supplementary data prior to decoding. Accordingly, the transmission cost of a stream with a higher GOP rate, *i.e.* number of frames per GOP, will be smaller than an equivalent stream with a lower GOP rate.

Looking to the near future the High Efficiency Video Coding (*HEVC*) [30, 31] (H.265) standard is expected to yield an increase in the number of channels of HD video content for a selected quality level while halving the transmission bit-rate [32]. HEVC will lead to more efficient encoding/decoding mechanisms and further enabling the transmission of high quality media streaming over constrained networks.

2.2 Layered Coding

The potential benefits of layered media streaming techniques are apparent by permitting the adaptation of video to match the device resolution and available network resources, without significantly reducing user Quality of Perception (QoP) [33]. But as previously mentioned, a major limitation in layered coding is the prioritised encoding hierarchy by which the increase in quality provided by an enhancement layer is subject to the availability of lower layers at the decoder that the enhancement layer is dependent upon. In this manner, the loss of a lower layer prohibits the receipt of a higher enhancement layer. More seriously, the loss of the base layer invalidates video decoding. The limitation is further exacerbated when the individual frame types *i.e.* I, P and B frames of a GOP, mandate inter-frame dependency such that the loss or a low quality decoding value of a frame can further limit the achievable quality of all dependent frames [26], thus mandating low quality decoding.

Figure 2.3 illustrates the frame interdependency of an eight frame GOP encoding. The interdependency of nine frames are shown, as the I-frame from the next GOP is used to reduce the bitrate of a number of B-frames within the current GOP. For layers one and two, the encoding creates P frames for each frame, while for layers three to eight, the encoding implements a *IBBBPBBB* design. The arrows in Figure 2.3 present the frame interdependency, with the arrow point denoting the dependent frame. As can be seen, the loss or a low quality value of an I or P frame will mandate low quality streaming for all frames which are dependent on it. Larger GOP values may contain a similar structure. This frame interdependency makes Layered Coding an

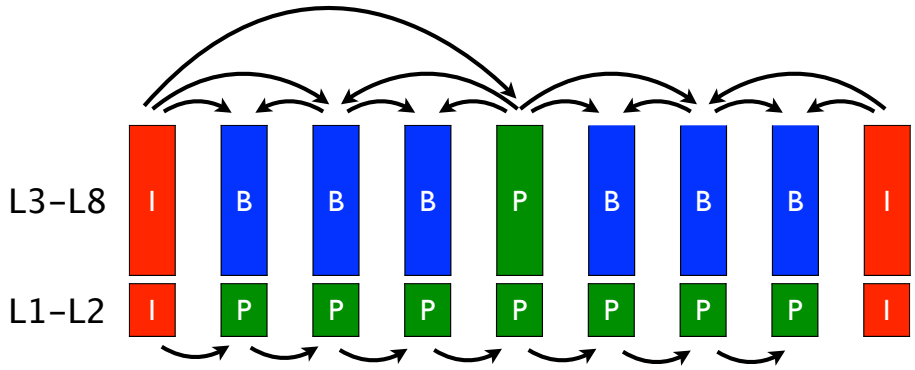


Figure 2.3: An example of the Inter-frame dependency for an eight frame GOP layered encoding. The second I-frame illustrates the inter-dependency of an adjacent GOP.

unattractive approach for links featuring a high error probability such as wireless links, as it necessitates the overhead of retransmission schemes to recover lost packets.

For layered coding, stream scalability can be imposed in three directions: temporal (*resolution*), spatial (*frame*) and quality (*fidelity*) [26]. As such, significant consideration is given to encoding decisions, such as the selection of the scalable encoding scheme, how many distinct resolutions will the stream contain, how may quality layers per resolution, and will the stream contain a variable frame rate. The answer to each of these questions will determine the initial encoding and as such the achievable quality, transmission cost, susceptibility to network loss and the benefits or constrictions imposed on the transmission medium. Common practice would indicate that the diverse requirements of the clients will denote the initial encoding, but there are implementations where the initial encoding may be based on a default encoding selection. These default settings may be used where the media stream is pre-encoded prior to client selection, such as occurs with stored pre-recorded media.

A structure hierarchy of a stream encoded with layered coding is composed of one base layer (BL), layer 1, and numerous enhancement layers (EL), layer 2 to N. In layered coding, the layer index value denotes the achievable quality, such that, relative to the current layer, layers with a smaller index value are seen as lower layers and layers with a larger index value are seen as higher layers. Layered coding implements a dual procedure prioritised encoding hierarchy;

1. It provides an inverse correlation between layer index value and priority *i.e.* the lower the layer index value, the higher the priority.

2. An EL layer of a higher quality may depend on a lower quality EL layer but all EL layers are dependent on the BL.

In this manner, the BL provides a coarse grained version of the stream, while the EL provide for an incremental increase in quality, by utilising the frame data

from the lower layers, thus facilitating datagram re-use and bandwidth reduction. The next section reviews Scalable Video Coding, which is an implementation based on the concepts of layered coding.

2.2.1 Scalable Video Coding (*SVC*)

Scalable Video Coding (*SVC*) is an extension to the H.264/MPEG-4 Part 10 or AVC (Advanced Video Coding) compression standard. The SVC standard [34] provides a mechanism for devices to adjust stream quality by varying the bitrate of the media stream so as to suit network conditions and device requirements. Similar to AVC, SVC only standardises the bit-stream format [35] and the central decoding process, thus permitting transmission protocols to be designed independently, such as a variant of RTP for SVC [36, 37]. [38] provides information on the transport structure of SVC using RTP. Benefits of SVC include permitting devices to utilise pre-buffered lower layer data when requesting an increase in stream quality. An example of this benefit is where the base layer has been received and their exists sufficient bandwidth and time to increase viewable quality by receiving an additional higher layer. Only the additional layer needs to be transmitted and not the base layer, thus minimising the bandwidth transmission cost. Thus providing the benefit of cumulative stream transmission, where different layers can be combined to increase overall viewable quality. This also allows devices with differing stream requirements to selectively choose between the layers on offer, to maximise their respective stream quality without requesting additional data to be transmitted. As well as implementing the key concepts of layered coding, SVC also inherits the GOP functionality of AVC. Looking to the future, a scalable extension to HEVC is proposed [39, 40].

SVC maintains the bitstream organisation introduced in AVC by encapsulating the bitstream components in Network Abstraction Layer (NAL) units, which are organised as Access Units (*AU*). An AU is associated with a single sampling instance in time, commonly referred to as a frame. In a single AU, the SVC bitstream is segregated into slices, which correspond to the distinct quality layers, based on dependency ID (Did), temporal ID (Tid) and quality ID (Qid). A subset of the NAL unit types is a Video Coding Layer (VCL), and contains the coded picture data associated with the source content. Non-VCL units carry data associated decoding information. In this manner, NALUs can be viewed as containing both data or control elements. Within an AU, the VCL(s) associated with a given Did and Qid are referred to as a "layer representation". A Did and Qid with value zero, denotes base layer decoding and is compliant with AVC (note in an SVC stream only the base layer is AVC compliant). Based on Did and Qid, all VCL and non-VCL in an AU are defined as a scalable layer. SNR scalability is based on either Course-Grained Scalability (CGS), which excludes/includes a complete layer when decoding a bitstream or Medium-Grained Scalability (MGS) selectively omitting

NAL units belonging to MGS layers. The selection of NAL units to be omitted can be based on the fixed length fields in the NAL unit headers.

In Section III of [38], the authors provides information on the transport structure of SVC. Figure 2.4 is reproduced from this paper and illustrates an example of three SVC layers being transmitted over three RTP streams, where each RTP stream contains data from different layers or multiple layers, for one or more frames.

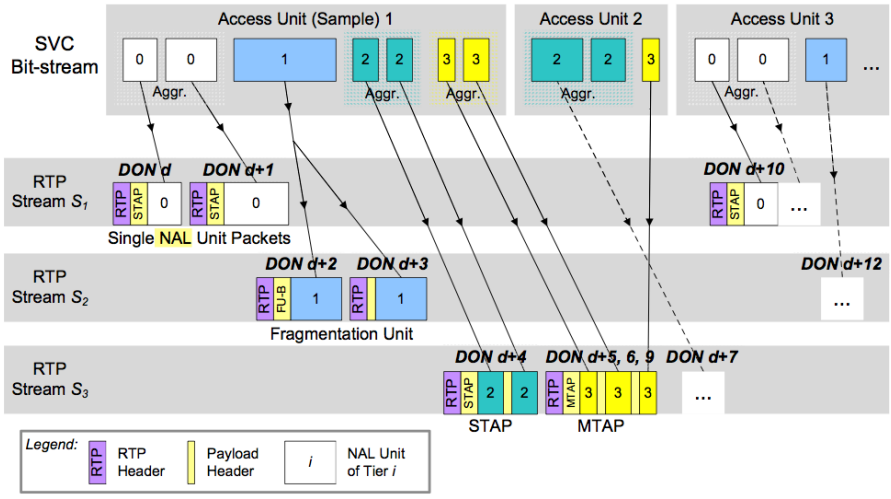


Figure 2.4: An example of three SVC layers being transmitted over three RTP streams

Figure 2.4, reproduced from [38], demonstrates how the SVC data are first defined based on frame (referenced as Access Units (AU)), then on layer allocation, such that NALUs can be created containing a combination of frame/layer data. Each datagram in the RTP stream is composed of a single NALU header and associated SVC data from a single layer but from one or more AU(s). If layer data for an AU is allocated to a single NALU, the NALU header is defined as a "Single Time Aggregation Packet" ($STAP$). If the layer data from a single AU is allocated to two or more NALUs, then a NALU "Fragmented Unit" (FU) header is utilised and finally if a NALU contains layer data from numerous AUs, then a "Multiple Time Aggregation Packet" ($MTAP$) NALU header is defined. These NALU headers are defined to provide information on the structure of each data segment contained within the datagram, so as to allow sub-stream extraction at the receiver.

The goal of SVC is to provide prioritised inter-dependent layers which mandate graceful degradation of viewable quality during periods of network loss, *i.e.* as the percentage loss rate increases, the viewable quality of SVC is incrementally reduced due to the lower available bit-rate. In reality this is not so. SVC is acutely affected by stream quality degradation, as the percentage of datagram loss increases. This is due to the layer dependency inherent in SVC, where the loss of a lower layer aversely affects the decodable quality, as the higher quality layer, which depends upon it, is unable to extract frame data and as such is un-decodable. In a mobile context this is a significant

Table 2.1: Notation

N	The number of SVC layers per Group of Pictures (GOP)
$L_{l,x}$	Transmission cost of SVC Layer l, L_l, for frame x
Section	A segment or a reduced piece of an SVC Layer
$S_{l,x}$	Byte-size of a Layer section of SVC Layer l for frame x
l	Integer value corresponding to the layer number of L_l
GOP	The number of frames per GOP
D_c	A complete description, containing sections from layers 1 to N
q	Number of SVC layers required to decode Layer q

factor that would affect its adoption.

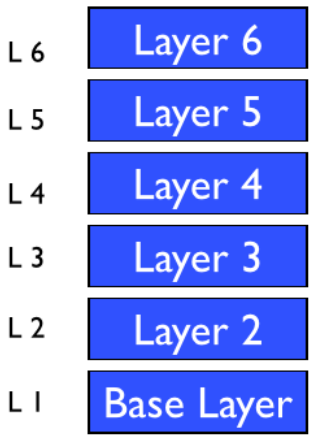


Figure 2.5: An example of a 6 Layered SVC stream

Figure 2.5 highlights a six-layer example of SVC. The interdependency of the individual layers are determined by the selection of stream scalability of the original encoding, *i.e.* temporal, spatial and quality scalability. Typically there are only two dependency scenarios, either:

1. Every layer is dependent on all lower layers, *i.e.* to decode any specific higher layer would require all lower layers. This dependency would occur if only one of the stream scalability options is utilised.

2. Only a subset of lower layers are required to decode a higher layer, as occurs when more than one the stream scalability options is used.

Figure 2.6, reproduced from [41], illustrates how the dependency of each specific layer (presented as a cube) is dependent on a combination of the resolution, frame rate or the quality (not labeled in the z-axis) of the layer. But irrespective of encoding scalability, the base layer, L1, is required to decode all higher layers, and the highest layer index, Layer 6, from our example, requires all lower layers to decode.

If we assume that quality layer q is dependent on all lower layer $1 \ldots q-1$, then the transmission cost for SVC to decode quality layer q can be seen as

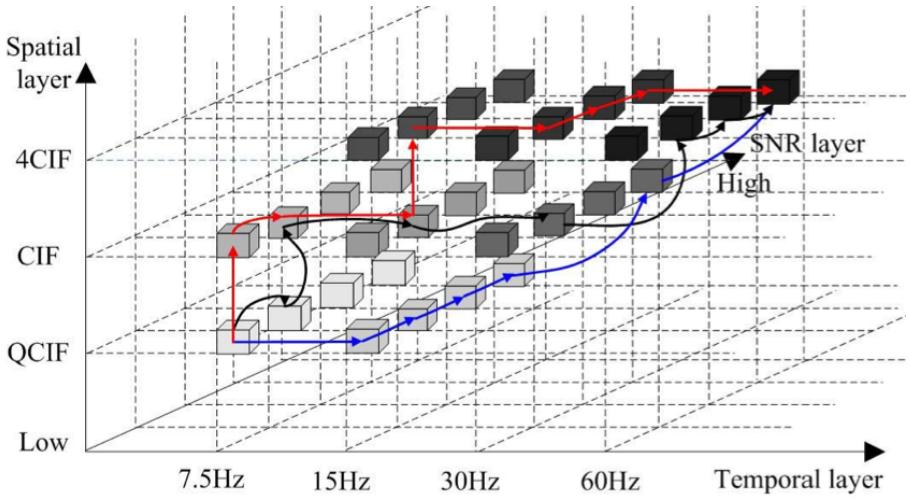


Figure 2.6: 3-dimensional representation of a layered stream composed of temporal, spatial and quality (not labeled in the z-axis) scalability

$$SVC(q) = \sum_{frame=1}^{GOP} \left(\sum_{l=1}^{q} L_{l,frame} \right) \tag{2.1}$$

or using a simple single frame per GOP example

$$SVC(q) = \sum_{l=1}^{q} L_l \tag{2.2}$$

Thus to recap, SVC provides:

1. Prioritised inter-dependent Layers. The viewable quality of higher (enhancement) layers can be dependent on the successful receipt of lower layers.
2. Non-graceful degradation of viewable quality. Dependent on the loss rates for each layer per frame, the variation in viewable quality can be large.
3. Viewable quality not reflective of loss rate. As the viewable quality is dependent on the maximum viewable layer decodable and loss may affect individual layer per frame differently, the probability of viewable quality being reflective of the loss rate is low.
4. A means for efficient bandwidth utilisation in networks. The greatest application of SVC is to reduce transmission cost when heterogeneous devices request live streaming such as for concerts, sporting events and TV.
5. A mechanism for the stream quality of a single user to adapt quickly in the presence of transmission loss, such that the content of the media is consistently

viewable, even though the quality is reduced.

2.3 Error Correction

As we have seen, SVC is acutely affected by stream quality degradation due to packet loss. Prior to introducing the next streaming model, Multiple Description Coding (*MDC*), which specifically focuses on overcoming the impact of packet losses without having to resort to retransmissions. We first consider mechanisms which reduce the effects of packet loss.

Generally, transmission errors encountered by packet data are handled by two mechanisms: Forward Error Correction (*FEC*)) [42] and automatic repeat request (ARQ). We shall introduce FEC later in this section. Transmission control protocol (TCP) is a key transport protocol that implements an ARQ scheme to achieve reliability. In [43], Wang *et al.* reveal that consistent media stream quality requires a TCP throughput twice the average media bit-rate. Additionally, the reliability and flow control mechanisms of TCP can further hinder delay sensitive real-time data [44]. These issues represent serious limiting factors when the user has constrained bandwidth and lossy links, as it is the case for mobile video. Hence, schemes adopting FEC, such as description-based encoding, are a good alternative for media transmission over lossy links or where it is desirable to minimise latency. It is important to highlight that not all of this datagram loss is produced from non delivery. The loss is a mixture of: late packet arrival, bit corruption in the delivered datagrams, out of order delivery and failings in prevalent transmission mechanisms, *i.e.* the TCP protocol is subject to high levels of loss, 21% [45] or more, where re-ordered packets are displaced by more than 2 positions, due to the embedded mechanisms that control the transmission window (*triple-ACK*) [46].

Several concepts have been offered to reduce the level of datagram loss, such as:

1. *Proactive datagram dropping* - Permitting the network or edge routers to determine which datagrams are of least importance, *i.e.* highest layers in adaptive streaming, B frames in media streaming, so as to achieve transmission energy savings and reduce packet delay [47]. The concept assumes that a reduction in data streamed will reduce transmission cost and increase delivery rate.

2. *Prioritising NAL selection in adaptive media streaming* - Sending the optimally selected subset of NAL units, so as to reduce packet transmission size [48], *i.e.* by reducing the quantity of packets being sent to the device. Rather than sending all NALs in a stream, the mechanism selects a subset that is beneficial, but not optimal, for the current streaming devices. Which is similar to layer selection but provides more granularity in requested data.

3. *Signalling in the Network* - [49] proposed that a dedicated network node that contains sufficient knowledge of the stream packetisation mechanism to provide real-time adaptation of the stream in times of network congestion, can reduce transmission cost and subsequent levels of datagram loss. While [50] proposes an adaptive delivery mechanism based on radio resource measurements in 802.11 wireless networks which reduces loss rates and increase user perceived quality.

4. *Unequal Error Protection (*UEP*)* - Contrasting to the previous mechanisms, some UEP concepts add additional data to the stream, so as to reduce the datagram loss at the device, these include:

   A. *Approximate Communication* - [51] offers a mechanism that exploits the problems inherent with corrupted data. It formulates that when a data symbol is received, it is still a good "approximation" of the original symbol, such that by selectively altering the positions of the most significant bit (*MSB*) and least significant bit (*LSB*) to more protected positions, the confidence in accurately decoding the symbol increases.

   B. *Partial packet recovery* - [52] offers a concept that attempts to reduce the quantity of retransmissions due to datagram corruption, also known as bit-errors, by only transmitting the portion of the packet that is corrupt. Partial packet recovery incorporates an expanded physical layer interface so as to increase confidence in determining the correct portions to request and a post-amble, located at the end of the datagram, which replicates the datagram preamble, so to be able to recover from transmission corruption in the preamble.

   C. *Forward Error Correction* - (*FEC*) [53] is an example of UEP widely used by the next streaming model we shall introduce, Multiple Description Coding (*MDC*). Full details on FEC shall be provided in the next section.

2.4 Description-based Coding

To overcome the impact of packet losses without having to resort to retransmissions, Multiple Description Coding (*MDC*) [54, 55, 56] has been proposed. The key idea of MDC is introducing redundancy to the transmitted video to compensate for packet losses. MDC partitions the original N SVC layers into M *descriptions* [57], where the receipt of any single description provides a coarse quality representation of the stream, *i.e.* base layer quality. Similar to SVC, all descriptions are required for maximum stream quality. In this regard, MDC provides a high level of consistency to stream quality by providing mechanism which mitigate network transmission issues albeit at a higher transmission cost in comparison to SVC.

Several variations of the MDC concept have been offered in the research literature [58] and four of the pertinent implementations are outlined, which vary regarding their implementation and subsequent benefits to mobile video.

- **Sub-Sample**

  In Sub-Sample MDC, the original SVC stream data are sub-divided into numerous descriptions based on temporal, spatial or quality domain sub-sampling [59], as well as motion vectors [60] and interpolated frames [61]. In this manner, a stream can be divided based on odd/even frame numbers, pixel block splitting (quality or spatial splitting) or similar divisions. Sub-sampling utilises the minimal change of adjacent pixels to improve the estimated error correction for lost descriptions.

- **Quantisation**

  Quantisation is typically based on either scalar quantisation, outlined in this section, or lattice vector quantisation [58]. In a two description scalar quantisation, each sample value x is mapped to a 2 dimensional M^*M pixel allocation matrix, with the distortion, range of values per matrix index, represented by the percentage of empty cells. This provides a mechanism to decrease distortion by increasing the value of M, which increases the number of empty cells, while decreasing the variance of the original sample value. But this mechanism leads to an increase in transmission cost. Each individual description is allocated as either a row or column, with the index value utilised to determine the range of approximate sample value, while the receipt of both descriptions provides for the accurate value selection. In [62], an MDC-based quantisation of AVC is proposed.

- **Transform**

  Transform provides a method of changing non-correlated stream data vectors x_1 and x_2 into correlated vectors y_1 and y_2. These correlated vectors are then distributed as independent descriptions. It is the interrelation between the correlated vectors that provide a means of estimating description loss. In this manner, correlating transforms offer an efficient mechanism for the inclusion of a minor degree of resilience but as the resilience increases due to escalating network loss, the efficiency is quickly lost. [63] is a seminal paper in this field and proposed transforms from the perspective of sub-space mapping.

- **Forward Error Correction**

  Forward Error Correction (*FEC*) provides a level of error resilience proportional to layer priority. The objective of FEC is to increase the amount of transmitted data, so as to strengthen the likelihood that a subset of the data will arrive at the device, thus improving the probability that the original data can be decoded [64]. FEC is achieved by taking k existing data symbols, increasing this to n FEC data

symbols (consisting of existing and redundant data symbols), such that receiving *k+1* FEC data symbols at the device, facilitates the recovery of the original k data symbols, as is constant with the Reed-Solomon block code rate [65].

The version of MDC associated with FEC, creates a description composed of a *section* from each of the N SVC layers. A section is created by dividing a layer based on its priority, typically with the divisor from 1 to N. To provide the incremental increase in viewable quality provided by each additional description, it is typical that M equal N. Thus, the cumulative receipt of additional descriptions increases stream quality proportionally.

We will primarily focus on Forward Error Correction (*FEC*) as it provides a means of dynamic adaptive stream encoding, low computational complexity, the large quantity of descriptions necessary for the substantial numbers of heterogeneous media streaming devices indicated for future deployment and it has attracted considerable attention in the literature [66, 67]. [68] is one of the initial papers offering the benefits of MDC-FEC for resilient adaptive streaming.

The next section gives an in-depth outline of the design and implementation of error correction with specific reference to MDC-FEC.

2.4.1 MDC Design and Structure

Figure 2.7 illustrates how the original six SVC layers from Figure 2.5 are partitioned over six descriptions. As can be seen each original layer, i, is divided by its layer index, i, and distributed over i descriptions. Thus for layer i, L_i, i *sections* are created, *i.e.* for layer 2, L_2, 2 sections are created and distributed over 2 descriptions while for layer 6, L_6, 6 sections are created and distributed over 6 descriptions. Each description is thus composed of layer sections. In this manner, i descriptions are required to decode and view layer i. This can be generalised to:

$$\frac{L_l}{l} \tag{2.3}$$

Once the sections are created, FEC is utilised to extend the layer data over the M descriptions, such that the higher the priority of the layer, the greater the level of error resilience, as illustrated in Figure 2.8. In this manner, MDC-FEC contains an adaptive mechanism for description creation and error resilience allocation, but these mechanisms increase transmission cost proportionally to the level of FEC and is proportionally high compared to the initial level of SVC data, thus leading to a large increase in transmission cost relative to SVC.

As can be seen in Figure 2.8 the original SVC data are shown in blue while the FEC is shown in green. This image illustrates the increase in transmission cost

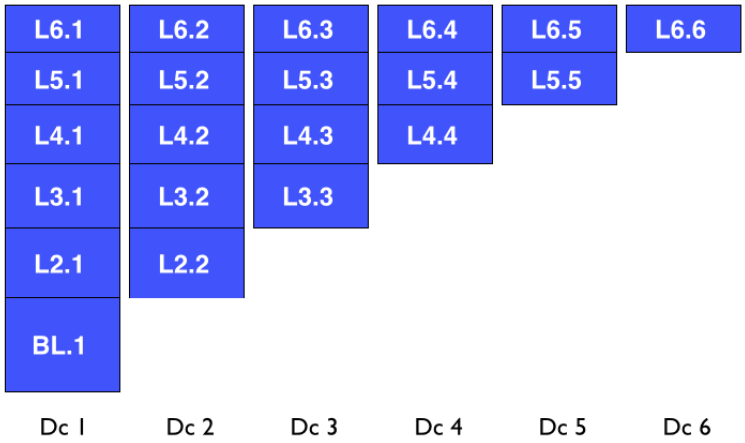


Figure 2.7: A six-layer SVC encoding distributed over six MDC descriptions, prior to FEC allocation

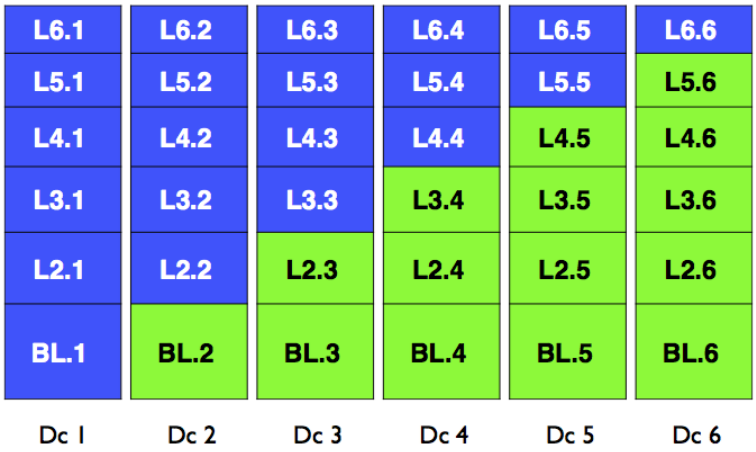


Figure 2.8: An example of a 6 Layered SVC stream encoded as MDC c/w FEC

with respect to FEC, relative to the specific SVC layer, but this image does not show the distribution of the respective SVC and FEC data per layer. Typically FEC can provide either systematic or non-systematic encodings. Systematic schemes encode the original symbols as part of the transmitted stream, while non-systematic schemes encode and transmit the original symbols as new symbols. Raptor codes [69] propose that a systematic encoding, with encoded symbols interspersed among the original symbols, provides a greater level of decodability. Thus in reality each section contains a mixture of SVC and FEC data assuming a systematic scheme, while all data are FEC data in a non-systematic scheme. It is important to keep this thought in mind while reviewing description-based schemes.

A few characteristics of Figure 2.8 are important to note:

1. The equal importance of each description is immediately noticeable, as each description contains one section from each SVC layer.

2. One of the fundamental benefits of FEC is that a minimum level of stream quality

is achievable once at least one description arrives at a device, while an incremental increase in stream quality is provided by the receipt of additional descriptions, but this minimum quality per description also outlines a significant flaw. To achieve a high level of quality, the base layer must be replicated in each description, while the higher enhancement layers contain levels of FEC inversely proportional to their layer index. This leads to large increases in the transmission cost of MDC-FEC.

3. Each MDC-FEC description contains a section from each of the N SVC layers. This is beneficial to devices requiring high quality, as each description provides a piece of the required dependent layers, but devices requiring low or even base quality are mandated to receive higher layer sections which are never utilised.

To extend the section allocation over a number of frames per GOP, we write that an MDC description section from layer l from frame x, $S_{l,x}$, contains $\frac{L_{l,x}}{l}$ of the layer size, while a single *complete* MDC description from frame x, as shown in Equation (2.4), contains the transmission cost of one section from layers 1 to N:

$$\sum_{l=1}^{N} \frac{L_{l,x}}{l} \tag{2.4}$$

While we view the total transmission cost of one complete MDC description, MDC_{D_c}, from each frame per GOP as

$$MDC_{D_c} = \sum_{frame=1}^{GOP} \left(\sum_{l=1}^{N} \frac{L_{l,frame}}{l} \right) \tag{2.5}$$

Note that it is not correct to multiply a single description by the number of frames per GOP, as each frame, as well as each layer, per GOP may have differing transmission cost, hence the requirement of the summation over all frames, using the $frame$ value, and the need to determine the layer cost per description section for each frame. Also note that the number of layers per frame, and number of frame rates per GOP depends on the underlying SVC encoding. In our equations for MDC we determine the total transmission cost based on all layers required at the maximum frame rate. If a reduction in the frame rate is necessary, then a modified version of Equation (2.5) would mandate an additional variable, *frameStep*, which would increment over the frames not required. The following example illustrates a frameStep of 2 which would half the frame rate. Note that the frameStep value is dependent on the governing GOP value, such that the frameStep value can never be larger than the GOP value and that the frameStep value must always be a power of 2:

$$MDC_{D_c} = \sum_{frame=0,frameStep=2}^{GOP-1} \left(\sum_{l=1}^{N} \frac{L_{l,frame=frameStep*frame+1}}{l} \right) \tag{2.6}$$

As can be seen from Figure 2.8, the number of descriptions required to view a select layer can be defined by the layer value, *e.g.* using Equation 2.4 with $N = 3$, the section size of layer three allocated to each description is a third ($\frac{L_{3,x}}{3}$), which mandates three descriptions are required to decode layer 3. Note that while the maximum viewable quality from three descriptions is layer 3, three sections from layers 4 to 6 are also received. Hence the total transmission cost of three descriptions can be defined based on the number of initial SVC layers times the number of sections per layer required to decode the requested quality level. In our example this would be six layers times 3 sections for each layer. Equation 2.5 defines the transmission cost of one complete description, *i.e.* one section from all six layers. We can define the transmission cost to view layer 3 as $MDC_{D_c} * 3$. Thus the total transmission byte cost of MDC per GOP and at the maximum frame rate required to decode quality layer q can be seen as

$$MDC_{D(q)} = MDC_{D_c} * q \tag{2.7}$$

While the total FEC transmission cost overhead for MDC quality layer q can be characterised as $MDC_{D(q)}$ from Equation 2.7 minus $SVC(q)$ from Equation 2.1

$$overhead = MDC_{D(q)} - SVC(q) \tag{2.8}$$

or using a simple single frame per GOP example

$$MDC_{D(q)} - \sum_{l=1}^{q} L_l \tag{2.9}$$

Note that layer l defines a specific layer within the encoding and transmission of SVC, while quality, or layer quality, q defines the viewable quality achievable by decoding a number of descriptions.

Thus to recap, MDC provides:

1. increased transmission cost relative to SVC.
2. increased error resiliency, proportional to the priority of the layer.

3. viewable quality in some part reflective of network loss rate. Viewable quality is dependent on the number of descriptions available for decoding. Thus as loss rates increase, the probability of few descriptions being available for decoding increases.

4. non-prioritised (equally important) descriptions. Each descriptions contains the same amount, *i.e.* number of sections, from each SVC layer.

5. single description provides base layer decoding. Each description contains the base layer, thus minimum quality decoding is available once any description is received at the device.

6. High transmission cost for lower layer decoding. Each description contains one section from each layer and if a low quality layer is preferred, then receipt of sections from higher layers are not required, but are received.

2.5 Metrics Used for Evaluating Video Quality

This section gives an overview of the relevant metrics [70] used for evaluating video quality. Items 1 to 4, PNSR, MSE, SIMM, and QoS provide a quantitive means of determining the same result irrespective of the number of evaluations. While items 5 to 6, MOS and QoE are subjective in nature and depending on the person evaluating the video clip, may produce variations in result during each evaluated.

1. Peak Signal to Noise Ratio (PSNR) - is the simplest and the most widely used video quality evaluation methodology. PSNR [71] is utilised to specify the pixel difference between the transmitted and received video data, on a frame per frame basis (known as full reference), so as to determine a quantifiable value for the variation in viewable quality. In this thesis, we utilise PSNR to evaluate the effects of packet loss on viewable quality.

2. Mean Squared Error (MSE) - MSE [72] is a signal fidelity measurement. MSE compares two signals by providing a quantitive value that describes the degree of similarity between them. Typically MSE is utilised to determine the overall PSNR values for a given set of images.

3. Structural Similarity Index Metric (SSIM) - Similar to PSNR, SSIM is a full reference objective video quality metric [73]. SIMM is proposed as an improvement on the traditional methods, such as PSNR and MSE, as SIMM views image degradation as changes to video quality, while traditional methods evaluate based on pixel similarity and ignore degradation in video quality.

4. Quality of Service (QOS) - Conversely QOS [74] is a measurement of the underlying transmission network on which a given service must traverse. In computer

networks, this result may consider error rates, bandwidth and throughput, while for video quality, this result may consider delay, latency and jitter.

5. Mean Opinion Score (MOS) - In subjective testing, quality can be measured by each viewer giving a score ranging from one (worst) to five (best). MOS [75] is the arithmetic mean of all these individual scores. In our subjective testing in Chapter 6, we utilise MOS, in Section 6.3.1.3, to tabulate the grading and ranking values of our test subjects.

6. Quality of Experience (QOE) - QOE [76] is a measurement of a persons experience with a given service. The results of this measurement tend to be specific to a given person and can vary dependent on the tolerance levels of a person to specific underlying issues. In video QOE determination this may be based on buffering times, startup delay, and quality variation over time.

2.6 Transmission and Optimisation Techniques for Video Delivery

This section gives an overview of existing transmission and optimisation techniques for video delivery. In this section we assume that IP is the underlying network technology for video transport. Before we present the specific video optimisation techniques, we provide an overview of relevant transmission network mechanisms for optimising delivery of transmitted data.

1. Multi-protocol Label Switching (MLS) - MLS [77] is a packet forwarding scheme. In an MLS network, labels are allocated to data packets, so as to define packet forwarding decisions for routing through the network based on the MLS label and not on the packet itself. An example of MLS usage is in the need to reduce the complexity of routing table lookups.

2. Multicast [8] is utilised to reduce the replication of IP traffic over the same path by sending data only once, based on a single IP address, called a Multicast group, and permitting clients with different IP addresses to access the data from the Multicast group. This can reduce overall congestion levels and is typically utilised for IPTV.

3. Differentiated services (Diffserv) - Typically the Internet offers a best-effort service based on a single class of user, *i.e.* the same settings for all users, and thus, treats all packets the same While type of service (TOS) bits are included in the IP header, which can be used to prioritise traffic, they are seldom used. Diffserv [78] is an architecture which offers different levels of service, so as to reduce delay and lower the drop rates for prioritised data. Diffserv normally creates classes of users and shapes overall traffic routing to suit user classifications.

4. Transmission Control Protocol (TCP) - TCP is a widely used transmission protocol which mandates reliable transmission, delivery order and incorporated error-checking. TCP can navigate through fire walls and NAT connections, so is generally used in scenarios where delivery must be guaranteed, *i.e.* web surfing, mail, ssh and FTP.

5. User Datagram Protocol (UDP) - Un-like TCP, UDP is an unreliable transmission protocol used for the delivery of data. UDP has no guarantees of delivery, no ordering and limited error checking. UDP is normally used when the latency, or delay, mandated by reliability is an issue. UDP has no handshaking or setup delays, so is used by time-sensitive application such as by DNS, DHCP, and some audio and video applications.

6. Content Delivery Network (CDN) - While not a specific transmission mechanism, CDNs [79] provide a means of reducing the initial start up delay in receiving data and increasing the speed, *bitrate*, at which data is delivered. CDNs typically transport priority data from the original server to a number of edge servers, which are closest to the relevant users of a specific application. This reduces RTT and enables delivery of data quickly to the user and normally with less transmission issues, *i.e.* reduction in both congestion levels and underlying packet loss rate.

The following video techniques optimise delivery, especially when issues occur during transmission as can materialise during congestion and subsequent moments of errors/losses.

1. Real-time Transport Protocol/Real-Time Control Protocol (RTP/RTCP) [80] - RTP provides an end-to-end delivery service for real-time applications, such as video delivery. RTP commonly uses Multicast and UDP as the preferred delivery methods. RTP does not guarantee timely delivery or include quality of service guarantees. It requires lower-layer services, such as RTCP to support these requirements. RTP introduces an additional header space, by which new fields can be used to improve stream quality.

   RTCP is commonly used by RTP to provide control feedback from client to server on the data that has been delivered and the variation in quality caused by loss in the network. RTCP is typically used for QoS monitoring and congestion control, while mechanisms for the synchronisation of received data (lip syncing) and balancing control traffic can also be utilised.

2. MPEG Transport Stream (MPEG-TS) - The MPEG-TS [81] defines how the MPEG stream is delivered or broadcast over the network and not how it is encoded or stored on your machine, *i.e.* the quality of the clip or the file format. The MPEG-TS can contain one or more content channels, but in an IP network, it is more useful and uses less bandwidth when each channel has its own multicast

address. MPEG-TS tends to use smaller byte-sized packets and can contain error correction mechanisms which reduces the effects of packet loss.

3. Digital Video Broadcasting (DVB) - DVB [82] is a group of international standard used for the delivery of audio and video data. As each of the underlying transmission networks, or user requirements, tend to differ depending on the broadcasting or receiving technology, DVB has developed solutions for a number of operators. These include: DVB-T, a solution for terrestrial broadcasting, DVB-H, a system delivering content to battery-powered devices, typically mobile devices, DVB-S2, the next generation satellite system, DVB-IP, a solution for the deliver of broadcast content over IP networks and Multimedia Home Platform (DVB-MHP), which enables interactive TV applications to be delivered over the broadcast channel.

4. SMPTE - SMPTE [83] is a global standard used by most media broadcasters to provide inter-operability between transmitted data and equipment from multiple manufacturers. SMPTE can be used to improve video delivery by augmenting the standard error correction mechanism contained within the transmission protocol with both row and column error protection supplemented with FEC. The video packets are grouped into rows and columns. One additional FEC packet is appended to each row and column, so that the loss of a packet within a row/column can be offset by the contents of the FEC packet. SMPTE can also be used to timecode individual frames, so as to improve the synchronisation of decoded audio and video signals during periods of packet loss.

5. Network Abstraction Layer (NAL) - As defined in Section 2.1 (AVC) and Section 2.2.1 (SVC), the NAL [84] was created to deliver network-friendly encapsulation and provides this by including header information which can be used by packet and bitstream based transmission. The NAL allows greater customisation of the video content to the transport layer.

6. HTTP Streaming - Commonly known as Adaptive Streaming over HTTP [85] utilises TCP and port 80 to bypass firewalls, contains in-built flow control to ease congestion and can adapt the underlying bitrate to suit network conditions. While the reliability built-in to TCP can mandate increases in overall bitrate, the adaptation in viewable quality can increase user QoE.

2.7 Improving Streaming Performance of Scalable Video

As our work is predominately based on re-allocation of layered data in both SVC and MDC, our research is primarily focused on this area of the literature.

2.7.1 Literature based on SVC

- As previously illustrated SVC is composed of dimensionalised layered data, such that for each given resolution, numerous frame rates and fidelity levels are available. Thus there exists a subset of layers, or NALs, which if lost are detrimental to a large number of higher layers, examples include the base layer and the lowest layer for each distinct resolution and frame rate. [41] proposes an adaptive layer-based ordering algorithm by which stream quality can be increased, while [86] prioritise based on macro block ordering, *i.e.* pixel pattern selection.

  Unlike the authors of [41, 86] our work is not selective in the choice of which subset of SVC data to transmit. SVC can be viewed as benefiting a single user by providing graceful degradation of quality while also reducing transmission cost when multiple users are viewing the same stream. For a single user if a specific viewable quality is optimised by transmitting only a selection of NALU, this may cause limitations in graceful degradation when network loss occurs. Also, in a large real-world deployment, the overall quality of a scalable IPTV stream will be governed by the maximum viewable quality selected. Thus a subset of NALU may not be sufficient for all users. Hence we focus on maintaining high levels of viewable quality for all users.

- The integration of the layered structure of SVC with adaptive HTTP streaming has also been researched. The authors of [87] investigated the merging of SVC with the MPEG-DASH standard [88] in mobile environments. While the authors of [89, 90] compare AVC DASH to SVC DASH, with respect to encoding complexity, storage requirements and service cost. The authors of [90] also consider HTTP caching within the network, rate adaptation and considers live streaming. The authors of [91] explore the integration of SVC and DASH with respect to the efficiency of network caches and the congestion bottlenecks that can occur for both cache feeder links and for access links. While the authors of [92] study an adaptive SVC-DASH client over multiple dynamic network connections.

  DASH in some regards is related to our research but DASH does not address issues of operating over unreliable channels. As previously stated, the reliability and flow control mechanisms of TCP can hinder delay sensitive real-time data, especially over constrained or lossy networks. While schemes adopting FEC, such as description-based encoding, are a good alternative for media transmission over lossy links. Thus we focus on adaptive streaming techniques which can leverage FEC, such as SVC or MDC.

- As SVC encoding is computationally expensive, several works in the literature present models which utilise the multilayer coding aspect of the spatial scalability to increase coding efficiency. The authors of [93] propose a motion estimation

scheme to lower complexity by reducing the search range of enhancement layers. The authors of [94] propose a modified content-adaptive spatial scalability SVC coder (*CASS-SVC*) which mandates additional side information for the decoder but reduces critical information loss during spatial scaling. While the authors of [95] examine the up-sampling process required to decode higher layers in the scalable extension of H.265 and proposes a design that is content adaptive and is motivated by compression noise in the base layer.

As distinct from the motion estimation scheme [93], CASS-SVC [94] and the content adaptive upsampling scheme [95], we do not alter the SVC encoding process but simply propose techniques which are resilient to network loss so as to maximise viewable quality, irrespective of the original SVC encoding. On a side note, our proposed techniques can be leveraged by any streaming model which utilises a layered hierarchy and this includes the work proposed by [93, 94, 95].

- The onset of study into Software Defined Networks (*SDN*) and Openflow-assisted QoE provides researchers with a new delivery technology with which to increase viewable quality. The work proposed by [96, 97] are examples of this field of study.

  Our research does not leverage in-network optimisation techniques or feedback mechanisms but future work will investigate the benefits of adapting FEC levels to suit reactive network feedback. As our work is description-based it is uncomplicated for an in-network transmission mechanism to selectively drop packets or entire descriptions used by our techniques to suit network and link requirements, with limited effects on viewable quality.

- Research is also being undertaken into protecting "regions of interest" within the data stream with higher levels of protection, *i.e.* protecting locations within a frame where the action is taking place, such that other regions contain less interesting data and their loss is less noticeable. Focusing on watermarking specific regions [98] and creating distinct enhancements layer for specific regions of interests [99] are examples of this research.

  Unlike the work proposed by [98, 99], our work does not highlight important areas within a frame or alter the SVC layered structure of individual frames within the encoded stream. Our research is concerned with adapting to loss irrespective of the original SVC encoding.

2.7.2 Literature based on MDC

Several mechanisms have been proposed to vary the transmission cost of MDC while maintaining achievable quality from MDC description allocation. These include:

- Adjusting the levels of FEC such as Adaptive FEC [100] and Enhanced Adaptive FEC [101]; The options here include:

  1. Removing the incremental FEC per layer thus providing the same level, or number, of FEC sections for different layers. This option mitigates incremental variation in achievable quality as loss rates fluctuate.

  2. Adjusting the byte allocation per FEC section thus maintaining the incremental FEC allocation per layer.

  Our techniques do not use Adaptive FEC or Enhanced Adaptive FEC specifically but we do adapt the FEC levels within the transmitted stream to suit our specific needs with reference to maintaining high levels of viewable quality and varying the degree of FEC to suit determined levels of network loss. Thus, some of our FEC-allocation research could be classified as a variation to Adaptive FEC or Enhanced Adaptive FEC, but no prior work has proposed the FEC adaptation as suggested by our research.

- Optimising FEC resilience, by determining FEC based on "layer intra-dependence". Such that FEC on layer k shall be composed from both SVC data on layer k as well as SVC and FEC data from a subset of layers 1 to $k-1$ dependent on layer intra-dependence [102]

  As distinct from [102] our work focuses on adapting the FEC level for distinct layers as apposed to collecting lower layers, and their respective FEC, together to create new FEC levels for enhanced layers. In our research we do investigate the benefits of including individual lower layer FEC data within the packetisation mechanism of higher layers, so as to maximise viewable quality for devices requiring lower layer video quality without the need for the retransmission of lost data.

- Modifying the layer allocation per MDC description, such as transmitting the base layer as a separate MDC description [103, 104]; The base layer and associated error resilience are transmitted separately with devices demanding higher quality requiring both base and enhanced streams. The authors of[103] base their work on Priority Encoding Transmission (PET) [105], a prioritised packetisation scheme.

  While our work does not create new description as proposed by [103, 104], we do investigate the re-allocation of layer sections, *i.e.* segments of the layer data, to minimise transmission cost and maintain viewable quality over lossy networks. We explore the re-allocation of layer sections in both packetisation and in the creation of distinct classes of layers which are better suited to reflect the requirements of users and their distinct layer quality needs.

- Modifying the base layer to create two individual spatial descriptions [106], based

on downsampling residual data obtained by temporal prediction. Each base layer contains control information and the same motion vectors. While the loss of one base layer can be mitigated by copying the residual data in the received base layer or implementing complicated interpolation methods between base layer and enhancement layers.

Unlike [106] we do not alter the encoded SVC stream and as such do not investigate the benefits of multiple spatial versions of the same SVC stream.

- Optimising transmission cost by reducing the number of higher quality layers being transmitted per description. The authors of [107] modified the descriptions per GOP based on odd/even frame distribution and level of redundancy allocated to each description such as based on medium grained scalability (*MGS*), DC & transform coefficients and dropping non reference frames.

  While [107] can reduce transmission costs by dropping frames, reducing redundancy, extracting layers or removing internal stream symbols, this will limit the number of viewable quality levels that can be decoded. Which is counter productive to the original design benefits of SVC. Our work focuses on mechanisms for maximising the number of layers available to all users, so as to reduce the overall transmission cost on the network by providing one stream to suit all needs.

- The authors of [108] proposed that utilising the concept of redundant pictures inherent in the AVC standard will provide an increase in the viewable quality offered by MDC. A redundant picture can be utilised when the original frame is unable during decoding. In this regard the concept of redundant picture provides error resiliency to the original stream. In [108] two versions of the original frame are created. One frame will be a direct copy of the original frame, while the second frame will be a coarse reduced quality version of the original frame. The created frames are alternated every description so as to balance the total load over all descriptions transmitted. This option will increase overall transmission cost as the redundant pictures are coarse representations of the original primary image, with maximum quality requiring both descriptions.

  Modifying the original SVC encoding so as to create multiple versions of the stream, albeit based on spatial, temporal or quality dimensionality has been widely researched. As previously stated our work is focused on the transmission of layered data over lossy networks and maintaining the viewable quality of same, irrespective of the original SVC encoding or any modifications made to same pre-transmission. In this regard, our research can be utilised by [108] or any researcher who is interested in maintaining the viewable quality of layered data over lossy networks.

2.7.3 Hybrid Schemes - Adding Scalability to MDC

As stated, the benefits of SVC are over-shadowed by the detrimental affects of lower layer loss, while the elevated transmission costs of MDC are a mitigating factor in its deployment. Recent publications have proposed hybrid schemes, where the benefits of SVC are utilised to increase the quality provided by MDC and in some instances to reduce the transmission costs of MDC. An example of which is Scalable Multiple Description Coding (*SMDC*) [107, 109]. SMDC encodes one or more layers of an SVC stream into a number of differing quality levels, known as bitrates in SMDC. SMDC then generates numerous descriptions, per GOP, composed from these differing quality levels, where each description contains a different bitrate allocation per layer. It is typical to allocate the bit-rates in an odd-even frame distribution so as to provide consistency to the stream during decoding. In SMDC the decodable stream quality is dependent on the number of descriptions received as well as the allocation of the different bit rates, while each individual description can provide a minimal level of quality. It is common practice to perform device preprocessing so as to determine from the various bit-rates the maximum available quality per layer, thus providing the highest quality to the user.

While the underlying design of SMDC is different to previous publications illustrated, the concept of creating different quality versions is well researched. We do not focus on this distinct adaptation mechanism but we do investigate the benefits of introducing prioritised layer data to MDC. We consider the benefits of creating prioritised descriptions within a stream, *i.e.* different types of descriptions containing distinct layers sections, and determine how loss affects quality when description-based streaming is segmented into prioritised data.

2.7.4 Post Encoding Optimisation

The research introduced up to now can be viewed as adaption of the streaming data pre-transmission from the server to the client. But adaptation pre-transmission is not the only option for maximising stream quality. Client-based feedback mechanisms, which adapt user quality such as based on variations of loss in the network or on user specific requirements such as used by DASH are well researched in the literature. While bi-directional server/client traffic shaping feedback schemes such as [110, 111] propose to maximise quality for numerous users by optimising bandwidth utilisation of service providers.

Adaptive models implementing *multi-path routing* have also received attention in the literature. Multi-path routing [112] can be defined as routing data simultaneously over multiple links, where the links can be within one network or over multiple technologies, *i.e.* Wi-Fi, cellular and satellite, examples of this concept include:

1. *HTTP-streaming* - The authors of [113] present a client-side request scheduler that distributes requests for the video over multiple heterogeneous interfaces simultaneously. Video data is divided into smaller segments with a predefined constant duration, which enables segments to be transmitted over separate links, thus utilising all available bandwidth.

   While the authors of [114] propose an adaptive receiver-driven algorithm which detects bandwidth changes based on the segment fetch time and is used to determine if the bitrate of the current media matches the end-to-end network bandwidth capacity.

2. *AVC* - (these tend to be MDC-based variants of AVC Multi-path). The authors of [62] present a network-adaptive multiple description coding method, Multiple Description Scalar Quantisation (MDSQ), for enhanced video streaming over multi-path channels. MDSQ is used to split an SD video stream into two complementary streams (two descriptions) for transmission over a multi-path channel.

   While the authors of [115] propose a new MDC scheme, where each description has a different prediction loop and contains additional motion information for the frames included in the other description. This additional motion information is used to enhance the reconstructed video quality when only one of the two descriptions is received.

3. *SVC* - The authors of [116] examine the H.264 and H.265 standard, with respect to both research challenges and potential solutions. They provide a detailed case study of SVC streaming in multi-path mobile networks.

   The authors of [117] propose a transparent multi-path video streaming mechanism based on SVC. Their scheme adapts to network bandwidth fluctuation by observing the changes in the available bandwidth over the multiple overlay paths, using a Video Distribution Network (VDN) as the overlay-based infrastructure, and updates the streaming strategy accordingly.

   The authors of [118] propose a QoE scheme, based on a user feedback mechanism, for automatically selecting the optimal overlay path by which the highest level of quality can be achieved.

4. *MDC* - The authors of [119] propose a 2-D layered multiple description coding (2DL-MDC) for error-resilient video transmission over unreliable networks. The proposed 2DL-MDC scheme allocates multiple description sub-bitstreams of a 2-D scalable bitstream to two network paths with unequal loss rates.

   The authors of [120] investigate the delivery of MDC over multiple paths.

   The authors of [121] investigate the delivery of node data packetised using MDC, for transmission over an ad-hoc network, using a new multi-path protocol called

Topology Multi-path Routing (TMR). TMR is a multi-path reactive protocol (i.e. each node begins to look for routes when it has data to send). Contrary to existing reactive protocols, the topology information is gathered in the source, which then defines routes to the destination.

In our research we focus on pre-transmission optimisation techniques, selective allocation of layered data and subsequent packetisation mechanisms and as such do not investigate the benefits of in-network multi-path routing. However future work will investigate the benefits of combining our current work with the selective routing proposed by these techniques.

2.8 Goals for our Research

Scalable coding has evolved to provide the promise of independent stream qualities with efficient cumulative stream transmission, while minimising storage and transmission costs for multiple users. As can be seen the initial goal of SVC is to provide graceful degradation of viewable quality by incrementally decreasing layer quality decoding as loss rates increase has not yet been achieved. MDC can improve viewable quality but the large increase in transmission cost is detrimental to devices in constrained networks and especially for devices requesting lower layer quality decoding. It is important to note that the difference between SVC and MDC is the increased cost of error correction and the method for distributing the layer data across multiple descriptions.

2.8.1 Research Topics

For this work, the research topics to consider are:

i. Scalable Media: both layered and description-based.

ii. Network Loss: with specific reference to packet loss of streaming data and the variation in viewable quality that can occur due to prioritisation in the stream.

iii. Packetisation: and how the stream data are divided over numerous packets.

iv. Encapsulation of layered data: relating to description structure and layer allocation.

v. Consistency of viewable quality over time: and why variations in network loss rates mitigate the stability of achievable quality.

vi. Adaptation: to both network conditions and user requirements.

The encoding efficiency aspect of SVC is sufficiently detailed in the standard and optimisation of this process is not part of this work.

2.8.2 Our Goal

These are the goals of our work:

i. To strike a balance between the consistency of viewable quality over time, *i.e.* over numerous GOP, and the increased transmission cost necessary for selective error correction.

ii. To identify how network loss affects viewable quality for each of the existing scalable adaptive schemes, *i.e.* SVC and MDC. How much of an increase in viewable quality can the FEC unequal error protection of MDC provide and is there a *threshold* where a further increase in error correction does not mandate increased levels of viewable quality.

iii. Assess the variation in viewable quality that occurs as the number of frames per GOP increases for selected network loss rates. As SVC is a prioritised layered concept and GOP is a prioritised frame interdependent mechanism; will lower layer decoding in a prioritised frame within a GOP mandate low viewable quality for all dependent frames.

iv. Determine a means of increasing data equality in scalable media; this can also be viewed as decreasing prioritisation in the layer data. SVC has no inherent equality as each individual layer provides a different level of prioritised data. MDC has taken preliminary steps to provide equality of data, *i.e.* each MDC description per GOP contains the same percentage of each SVC layer, thus each description has the same priority.

v. Create new streaming models which re-allocate, or distribute, the original SVC layer data coupled with selective levels of error correction. Thus providing adaptive techniques for mitigating network loss with minimal levels of increased transmission cost.

vi. Design new techniques which can increase viewable quality for existing models, without the need for an increase in transmission cost.

2.9 Conclusion

As we have seen MBR techniques, such as DASH, overly increase the transmission cost over the network, and the storage requirements at the server and in-network CDNs, due to the heterogeneous user requirement of numerous versions of the same video content, albeit at different resolution or quality levels. Scalable, or layered, video provides the promise of independent stream qualities with efficient cumulative stream transmission, while minimising storage and transmission costs for multiple users, and providing graceful degradation for individual users during periods of packet loss in the

network. Current implementations of scalable media streaming either fair badly during lossy transmission due to the inherent prioritised hierarchy, *i.e.* SVC, or overly increase transmission cost by implementing static levels of unequal error protection, *i.e.* MDC.

In the following chapters, we present our techniques and scalable streaming models which offer a balance between the consistency of viewable quality over time and the increased transmission cost necessary for selective error correction, while mandating consistency of quality, at higher quality levels, for longer periods of time.

Chapter 3

Scalable Description Coding (SDC)

In this chapter, we present Scalable Description Coding (*SDC*) [14], a technique that enhances MDC with a novel transmission scheme to achieve lower data rates without sacrificing user-perceived quality. SDC operates by redefining the MDC description prior to transmission, to reduce the required bandwidth. Compared to MDC, SDC improves the user-perceived quality with lower bandwidth usage, while offering increased robustness against packet loss. Our analysis quantifies the data rate reductions, showing that in some instances the SDC data rates are on par with those of SVC. Furthermore, we propose several optimisations to SDC, including SDC with network coding [122], that further improve SDC performance.

Table 3.1: Main SDC Notation

Q	The stream quality value, based on SVC layer index
M	The number of SVC layers per frame
$L_{l,x}$	Transmission cost of SVC Layer l, L_l, for frame x
$S_{i,j}$	The i^{th} section of layer j
D_c	A complete MDC description composed of a section from all layers
D_s	A scalable SDC description composed of enhancement layer sections
D_r	A redundancy SDC description composed of an FEC section from all layers

3.1 Scalable Description Coding (SDC)

An SVC stream consists of M layers, and to increase resilience to network loss this stream is encoded as an MDC of N descriptions. Hence, the MDC representation of

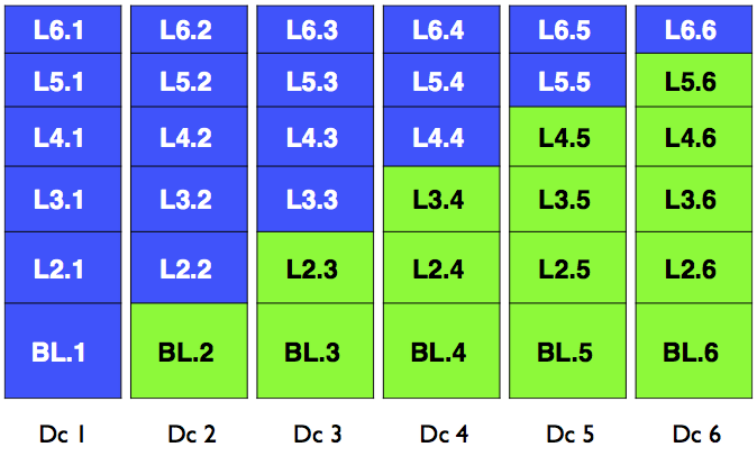


Figure 3.1: An example of a 6 Layered SVC stream encoded as MDC c/w FEC

each frame can now be viewed as a matrix of M times N *sections*. In keeping with the relationship between the receipt of an additional MDC description and the incremental increase in stream quality, it is most natural that M and N be equal, *e.g.* to increase the viewable quality from layer i to $i+1$, one additional description is required, such that to view layer M, M descriptions are required. The distribution of M layers over M descriptions is not required for MDC, but is commonly used in the literature as it provides graceful degradation of viewable quality as network loss increases. The allocation of an increased number of layers per description will reduce the number of descriptions transmitted, *i.e.* less than M, but will increase the effects of network loss on viewable quality as a greater number of layers will be undecodable due to lost descriptions.

As we have seen in Section 2.4.1 and specifically using the 6-layer example in Figure 2.8 on page 24 (reproduced in Figure 3.1 for each of access), MDC utilises layer partitioning to create i sections for SVC layer i (shown in blue). Then MDC uses FEC to extend the i section(s) over N descriptions (shown in green), with the maximum stream quality, $\max_Q$, of MDC based on the number of sections received for each layer. Thus, $\max_Q$ is based on the higher layer i where a minimum of i sections have been received at the device. We can see that horizontally the lower layers have an increased level of FEC redundancy, each layer i has N-i redundant sections and the highest layer contains no FEC redundancy. While vertically, as the index of the MDC description increases, the number of original SVC layers contained within the description decreases, i.e. all layers in Dc 1 are composed of original SVC data while only layer 6 in Dc 6 contain SVC data, assuming a systematic scheme (a non-systematic scheme contains the same level of FEC and SVC data but interspersed together over all descriptions). This FEC redundancy translates to an increased transmission cost and a higher consumption of device computation and energy resources. The following subsection explains how SDC can reduce these drawbacks, followed by the analysis of a four-layer video example.

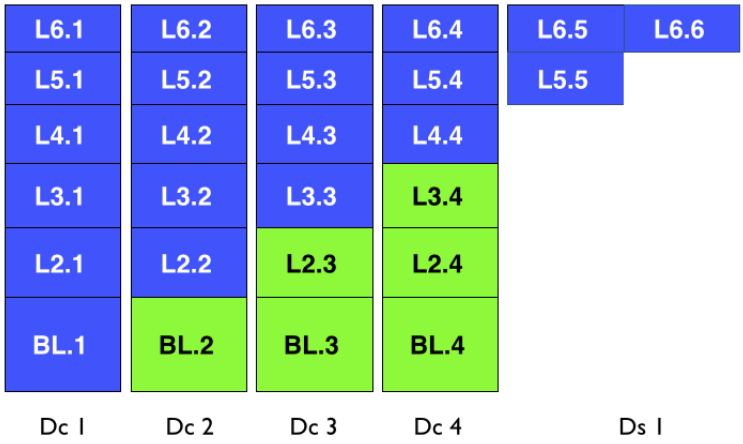


Figure 3.2: An example of the reduced levels of FEC required when a scalable description is created from two MDC descriptions (D_c 5 and D_c 6)

3.1.1 SDC Overview

As we have seen in Figure 3.1 the highest enhancement layers contain the lowest FEC redundancy, *e.g.* layer 6 contains no FEC allocation. In comparison the MDC descriptions with the highest index value, *e.g.* D_c 5 and D_c 6, contain the greatest levels of layer FEC redundancy. While the difference in transmission cost between SVC and MDC is the total level of FEC over all layers (all green sections). The design of SDC aspires to reduce the number of MDC descriptions required, and subsequently the level of inherent FEC redundancy. It achieves this by reallocating a subset of the enhanced layer sections from the higher index MDC descriptions to a new *scalable* description prior to transmission and removing the FEC redundancy of all lower layers within the higher index MDC descriptions no longer required.

Figure 3.2 illustrates an example of the reduced FEC allocation achievable when we create a scalable description, D_s 1, based on Figure 3.1, composed of original SVC data sections from the highest two layers, L5 and L6, of descriptions D_c 5 and D_c 6. It can be seen that L5.5, L6.5 and L6.6 are grouped together to create D_s 1, while the FEC sections from D_c 5 and D_c 6 have been removed, thus reducing overall transmission cost. Note how the layer 4 and the layers within D_s 1 contain no FEC allocation. FEC redundancy for these layers are discussed later in the chapter.

The goal of SDC is to decrease the number of N descriptions required to transmit a media stream while maximising the level of stream quality, at $\max_Q$, received at the device. Rather than remove the FEC sections from Figure 3.1 as illustrated in Figure 3.2, SDC works from the pre-FEC allocation state of MDC, as previously illustrated in Figure 2.7 in Section 2.4.1 (reproduced here in Figure 3.3). Pre-FEC allocation, SDC will reallocate a number of the original SVC enhancement layer sections to a scalable description, so as to reduce FEC resiliency, *e.g.* reduce the number of FEC sections generated for lower layers and thus the number of transmitted descriptions.

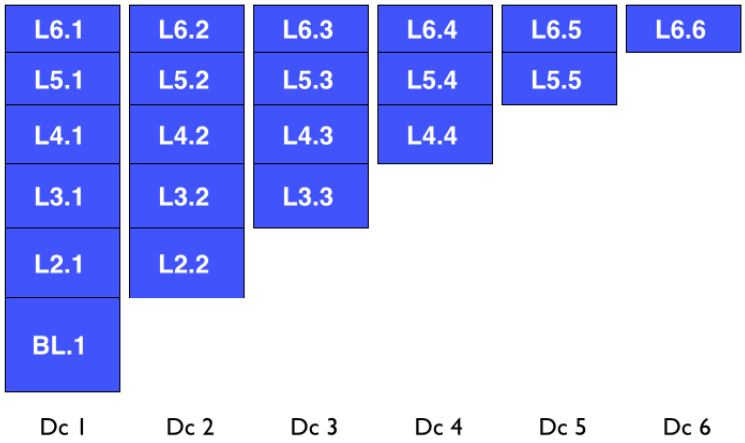


Figure 3.3: A six-layer SVC encoding distributed over six MDC descriptions, prior to FEC allocation

FEC will be allocated to the MDC descriptions once the scalable description has been created, thus creating the description allocation as previously illustrated in Figure 3.2.

SDC consists of three types of descriptions:

i) A Complete description - D_c is identical to an MDC description in that it contains one section from each layer in a GOP. Similar to MDC, one or more D_c descriptions can be transmitted per GOP, with n D_c descriptions providing a maximum quality level of layer n. In an SDC stream only the D_c description(s) will contain a combination of SVC and FEC data. The primary role of the D_c description is the delivery of the lower layer sections of the stream so as to provide a maximum quality level of layer n.

   Based on Equation 2.4 from Section 2.4.1, Equation (3.1) defines the transmission byte cost of a D_c from frame x as:

$$Cost_{D_c} = \sum_{l=1}^{N} \frac{L_{l,x}}{l} \tag{3.1}$$

   Pseudocode to define the D_c is presented in Algorithm 1.

ii) A Redundant description - D_r: the role of this description is to reduce the effects of network loss. As illustrated in Figure 3.2, layer 4 and the layers within D_s 1 contain no FEC allocation. We propose to transmit one D_r per GOP to increase the resiliency of these layers to network loss. Further increasing the number of D_r is counter productive to the goal of SDC where by the level of FEC is reduced. The D_r is formed by utilising FEC to create an additional FEC section for each layer per GOP. The composition of D_r is identical to the D_c in that it contains one section from each layer in a GOP. Equation (3.1) can be rewritten to define

Algorithm 1 MDC byte size calculation:

```
1: sizeMDC = 0 //initial byte size of the MDC description
2: i = totalNumSVCLayers //total number of SVC layers
3: //for each layer
4: for i > 0 do {}
5:    //increase byte size of MDC by the byte size of the current layer section i
6:    sizeMDC ← sizeMDC + layerSectionSize(layerSize, i)
7:    //layerSectionSize() allows us to determine the section byte size for a given layer
      byte size by divided by the current layer i
8:    i ← i − 1 //decrement i by one
9: end for
10: return sizeMDC
```

D_r:

$$Cost_{D_c} = Cost_{D_r} = \sum_{l=1}^{N} \frac{L_{l,x}}{l} \tag{3.2}$$

The same pseudocode as defined for the D_c in Algorithm 1 can be used for the D_r.

iii) A Scalable description - D_s: the main role of this description is to provide for an increase in the level of viewable quality from layer n to layer N. This is achieved by increasing the number of higher enhancement layer sections available during decoding. We mandate the transmission byte cost of the D_s to be less than or equal to that of a D_c description, so as to not increase overall transmission cost in edge cases where the number of transmitted descriptions remains the same. Finally, similar to D_r only one D_s will be transmitted per GOP, as increasing the number of D_s will reduce the FEC resiliency of lower layers to a level which is unable to cope with network losses, which in turn reduces overall viewable quality and is counter productive to the goal of SDC where by high levels of viewable quality is maintained. D_s is formed by combining several sections from the higher enhancement layers in an iterative downward fashion and as such contains no FEC resiliency.

The intuition for this design is based on the MDC structure pre-FEC allocation, as illustrated in Figure 3.3. The allocation of the last section from the highest layer to the D_s reduces the number of D_c required by one and reduces the FEC transmission cost of all lower layers by one section. Each iteration will add all layer sections from the next lower description while $Cost_{D_s}$ is less than or equal to $Cost_{D_c}$, thus further reducing the number of D_c and the level of FEC required by all lower layers. If we reuse our previous example from Figure 3.2, the steps

required to go from Figure 3.3 to Figure 3.2 are:

1. Initially, one section from the highest layer is added to the scalable description, $S_{N,N}$ ($S_{6,6}$ - section 6 from layer 6 in our example), this iteration is the base case.

2. As state, we do not want to increase overall transmission cost so we mandate that the transmission byte cost of the D_s to be less than or equal to the transmission cost of a D_c description. Thus, subsequent iterations shall only commence, iff the byte size of one section from the next layer down, $S_{N-1,N-1}$ (*e.g.* $S_{5,5}$), plus one additional section from every layer added so far, $S_{N-1,N}$ (*e.g.* $S_{5,6}$), is less than or equal to $Cost_{D_c}$.

   For each iteration, including the base case, we reduce the number of D_c required to decode the stream by one, while also reducing the levels of FEC transmitted. In this example, we assume that the cost of the sections allocated so far, *i.e. sections* $S_{6,6}$, $S_{5,6}$ *and* $S_{5,5}$, equate to the transmission cost of the D_c. Thus, we have reduced the number of D_c required by two, which also reduces the level of FEC transmitted, with respect to the lowest layers. Note how two sections from the Base layer to Layer 4 inclusive, and one section from Layer 5 have been discarded due to the creation of the D_s in Figure 3.2.

3. Thus to finalise, FEC is allocated to descriptions Dc 1 to Dc 4 inclusive. Equation (3.3) defines the section allocation, and overall transmission cost, for the scalable description.

$$Cost_{D_c} \geq Cost_{D_s} \ : \ Cost_{D_s} \ = \ \sum_{i=1}^{N} \sum_{j=1}^{i} S_{N-i+1,N-i+j} \tag{3.3}$$

One benefit from this manner of D_s creation, is that the value of the layer, layer i, with only one section added to the D_s (layer 5 in our example), is the combined number of D_c and D_r required by SDC. Thus allowing us to define the total transmission cost as Equation (3.4):

$$Cost_{total} \ = \ Cost_{D_c} * (i-1) + Cost_{D_s} + Cost_{D_r} \tag{3.4}$$

The transmission scheme for an SDC stream is comprised of one or more D_c, one D_s and one D_r, transmitted in that order. Pseudocode to define the D_s is presented in Algorithm 2.

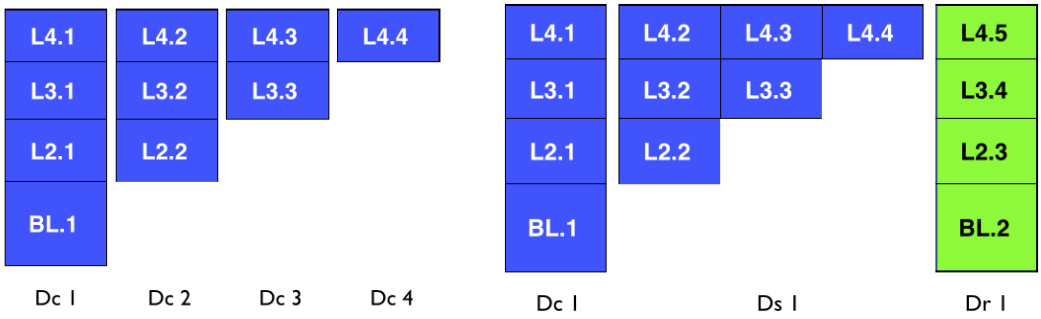
Algorithm 2 Determine the byte size of the SDC scalable description:

```
1: //This function scalableLayer( byteSizeMDC, currentLayer, currentSDCByteSize) is initially called with scalableLayer( byteSizeMDC, 0, 0)
2: numSVCLayers //total number of SVC layers
3: layerSize = SVCLayerByteSize
4: cl = currentLayer
5: sBS = currentSDCByteSize
6: while cl ≥ 0 and sBS ≤ byteSizeMDC do
7:    sBS ← sBS + (layerSize/(numSVCLayers − cl))
8:    cl ← cl − 1
9: end while
10: if sBS < byteSizeMDC then
11:    //call this method again, with the same MDC bytesize, increase the number of layers by 1, and include the current value of the scalable bytesize
12:    return scalableLayer(byteSizeMDC, currentLayerd + 1, sBS)
13: else if sBS ≥ byteSizeMDC then
14:    //we now know the lowest layer we can reach, so create the SDC scalable layer
15:    clCounter = currentLayer
16:    numSection = 1
17:    while clCounter ≤ numSVCLayers do
18:       //add the number of section for the current layer to the scalable descriptions
19:       addSections(clCounter, numSection)
20:       clCounter ← clCounter + 1
21:       numSection ← numSection + 1
22:    end while
23:    //return the total number of complete and redundancy descriptions required
24:    return currentLayer
25: end if
```

3.1.1.1 A Four-layer Video Example

As our previous six layer example was used to illustrate the D_s creation, such that the size of the D_s was assumed, we now provide an example in which the D_s is determined based on the byte size of the SVC layers, which is consistent with the evaluated results shown later. We provide a simple four layer example in which we assume that the transmission byte cost of each SVC layer is 300 bytes. Reducing the example from six layer to four layers permits a simplification of SDC design and D_s creation. While the byte size is chosen purely to simplify the example by allowing each layer/description to be transmitted within the size of a typical un-fragmented IP packet. By assuming a value of 300 bytes per layer, this yields an SVC GOP of 1,200 bytes - 4 layers x 300 bytes (assuming a GOP value of one, *e.g.* one frame per GOP). Note that in a more typical case where the layers are not the same byte size, but incrementally larger/smaller, the same mechanism is employed but performance gains will vary due to the changes in section sizes and the number of D_c transmitted. Let the corresponding MDC representation have four descriptions ($N = M$), with each description consisting of 625 bytes

(BL: 300 ($\frac{300}{1}$) + L2: 150 ($\frac{300}{2}$) + L3: 100 ($\frac{300}{3}$) + L4: 75 ($\frac{300}{4}$)), determined using Equation 3.1, thus totalling 2,500 bytes, 625 x 4, for the MDC GOP of 1, an increase cost of 109% relative to SVC. Clearly, to improve the stream reliability, MDC adds significant overhead to SVC, providing a sufficient motivation for SDC to offer efficiencies by reducing bandwidth demands but without sacrificing user-perceived quality.



(a) MDC Pre-FEC **(b)** SDC Post-FEC

Figure 3.4: (a) 4 MDC description example pre-FEC allocation as a (b) 3 SDC description example post-FEC allocation

By utilising Equation (3.2) and Equation (3.3), SDC can re-packetise the sections from Figure 3.4a into the descriptions in Figure 3.4b. D_s 1 is composed of three sections from layer 4 (3*75), two sections from layer 3 (2*200) and one section from layer 2 (1*150), thus yielding a $Cost_{D_s}$ of 575 bytes. Similar to $Cost_{D_c}$, $Cost_{D_r}$ has a byte cost of 625 bytes. It can be seen that D_c 1 combined with D_s 1, now contains all of the data required to decode the original stream to its highest quality, *e.g.* four layers in this example, and is on par with the transmitted byte size of the original SVC encoding. Owing to the manner in which the D_s 1 is created, one additional section, L4.5, is required. This additional section is mandatory, as D_r 1 must contain one FEC section for layer four. This section can be computed by replicating L4.1, as D_s 1 already contains the other three sections from layer four, or by adding an FEC section. In this manner, we have reduced the number of descriptions transmitted over the network from four to three.

By using Equation (3.4) we can determine the $Cost_{total}$ to be $625 + 575 + 625 = 1,825$ bytes, thus yielding bandwidth savings of 28% over MDC but a bandwidth increase of 52% over SVC in this example. Increased savings can be achieved for SVC streams with larger numbers of layers. Total relative overhead between SDC and SVC can be calculated as the difference between the total transmission cost of SDC, $Cost_{total}$ from Equation (3.4) less the transmission cost of SVC for the highest quality layer, layer 4 in this example, as defined in Section 2.2

$$Cost_{total} - SVC(4) \tag{3.5}$$

Table 3.2: Number of sections allocated to each type of description in SDC, byte size of each section per description and number of sections required to decode a layer using a four layer example

	Sections Required:				
	MDC/SDC Complete Desc		SDC *Scalable* Desc		To Decode
Layers	4 Layer		4 Layer		a Layer
4	1	75	3	225	4
3	1	100	2	200	3
2	1	150	1	150	2
BL	1	300	0	0	1
Total		625		575	

Table 3.2 highlights the number of sections allocated to each type of description in SDC, byte size of each section per description and number of sections required to decode a layer using a four layer example. Note that to decode layer N, all sections for layers 1 to N must have been received. As can be seen, a D_c contains one section from each layer, whereas the D_s can contain numerous sections for a given layer. Note that the byte size difference between D_c, 625, and D_s, 575, could be used by SDC as a construct for message handling between server and device. SDC has similar dependency hierarchies to both SVC and MDC. Like SVC, the D_s is of a higher priority due to the potential of reduced bandwidth and increased stream quality but similar to base layer loss in SVC, the dependency hierarchy in SDC, also increases the possibility of frame loss where only D_s is received. Whereas like MDC, if any combination of D_c or D_r is received, the system performs exactly as an MDC system, in which a reduced quality version of the stream is decodable and the device/network incurs approximately 52% inherent bandwidth loss, due to the receipt of sections of higher enhanced layers which are undecodable. Furthermore, SDC does not introduce any additional constraints on media coding/decoding beyond those for MDC. Clearly, SDC yields significant bandwidth savings in comparison to MDC as shown in Table 7.1. However, the bandwidth overhead of SDC relative to SVC, 52% in our example, is certainly not negligible. This issue will be addressed by the optimisation mechanisms presented in the following section. Note how for the D_s contains only one section from layer 2 which mandates that one D_c is needed and that a total of three descriptions shall be transmitted for this frame (1* D_c, 1* D_s and 1 * D_r), a reduction of one descriptions over MDC.

Table 3.3 presented the same overview as Table 3.2 but using a six layer example based on a 300 byte cost per layer. Note how for the D_s contains only one section from layer 3 which mandates that two D_c are needed and that a total of 4 descriptions shall be transmitted for this frame (2* D_c, 1* D_s and 1 * D_r), a reduction of 2 descriptions over MDC.

Table 3.3: Number of sections allocated to each type of description in SDC, byte size of each section per description and number of sections required to decode a layer using a six layer example

	Sections Required:				
	MDC/SDC Complete Desc		SDC *Scalable* Desc		To Decode
Layers	6 Layer		6 Layer		a Layer
6	1	50	4	200	6
5	1	60	3	180	5
4	1	75	2	150	4
3	1	100	1	100	3
2	1	150	0	0	2
BL	1	300	0	0	1
Total		735		630	

3.2 SDC Optimisations

In this section we present two additions to the basic SDC approach, each of which serves to offer an improvement on performance.

3.2.1 SDC-NC: Network Coding for SDC

We propose incorporating network coding (NC) [123, 124] into SDC to improve the overall performance. Network coding provides a means of reducing transmission cost by using exclusive disjunction, or XOR, on the bits within two packets, which effectively halves transmission cost. The concept of network coding is commonly presented by using routing over multi path networks, where data are from multiple nodes is combined over low throughput links. One feature with network coding is that to recreate the original two packets, one of the original packets must be available at the receiver.

In SDC-NC, the redundancy description, D_r 1, as outlined in Section 3.1, is replaced by a network coded (NC) description, D_r 1 $_{-nc}$, which is the exclusive disjunction, or XOR, symbolised by $\oplus$, of all the leading descriptions for a given GOP, D_c 1 $\oplus$ D_s 1 in our example. The creation of D_r 1 $_{-nc}$ using D_c 1 $\oplus$ D_s 1 is illustrated in Figure 3.5. How the individual sections of D_c 1 and D_s 1 are XORed is clearly seen in D_r 1 $_{-nc}$. As with most NC implementations, should the byte size of the D_s be smaller than a D_c, then the D_s is padded with trailing 0s [125], to maintain a description of equal size and balance the impact of the NC mechanism. In our example sections of the same byte size are XORed together, while BL.1 (300 bytes) is XORed with L3.3 (100 bytes), L4.3 (75 bytes) and L4.4 (75 bytes), with the remaining 50 bytes being created from padded 0s. By utilising SDC-NC, individual devices can recover from the loss of any description (scalable or complete) by receiving the NC description. In this manner, if *any* two descriptions are received without loss by the client device in the

Table 3.4: Notation and Definitions

N	The number of SVC layers per Group of Pictures (GOP)
L_l	Byte-size of SVC Layer l
$S_{l,x}$	Byte-size of a Layer section of SVC Layer l for frame/GOP x
l	Integer value corresponding to the layer number of L_l
GOP	The number of frames per GOP

four-layer example, the full quality stream can be decoded.

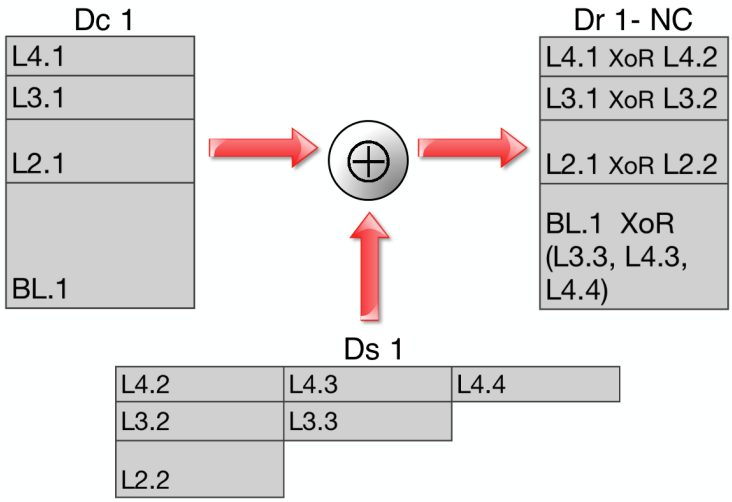


Figure 3.5: Network Coding design for SDC-NC.

While NC significantly improves the resiliency of SDC to errors and losses, it is important to note that while the quality of the decodable stream generally increases, the number of scenarios in which no frame can be decoded also increases. In this manner, the NC description can also be seen as a prioritised description, such that receiving only that description increases frame loss and as a result decreases stream quality.

3.3 SDC Packetisation

Before we introduce our Evaluation Methodology, we first present our novel packetisation techniques [15, 16] which are common to all our new streaming models and can compliment existing streaming models. To provide ease of access to notation used in this section, we repeat the relevant definitions from Chapter 2, Table 2.1.

As we have seen in scalable streaming, packet loss in lower layers can drastically reduce viewable quality. Our packetisation technique reduces the impact of packet loss on any description-based scalable video by selectively choosing how to packetise the data and which data to transmit. As previously stated the application transmission

unit for MDC is its description. For purposes of illustration, we use a single GOP example from the widely-used video clip known as crew.yuv, encoded as a six-layer SVC stream, consisting of three resolutions and two quality levels per resolution. Table 3.5 shows the byte-size of each layer for a selected frame.

Table 3.5: GOP SVC Layer sizes (bytes)

Layer	1	2	3	4	5	6
Layer Size	1442	1577	1601	1546	1255	3372

In today's Internet, the maximum packet size observed is usually limited by the use of Ethernet, with a maximum payload of approximately 1,500 bytes. We assume an application packet payload of approximately 1,440 bytes, allowing for overhead due to headers of network, transport and streaming media protocols. For our single GOP example, eleven Ethernet packets would be required using SVC where the application transmission unit is an individual layer, such that each layer is individually packetised, *e.g.* layer 1 would be transmitted over 2 packets, layer 2 would be transmitted over 2 packets, and so on. The same frame would require eighteen packets when encoded using MDC in which the description represents the application transmission unit.

Dependent on the transmission cost of the layer sections and on how the MDC descriptions are packetised, an MDC packet may contain a complete section from various layers, or a subset of an entire section for a given layer, i.e. some layer sections may be shared between numerous packets. In our example frame, the base layer is 1,442 bytes which can be directly allocated to an MDC packet. Layer 2 is 1,577 byte, which equals to an MDC section size of approximately 789 bytes. Layer 3 MDC section is 534 bytes and layer 4 is 387 bytes. The sections of layer 2 and layer 3 can be allocated to the next packet as well as 119 bytes from the layer 4 section. Thus the layer 4 section must be shared over 2 packet.

On losing any of these MDC packets, the application would not be able to decode the *entire* frame to the highest quality, i.e. any percentage of MDC loss negates decoding of the highest viewable quality. In order to reduce the impact of losses on the stream quality, we utilise two packetisation mechanisms, called section-based description packetisation and section distribution.

3.3.1 Section-Based Description Packetisation

With section-based description packetisation (SDP), we propose using sections as application transmission units instead of the entire description for description-based layered coding techniques. In this regard, we separate the entire description into the number of sections of the underlying SVC layer data. Each section is transmitted as a single unit, thus limiting the effect of packet loss to an individual section, i.e. and

individual SVC layer, while allowing partial description re-use of the received sections. Partial description re-use in this instance means the availability at the device of one or more layer sections from a single description. The probability of loss affecting all sections from a single description, or all sections from a single SVC layer, is low, while the probability of partial description re-use is high. In the literature numerous examples are found where an individual MDC description is separated into numerous descriptions [103, 104] or retransmitted using different scalability levels, i.e. temporal, spatial or quality levels [106]. SDP does not create new descriptions, change the FEC allocation of the underlying steaming model or modify the original SVC layer encoding. We simply selectively allocate MDC description sections during packetisation so as to reduce the impact of packet loss on viewable quality.

As description-based streaming models contain higher levels of FEC, *i.e.* a greater number of sections, for lower layers, SDP improves the possibility of higher stream quality by mitigating the effects of lower layer loss thus increasing the availability of a sufficient number of lower layer sections at the device.

SDP can be applied in several ways as follows:

- *Option 1 - Individual layer sections* - this option transmits each layer section as a separate group of one or more packets. This option may increase the number of packets being transmitted, depending on the original encoding but maximises the number of sections available during decoding. Using the example frame, it can be seen that for each MDC description six packets are required for transmission as shown in Figure 3.6. This option increases the number of packets and in some instances creates packets not containing a full data payload. Consequently the overhead due to packet headers and processing is higher.

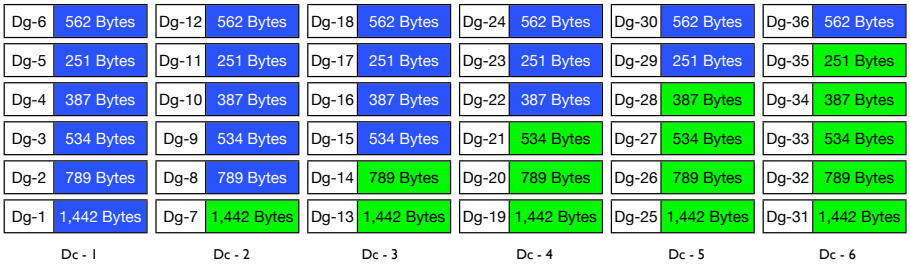


Figure 3.6: MDC-SDP Option 1 - with six descriptions (D_c) consisting of six packets (D_g)

- *Option 2 - Minimising packet quantity* - this option groups layer sections together to fully occupy each transmitted packet, thus mitigating the problems with Option 1. Figure 3.7 illustrates this option for the example frame. This option reduces the number of transmitted packets relative to Option 1, but can increase the number of packets transmitted relative to MDC. From our example frame the number of packets required by option 2 is eighteen packets, which in this instance is the same number of packets as MDC.

The loss of a packet for option 2 may cause the loss of numerous layer sections, i.e. in Figure 3.7 D_g-3 contains sections for layers four, five and six. To reduce the probability of stream degradation due to packet loss, only one section for each layer should be included within a packet. If a packet were to contain numerous sections for one specific layer, then the loss of that specific packet may aversely affect the decoding of that layer and all enhanced layers that rely upon it. It can be seen that for each description, D_c, three packets, D_g, are required for transmission.

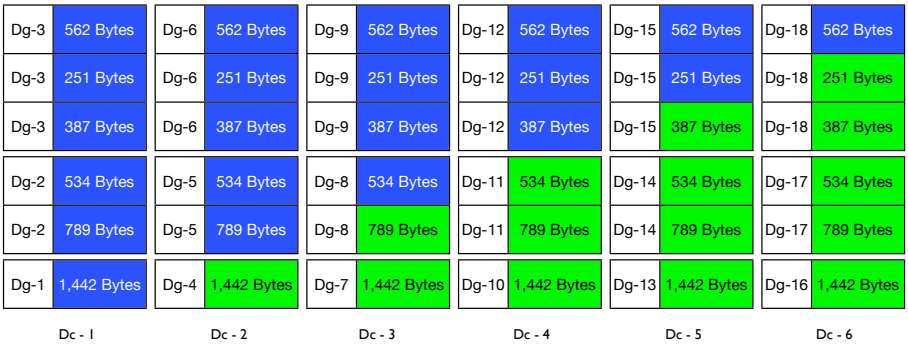


Figure 3.7: MDC-SDP Option 2 - with six descriptions (D_c) consisting of three packets (D_g)

It is worth noting that the blue (dark) sections are the critical SVC data and the green (light) sections the FEC section allocation. It can be seen that in Figure 3.6 and 3.7 that the base layer consumes a single packet, D_g-1 in description one; in Figure 3.6 each section is allocated to an individual packet while in Figure 3.7, a section from layer two and three are allocated to D_g-2 and a section from layer four, five and six are allocated to D_g-3. As six descriptions are transmitted, a total of thirty six packets are transmitted over the network with Option 1 in which only twenty one specific packets are required for maximum stream quality. In Option 2, eighteen packets are transmitted among which only ten specific packets are required for maximum stream quality. Thus Option 1 increases the ability to maximise stream quality in the presence of high levels of packet loss, *i.e.* if in a bursty loss scenario fifteen packets were lost from this GOP, then for option 2 the maximum achievable quality would be layer 2 (assuming one base layer packet and two packets containing layer 2 are received), while for option 1 the maximum achievable quality would be layer 6 (assuming all FEC packets are lost).

3.3.2 Section Distribution

Network traffic can be affected by both *individual* and *burst loss* corresponding to a single packet loss or numerous contiguous packet losses. Lower layer packet losses have a negative impact on scalable video due to inter-layer dependency. As shown above with SDP, by manipulating the stream packetisation, we can increase stream quality and consistency. With this in mind, we propose Section Distribution (SD),

Figure 3.8: SD packetisation of D_c 2 from MDC in Figure 2.8 from Chapter 2. It can be seen that the packet contains a section segment from each layer (red denotes packet header)

a mechanism that leverages the concept of section packetisation proposed by SDP but distributes all sections contained within a description over the same number of packets. In this manner, each packet contains a piece, known as a *segment*, of each section per description, as illustrated in Figure 3.8. Thus aiming to limit packet loss to only a *segment* of each section per description. A segment in this context is a subset of a layer section. In this regard SD packetisation can be used to transmit a number of descriptions from the same GOP so as to further reduce the impact of losing critical sections. SD takes inspiration from both the well-known Interleaving [126] technique, which is widely used to combat the effect of burst loss and from Priority Encoding Transmission (PET) [105], a prioritised packetisation scheme. Interleaving is the concept of moving data from one location to another location, from a layer to a packet in our case, so as to minimise the effects of loss, while PET prioritises data during packetisation so as to recover from network loss. While SD proposes that we move a segment from all sections of a description to a single packet, which is similar to Interleaving, and we manipulate the allocation of data during packetisation, which is comparable with PET, no prior work has proposed the packetisation as suggested by SD.

We first determine the number of packets, denoted as R, required to transmit a single description for each frame per GOP. This is achieved by dividing MDC_{D_c} from Equation (2.5), MDC from Section 2.4.1, by the data byte-size of a packet payload.

$$R = \left\lceil \frac{MDC_{D_c}}{packet\ payload} \right\rceil \tag{3.6}$$

Next, we specify the byte-size of each layer section, S_l, that is allocated to a single packet, D_g. For generalised use, we can define the byte-size of each layer section based on the underlying frame number, $S_{l,frame}$. This permits us to define layer section allocation for each packet over a predetermined number of frames per GOP. Thus each layer section per frame per GOP, denoted as $S_{l,frame}$, is spread over the R packets by allocating a segment of each layer from each frame per GOP to a single packet, D_g, as per the following

$$D_g = \sum_{frame=1}^{GOP} \left(\sum_{l=1}^{N} \frac{S_{l,frame}}{R} \right) \tag{3.7}$$

In this manner, all packets per GOP are of *equal priority*, as each packet contains the same byte-size, *i.e.* quantity, of each layer per frame per GOP. Thus the loss of an individual packet, will result in a partial loss from each layer. Thus the quality of the packetised stream is limited only by the percentage of lost packets rather than the specific carried description or layer. Additionally, the probability of losing critical sections is reduced since lower layers enjoy greater FEC redundancy.

Furthermore, on using section distribution, packets per frame would be identical in size and content, thus providing *packet equality*. This equality is provided in both packet byte-size and packet priority. Also as the GOP value increases, then SD will provide data equality for all frames within the GOP. In [45], the authors highlight that packets of dissimilar processing times in the queue of the routing device, *i.e.* router, switch, firewall, etc. produce dissimilar transmission times due to packet size, firewall packet checks or QoS requirements. Such that by maintaining such packet byte-size equality, the order of packet delivery is improved. Thus SD packet equality results in a consistent delivery in network transmission. The SDP and SD claims made in this section are generally applicable to all videos that have varying numbers of scalable layers. In the following chapters, when either an existing or our new streaming models use these techniques, it shall be clearly highlighted in the text. An example of which would be: *MDC-SDP*, which denotes MDC packetisation using SDP.

3.4 Evaluation Methodology

In this Section we present an overview of the methodology utilised to evaluate our research. This methodology is utilised in the evaluation sections of this and later chapters. Figure 3.9 presents a flow chart overview of the methodology steps. As each of these steps contains numerous variables which can be combined and adapted to suit user and network conditions, we only highlight the most commonly used and provide additional information in the relevant evaluation section of this and later chapters as well as in the appendices.

We begin this section by describing the video clips used in our work.

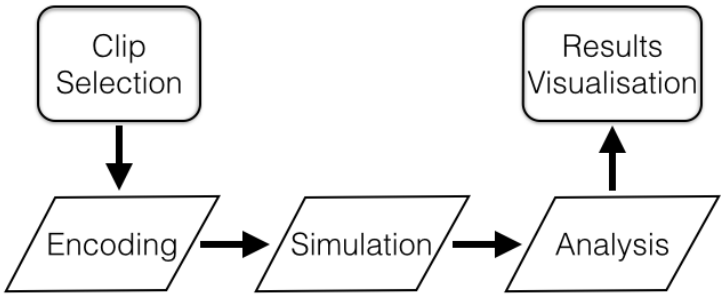


Figure 3.9: Overview of the steps implemented to evaluate our research

3.4.1 Video Clip Types

We utilised five, well known, distinct YUV video clips. Four of which are ten seconds in duration: the *city*, *crew*, *harbour* and *soccer* videos, all obtained from the Leibniz Universität Hannover video library [127] and the fifty two second *sintel* trailer obtained from [128]. We choose these specific clips because 1) they are well-known and widely used in the Literature, 2) they offer different levels of temporal and spatial complexity and 3) they offer two levels of duration. During the decoding process, typically the variation in achievable quality is dependent on the quality determined for the frames within each GOP. Thus, the clips of ten-second in duration are sufficient to illustrate the variation in achievable quality for a selected streaming model, as well as for a different number of frames per GOP. The longer fifty two second clip is used to corroborate our reasoning for the ten-second clips as well as illustrate how variations in quality over longer periods of time can influence a viewers choice in perceived quality. YUV is a file format which defines the colours displayed in a frame as tuple of numbers. Y typically stands for the luma or brightness component, while UV represents the chrominance or colour components. U and V tend to be half the bitrate of the Y component. Our evaluations results reference the Y-PSNR luminance values.

1. *City* - 10 second 300 frame low resolution low moving media clip, of an aerial view of a city skyline, with specific focus on one building. Maximum resolution of the original YUV file is 704x576 (*4CIF*). SVC encoded resolutions include: 176x144 (*QCIF*), 352x288 (*CIF*) and 704x576 (*4CIF*).

2. *Crew* - 10 second 300 frame low resolution low moving media clip, of a number of astronauts walking down a corridor while waving. Maximum resolution of the original YUV file is 4CIF. SVC encoded resolutions include: QCIF, CIF and 4CIF.

3. *Harbour* - 10 second 300 frame low resolution low moving media clip, of a number of boats, complete with flying birds. Maximum resolution of the original YUV file is 4CIF. SVC encoded resolutions include: QCIF, CIF and 4CIF.

4. *Soccer* - 10 second 300 frame low resolution fast moving media clip, of a number of people playing soccer. Maximum resolution of the original YUV file is 4CIF. SVC encoded resolutions include: QCIF, CIF and 4CIF.

5. *Sintel* - 52 second 1248 frame high resolution slow/fast moving media clip, of a trailer for an animated movie. Maximum resolution of the original YUV file is 1920x1080p (*1080p*). The resolution of the Sintel clips shall be cropped to provide consistency with the other clip types, also known as *rescaling*, while maintaining the longer clip duration, thus the SVC encoded resolutions include: QCIF, CIF and 4CIF.

   ffmpeg [129], an open-source multimedia framework was utilised to rescale the Sintel YUV files to lower resolutions. A sample of the underlying commands utilised are illustrated below. A helpful site for ffmpeg code snippets is [130].

   1. Command used to crop Sintel from 1280x720p to 704x576, while maintaing the entire 52 seconds of the clip

      ffmpegv1.2 -s 1280x720 -i sintel_trailer-1280x720px24fpsx52sec.yuv -vf crop=704:576:288:72 -vcodec rawvideo sintel_trailer-704x576x24fpsx52sec.yuv

      A. *-s* - resolution size

      B. *-i* - input file

      C. *-vf crop w:h:x:y* - crop to size w:h, starting at x:y co-ordinates in original media file

      D. *-vcodec rawvideo* - used to output raw YUV

   2. Command used to resize Sintel from 704x576 in item 1 to 352x288, same for 176x144

      ffmpegv1.2 -s 704x576 -i sintel_trailer-704x576x24fpsx52sec.yuv -s 352x288 sintel_trailer-352x288x24fpsx52sec.yuv

For subsequent steps in our methodology, we shall only use one of the clip types as an example of the input and output generated from the encoding of an original YUV file.

3.4.2 Encoding

This section encompasses the encoding process and details the decisions which are considered during scalable stream conversion from YUV to SVC. Stream scalability can be imposed in three directions: temporal (*resolution*), spatial (*frame*) and fidelity (*quality*). Significant consideration is given during the selection of the scalable encoding scheme options. The options to consider include: how many distinct resolutions will

the stream contain, how may quality layers per resolution, and will the stream contain a variable frame rate. The answer to each of these questions will determine the initial encoding and consequently the achievable quality, transmission cost, susceptibility to network loss and the benefits to, or restrictions imposed by, the transmission medium. Common practice would indicate that the diverse requirements of the clients will denote the initial encoding, but there are implementations where the initial encoding may be based on a default encoding selection.

In our evaluations, and for ease of comprehension, a three resolution encoding is utilised, namely: 176 * 144 (QCIF), 352 * 288 (CIF), 704 * 576 (4CIF) and for each of these resolutions, we define a number of different bitrates or quality levels. For each stream we define a tuple, where the value per resolution equates to the number of quality levels at that resolution, which we call resolution allocation. We allocate the quality levels based on the following resolution schema: (QCIF, CIF, 4CIF). An example would be: (1, 2, 5) which equates to one quality level at resolution QCIF, two quality levels at resolution CIF and five quality levels at resolution 4CIF. In Section 3.4.2.1 we will consider four different eight-layer encodings, based on a varying number of resolutions, quality levels per resolution and maximum achievable stream quality (measured using the popular PSNR [71], a pixel difference correlation between the reconstructed media stream and the original stream, typically measured in decibels (dB)):

1. (1, 2, 5) - encoding at a maximum PSNR of 38.9
2. (1, 3, 4) - encoding at a maximum PSNR of 38.5
3. (2, 3, 3) - encoding at a maximum PSNR of 38.5
4. (2, 3, 3) - encoding at a maximum PSNR of 36.8

To create the SVC encoded video files, we use "JSVM (Joint Scalable Video Model) [131] software from the Scalable Video Coding (SVC) project of the Joint Video Team (JVT) of the ISO/IEC Moving Pictures Experts Group (MPEG) [132] and the ITU-T Video Coding Experts Group (VCEG) [133]". JSVM is an open source project that is widely-used by the research community and is still under continued development. It is written in C++ and is provided as source code. JSVM contains libraries for encoding/decoding of AVC and SVC streams, for transcoding from SVC to AVC, for scaling videos from one resolution to another resolution, for extracting and decoding specific layers from an SVC sub stream, and for controlling bitrate generation, to name but a few.

In JSVM a maximum of three layers can be encoded using Coarse-Grained Scalability (CGS). While Medium-Grained Scalability (MGS) provides for an increase to eight in the number of layers that can be encoded. Based on the findings in Section 3.4.2.1 and the CGS limitation in JSVM, in our evaluations all SVC streams will be MGS and shall be encoded with eight layer and will have a resolution allocation of

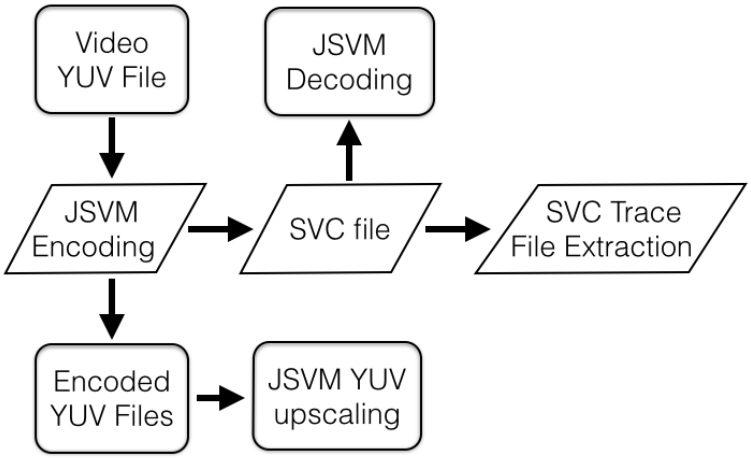


Figure 3.10: Overview of the encoding process, with respect to the JSVM libraries used

(2,3,3), *i.e.* two fidelity levels in the lowest resolution and three fidelity levels in each of the higher resolutions.

In our encoding settings we are concerned with only four of the JSVM libraries. Figure 3.10 illustrates the relationship of these libraries and their respective outputs. We begin with the JSVM encoding library:

1. H264AVCEncoderLibTestStatic - This library is used to create the SVC file, output_file.264, based on *encoder* and *layer* text-based configuration files. The encoder config file contains information such as output file name, the frame rate of the file, the GOP value, the *intraperiod i.e.* the number of frames between I or P frames, and the number of layers to be encoded. For each layer specified within the encoder config file, a layer config file must be defined. The layer config files contain information such as resolution width and height, frame rate in and out, the original YUV file at the specific resolution and the Quantisation parameter (QP). The QP defines the overall fidelity of this specific layer. The higher the QP, the lower the fidelity. The lower the QP, the higher the transmission cost, or bitrate of this layer. The encoding process creates two by-products and these are:

    1. For every layer defined in the SVC stream, a YUV video file of the specified resolution and quality is created by JSVM. These YUV files are used later in the evaluation section to determine the variation in quality with respect to the decoded files which were transmitted over the lossy network.

    2. The trace data from this process is stored in a file called "sdout.txt". The following is an example of the output. It can be seen that for each layer per frame, the frame type, the DTQ (dependency_id, temporal_id, quality_id) values, QP value, PSNR values for Y, U and V and total bit rate is provided.

```
frame_type   DTQ   QP_value   Y_PSNR    U_PSNR    V_PSNR    total_bitrate
```

```
AU     0: IK   T0 L0 Q0   QP 32   Y 33.1791  U 41.2554  V 42.2076     18288 bit
       0: IK   T0 L0 Q1   QP 26   Y 37.1582  U 42.7726  V 43.9366     17352 bit
       0: IK   T0 L1 Q0   QP 31   Y 33.8355  U 41.6546  V 43.2465     61080 bit
       0: IK   T0 L1 Q1   QP 28   Y 35.8387  U 42.1569  V 43.9621     44688 bit
       0: IK   T0 L1 Q2   QP 26   Y 37.0574  U 42.8566  V 44.4697     37384 bit
       0: IK   T0 L2 Q0   QP 33   Y 33.5901  U 40.9129  V 43.5914    136048 bit
       0: IK   T0 L2 Q1   QP 30   Y 35.2541  U 42.0130  V 44.3455    110384 bit
       0: IK   T0 L2 Q2   QP 28   Y 36.5230  U 42.4217  V 44.6180    109720 bit
```

2. BitStreamExtractorStatic - This library provides a means of extracting the bit-streams for each of the encoded layers and from this data we get two trace files: "layerRate.txt" and "layerTrace.txt". In "layerRate.txt" we can determine for each layer: the resolution, the frame rate, the bitrate, the SVC layer hierarchy, *i.e.* which layer is dependent on which layer based on stream scalability as outlined earlier in this section and from "layerTrace.txt" we receive the individual byte cost for each layer per frame. The following output snippet from "layerRate.txt" gives an example of the information provided by this library. The only heading that may not be immediately clear is the JSVM defined **DTQ**, which equates to three degrees of scalability: resolution (**D**ependency ID), frame (**T**emporal level) and PSNR (**Q**uality level). MinBitrate is defined as the minimum bitrate required to view the lowest layer of each resolution and is cumulative for higher resolutions.

Layer	Resolution	Framerate	Bitrate	MinBitrate	DTQ
0	176x144	30.0000	521.10	521.10	(0,0,0)
1	176x144	30.0000	1024.30		(0,0,1)
2	352x288	30.0000	2648.00	2144.80	(1,0,0)
3	352x288	30.0000	3914.00		(1,0,1)
4	352x288	30.0000	5006.00		(1,0,2)
5	704x576	30.0000	8664.00	5802.80	(2,0,0)
6	704x576	30.0000	11679.00		(2,0,1)
7	704x576	30.0000	14792.00		(2,0,2)

The next output snippet from "layerTrace.txt" illustrates the byte cost and DTQ values per layer, thus permitting calculations of per layer and total transmission cost and the layer dependency per frame. The start-pos is the starting point of the specific layer within the bit stream. As previously mentioned it is this field and the Length header that can be used by Course-Grained Scalability (CGS) and Medium-Grained Scalability (MGS) for sub stream extraction and decoding. The packet-type defines if the packet is a data or control NALU. In the JSVM manual, the individual layers are defined as packets, and as such we assume these packets to be NALU for a given Access Unit (AU).

Start-Pos.	Length	LId	TId	QId	Packet-Type	Discardable	Truncatable
0x0000018d	18	0	0	0	SliceData	No	No
0x0000019f	2277	0	0	0	SliceData	No	No
0x00000a84	2169	0	0	1	SliceData	Yes	No
0x000012fd	2534	1	0	0	SliceData	No	No
0x000030d0	5586	1	0	1	SliceData	Yes	No
0x000046a2	4673	1	0	2	SliceData	Yes	No
0x000058e3	17006	2	0	0	SliceData	No	No

```
0x00009b51   13798    2    0    1     SliceData         Yes          No
0x0000d137   13715    2    0    2     SliceData         Yes          No
```

Each column is defined as the following:

1. Start-Pos. - A hexadecimal value for the start positions (in units of bytes) of this packet inside the bit-stream
2. Length - The length of the packet (in units of bytes)
3. LId - Dependency ID, based on different resolutions
4. TId - Temporal ID, based on different frame rates
5. QId - Quality ID, based on different quality values
6. Packet-Type - The type of packet, normally either StreamHeader, ParameterSet or SliceData
7. Discardable - Defined as not required for minimum quality decoding for each resolution
8. Truncatable - Defined if the packet can be made smaller (truncatable packets are not supported in SVC)

This information provides us with a means of introducing loss to the stream, either by dropping or deleting complete layers or by determining the number of packets transmitted per layer and then dropping a specified percentage of these packets.

3. H264AVCDecoderLibTestStatic - This library utilises the JSVM created SVC file, output_file.264, and creates a YUV file based on the highest quality layer within the SVC file. The output is saved to a decoder trace file, "decoderOutput.txt", which contains the associated information for each AU (frame) as well as the specific per layer information. The following is the output for a single frame from an eight layer single GOP stream:

```
---------- new ACCESS UNIT ----------
NON-VCL: SEI NAL UNIT [message(s): 10]
Frame    0 ( LId 0, TL 0, QL 0, AVC-I, BId-1, AP 0, QP 32 )
Frame    0 ( LId 0, TL 0, QL 1, SVC-I, BId 0, AP 0, QP 26 )
Frame    0 ( LId 1, TL 0, QL 0, SVC-I, BId 1, AP 1, QP 31 )
Frame    0 ( LId 1, TL 0, QL 1, SVC-I, BId16, AP 0, QP 28 )
Frame    0 ( LId 1, TL 0, QL 2, SVC-I, BId17, AP 0, QP 26 )
Frame    0 ( LId 2, TL 0, QL 0, SVC-I, BId18, AP 1, QP 33 )
Frame    0 ( LId 2, TL 0, QL 1, SVC-I, BId32, AP 0, QP 30 )
Frame    0 ( LId 2, TL 0, QL 2, SVC-I, BId33, AP 0, QP 28 )
```

Each row is defined as the following:

1. Frame or non-VCL data
2. Frame number

3. LId - Dependency ID, based on different resolutions
4. TId - Temporal ID, based on different frame rates
5. QId - Quality ID, based on different quality values
6. Player and frame type, this example highlights AVC for the base layer and SVC for all other layer, as well as all frames are I frames
7. BId - The BaseLine ID of the previous layer on which this layer is dependent
8. AP - Specifies the use of inter-layer prediction, based on the ILMotionPred value in the JSVM encoding file. A zero value denotes no prediction, while a one implies prediction with or without a macro-block partition.
9. QP - Quantisation Parameter - achievable quality of this specific layer

It is important to note that the decoding process only works for complete unaltered bitstreams. Bitstreams which are corrupt due to lost packets or lost layers, known as modified streams, cannot be decoded by JSVM. Early versions of JSVM contained error concealment options which modified the reconstructed YUV to account for loss. But after version 9.8 of JSVM, the error concealment option was discontinued. Even the old tool was not well suited to decoding most damaged bitstreams. Due to this fact and because we were working with modified bitstreams throughout our research (due to the defined percentage of loss uncounted during transmission), we were never able to use JSVM to decode our modified streams. We shall explain later in the evaluation section how we were able to determine the viewable quality of the received bitstreams.

4. DownConvertStatic - As stated, a by-product of the initial JSVM SVC encoding is the creation of individual YUV files for each quality layer of the original SVC stream. As PSNR, or quality, calculations require that the resolution of evaluated streams have identical resolutions, this library is utilised to up-sample, or upscale, the lower resolution streams to the maximum resolution in the encoded stream. The default JSVM method for up-sampling is a normative up sampling method designed to support the Extended Spatial Scalability (*ESS*) [134] and is based on a set of integer-based 4-taps filters derived from the Lanczos-3 filter [135]. ESS offers additional spatial scalability by providing options such as cropping (viewing only a portion of the original image) and up-sampling. Lanczos-3 filter is typically used to smooth the variation between the original image and sampled image by increasing the sampling rate or shifting the sampling interval.

Appendix A provides an in-depth overview of JSVM encoding and subsequent output with respect to a GOP value of 1 and a GOP value of 8, while also outlining the decision process used to determine the frame hierarchy selected for our simulation tests.

3.4.2.1 JSVM Results for Variation in Layer Allocation per Resolution

Table 3.6: Key metrics per adaptive scheme for each of the selected encodings, including transmission byte cost

Scheme	GOP	max PSNR	Encoding Time	SVC (byte)	MDC (byte)
Resolution (1, 2, 5)	1	38.9	15.58 minutes	9,870,321	21,247,605
Resolution (1, 2, 5)	8	36.6	202.32 minutes	3,613,019	8,138,744
Resolution (1, 3, 4)	1	38.5	15.72 minutes	8,845,840	19,400,400
Resolution (2, 3, 3)	1	38.5	13.90 minutes	8,491,556	16,659,188
Resolution (2, 3, 3)	1	36.8	13.63 minutes	6,018,201	13,906,335

JSVM v9.19 [131] is utilised to encode a widely used ten second sample clip of astronauts walking in a corridor, called crew.yuv, into eight-layer SVC streams, from which the transmission trace data was extracted. JSVM contains a quantisation parameter (QP), which provides a simple mechanism to encode the stream with varying bitrates, thus providing differing levels of quality per layer. In our evaluation, we encode each scheme with a group of picture (GOP) rate of 1. As GOP is utilised to reduce transmission cost by increasing inter frame dependency, and inter frame dependency reduces stream quality as loss increases; due to the loss of key frames. A GOP of one would provide an awareness of how loss affects quality, without the increased quality degradation provided by a higher GOP.

Table 3.6, provides a comparison between the different encodings and the relevant transmission cost per adaptive scheme, while Figure 3.11 provides a graphical representation of the transmission costs. Note, the specifications of the encoding machine are: 64-bit Windows 7 professional, 8GB ram, Intel Core2 Quad 2.66Hz. To provide a comparison of GOP, one of the schemes is also encoded with a GOP value of eight to view the affects on transmission cost and encoding time. Note that the GOP is the only value altered in the encoding. It can be seen that the increase in GOP, mandates a reduction in achievable quality and transmission cost, while drastically increasing encoding time.

3.5 Encoding Efficiency

In this section the data in Table 3.6 is utilised to determine how the initial encoding decision affects the video, as per the following:

1. A reduction in achievable quality, maximum PSNR, mandates a reduction in transmission cost.
2. The addition of FEC resilience to the resilient MDC scheme increases transmission cost.

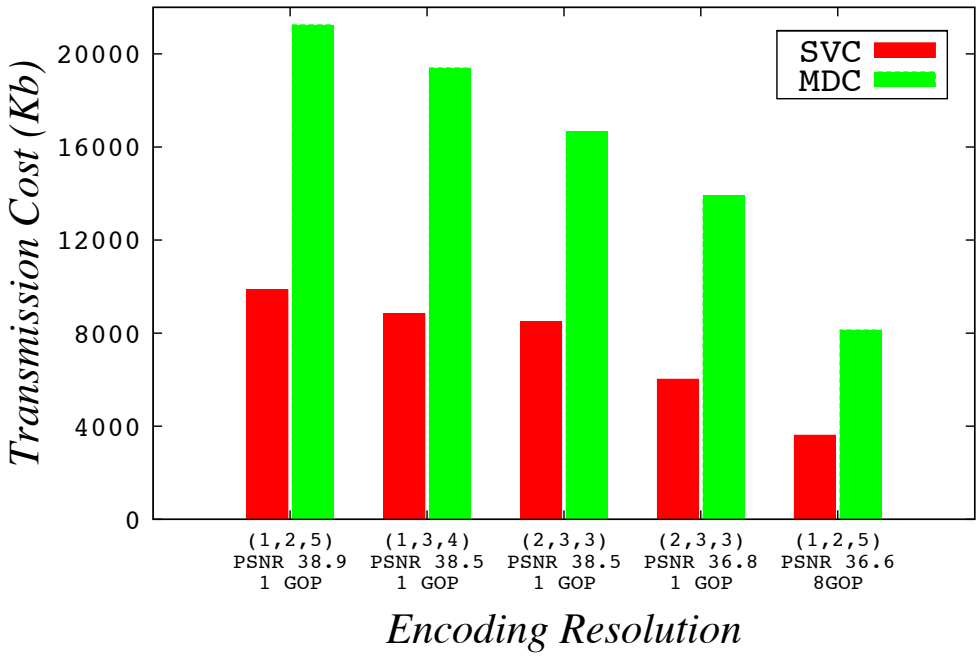


Figure 3.11: Transmission cost of encoding at different PSNR and with different layer quality per resolution and GOP

3. The selection of allocated resolutions can determine transmission cost, as outlined in (1, 3, 4) and (2, 3, 3). While both provide the same quality, 38.5, and similar SVC transmission cost, (1, 3, 4) commands a differing transmission cost for MDC, which highlights that the expense of resilience is based on the byte size of the underlying layers and more importantly on the byte size of the lower layers. Such that, a scheme of consistent bit-rates per layer would offer balance to the resilient schemes.

4. Note that the encoding times offer a means of comparing encoding complexity.

5. It is also worth noting that creating scaling video reduces the max available stream quality [136] and PSNR values of less than 20dB correspond to inferior video quality [71].

3.6 Transmission and Loss

The objective of this section is to determine how loss affects quality for each of the selected adaptive schemes. So, for ease of comprehension, we provide a UDP implementation of scalable streaming in a lossy network, such that loss is determined without the benefits of datagram retransmission. Mobile network datagram loss is a consistent occurrence due to interference, while the percentage of loss is irregular. This leads to causality similar to the *chicken* and the *egg* paradigm, *e.g.* which comes first loss or resilience.

Table 3.7: PSNR 38.9: Crew comparison results as per the eight layer JSVM streaming trace data, with a resolution allocation of (1, 2, 5) and GOP of 8

Scheme	SVC	MDC
Transmission (bytes)	9,870,321	21,247,605
% Byte size wrt SVC	100%	216%
Layer/Description $\overline{Size}$	4,112	8,853
# Layer/Description	8	8
# Datagrams (D_g)	7,741	22,848
$\overline{PSNR}$ at 10% D_g loss	32.45 dB	38.45 dB

As an example of the encoding schemes, Tables 3.7 highlights the transmission byte cost, relevant size of the transmission compared to SVC, average size of each layer/description, number of layers/description per frame, the number of Ethernet MTU datagrams required by the scheme and the average PSNR of stream (1, 2, 5) when confronted by 10% datagram loss, all of these details were determined by utilising the JSVM trace data. The other schemes are consistent with these results.

It can be seen that SVC yields the lowest PSNR, due to the absence of error resilience and its prioritisation hierarchy. This highlights that while SVC transmits the lowest number of datagrams, it produces the highest percentage of transmitted loss, *e.g.* layer data that is transmitted but is un-decodable at the device due to lower layer loss. The level of increased error resiliency offered by MDC increases the achievable PSNR values for MDC by approximately 6 dB.

It can be recognised that while resilience increases transmission cost, it also reduces transmitted loss. Such that for bandwidth-constrained or lossy networks, a mechanism is required that provides a balanced approach to cost and loss.

3.6.1 Simulation

Prior to simulating loss in our SVC bitstream, we need to first take the trace files generated in the Encoding section and perform some additional processing. Figure 3.12 provides an overview of the simulation undertaken and the interaction between the different components used. We begin our additional processing by using SVEF [137]: an open-source experimental evaluation framework for H.264 scalable video streaming. One component contained within SVEF is f-nstamp. f-nstamp takes the "decoderOutput.txt" trace file and the "layerTrace.txt" trace file as input and creates "layerTrace-frameno.txt" as output. "layerTrace-frameno.txt" contains the information contained within "layerTrace.txt" and adds two additional columns. The first additional column will be populated with the frame number corresponding to the NALU associated to that line. These additional frame numbers are determined based on the input of "decoder-

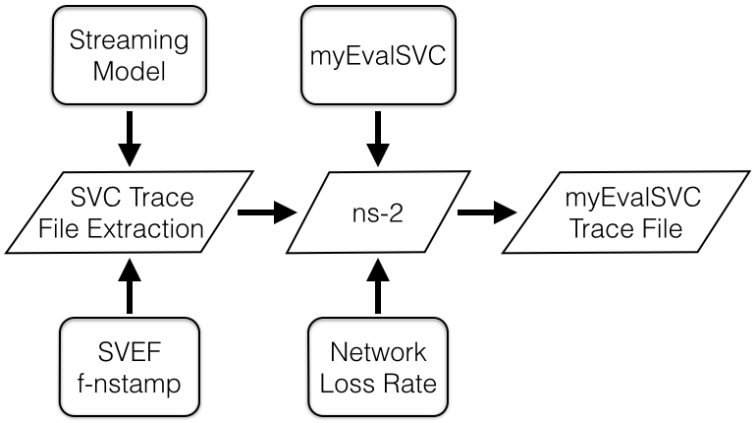


Figure 3.12: Overview of the simulation process, from SVC trace file input to ns-2 trace file output

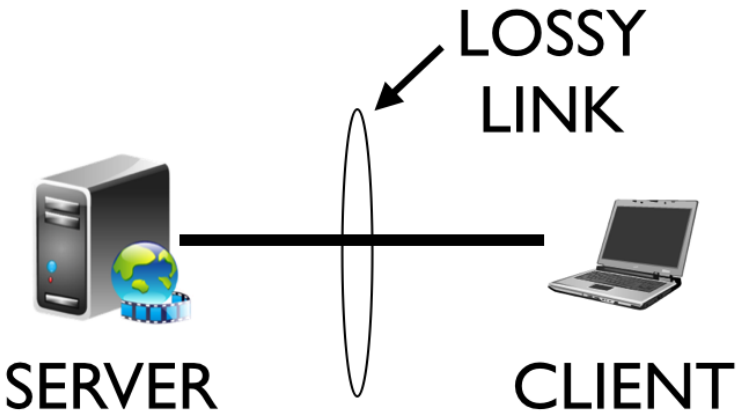


Figure 3.13: A two-node, server/client, model is utilised for the simulated ns-2 topology

Output.txt". The second column contains the time at which the NALU was transmitted from the server. This second column permits selective dropping of NALUs which arrive at the client after the subsequent frame has been decoded. In our evaluations we do not use this second column as we evaluate based on defined percentage levels of packet loss and including these late packets may skew our results by increasing the percentage of packets beyond our defined percentage. We now have a trace file which contains the per layer bitrate for every frame in the encoded stream and are ready to simulate stream transmission.

Start-Pos.	Length	LId	TId	QId	Packet-Type	Discardable	Truncatable	frame_num	trans_time
0x0000018d	18	0	0	0	SliceData	No	No	0	0
0x0000019f	2277	0	0	0	SliceData	No	No	0	0
0x00000a84	2169	0	0	1	SliceData	Yes	No	0	0
0x000012fd	2534	1	0	0	SliceData	No	No	0	0
0x000030d0	5586	1	0	1	SliceData	Yes	No	0	0
0x000046a2	4673	1	0	2	SliceData	Yes	No	0	0
0x000058e3	17006	2	0	0	SliceData	No	No	0	0
0x00009b51	13798	2	0	1	SliceData	Yes	No	0	0
0x0000d137	13715	2	0	2	SliceData	Yes	No	0	0

Network Simulator 2 (*ns-2*) [138], a well-known and widely-used network simula-

tion tool which supports TCP, routing and multicast protocols over wired and wireless networks is utilised to simulate a lossy channel. As illustrated in Figure 3.13, a two-node, server/client, model is utilised for the simulated topology in which we vary the packet error rate, μ, from 1% to 10% to test the streaming performance of different schemes over lossy links. The goal of our work is to determine for a given loss rate, the maximum achievable quality for a selected streaming model. Thus a one-to-one model is sufficient for our needs, as a defined level of network loss can be imposed on the individual link.

In our simulations we limit the maximum loss rate to 10% as: 1) PSNR plots containing loss rates from 1% to 10% demonstrate the variance in achievable quality as packet loss increases, thus illustrating the benefits or drawbacks of a streaming model at relatively low loss rates. Higher loss rates would only further exacerbate the achievable quality of a given model. 2) For description-based models with statically defined FEC levels, such as for MDC, their is a packet loss threshold where the level of applied FEC is unable to cope with a given loss rate. Thus, there is a tipping point where the benefits of description-based models are negated by increased loss rates. For our streaming models, we adapt the FEC levels to cope with a given loss rates, so as to reduce overall transmission cost. Thus, we reduce the tipping point of the packet loss threshold to a point within the simulated loss rates. For levels of packet loss greater than 10%, the same mechanism is used to increase the FEC levels but this can lead to increases in overall transmission cost which is counter-productive to the goals of our research.

We employ the widely-used ns-2 Errormodel to define a total packet error rate with a uniform distribution. This defines that the average packet loss shall be equal to μ, but does not mandate that the individual frame loss rate shall also be equal to μ, thus permitting bursty loss during simulation. The ns-2 Errormodel simulates link-level error or losses by marking the packet's error flag or dumping the packet to a drop target. Typically the Errormodel is utilised to generate a simple model based on a defined packet error rate, as used in our simulations, but can be used for more complicated statistical and empirical models. The unit of error can be defined based on packet, bits, or time-based. In our simulation, UDP is the transport protocol. Per distinct μ packet error rate, we simulate 16 experiments for each streaming model. As the effects of the Errormodel differ for each experiment, this provides us with a means of averaging the effects of loss over multiple runs. As ns-2 does not contain an inherent mechanism for simulating trace-based streaming models *i.e.* JSVM trace files, we use myEvalSVC for this functionality.

myEvalSVC [139], an open source tool for evaluating JSVM stream trace data in ns-2, presents a means of dynamically determining bitrates based on the JSVM trace data and simulating real-time packetisation over a lossy network in ns-2. myEvalSVC

adds extra functionality to ns-2 by providing additional traffic models, a new Sink node and numerous new variables which permit packetisation of the various layered bitrates.

In our evaluation myEvalSVC is utilised to create streaming packetisation models for each of the scalable schemes. myEvalSVC mandates that the JSVM encodings utilised are specific in both scalability (resolution, frame, quality) and quantity of layers. To operate with our encoding of an eight layer streaming model, *i.e.* SVC, MDC and the new streaming models proposed in this and later chapters, minor modifications are made to the original myEvalSVC scripts. Due to the multi-datagram requirements of each of the streaming schemes and possible out of order delivery of datagrams at the device, the modified myEvalSVC scripts provide mechanisms so as to calculate the maximum achievable stream quality at the device. Prior to transmission, a transmission time, time sent from server, is added to each frame. In this example we assume transmission begins 500ms after stream request is received, but this is only used to reflect transmission time in the ns-2 simulation file. It can be seen that the 4 highest layers are transmitted in a different time period that the lower layers, again just an example of the variation in time that can be allocated to different layers/frames.

Start-Pos.	Length	LId	TId	QId	Packet-Type	Discardable	Truncatable	frame_num	trans_time
0x0000018d	18	0	0	0	SliceData	No	No	0	500
0x0000019f	2277	0	0	0	SliceData	No	No	0	500
0x00000a84	2169	0	0	1	SliceData	Yes	No	0	500
0x000012fd	2534	1	0	0	SliceData	No	No	0	500
0x000030d0	5586	1	0	1	SliceData	Yes	No	0	500
0x000046a2	4673	1	0	2	SliceData	Yes	No	0	533
0x000058e3	17006	2	0	0	SliceData	No	No	0	533
0x00009b51	13798	2	0	1	SliceData	Yes	No	0	533
0x0000d137	13715	2	0	2	SliceData	Yes	No	0	533

This file is then modified to provide a myEvalSVC compliant file structure. The first column is the time, the layer transmission cost, followed by LId, TId and QId and the last column is the frame number. Also note how the header information and the base layer (both with LId, TId and QId of 0,0,0) are added together:

time	trans_cost	LId	TId	QId	frame_num
0.000000	2295	0	0	0	0
0.000000	2169	0	0	1	0
0.000000	2534	1	0	0	0
0.000000	5586	1	0	1	0
0.033333	4673	1	0	2	0
0.033333	17006	2	0	0	0
0.033333	13798	2	0	1	0
0.033333	13715	2	0	2	0

Comprehensive simulations are performed and once complete, the myEvalSVC trace files are passed to the Evaluation section to determine the maximum, per-frame, stream quality at the client. The following is an example of the myEvalSVC trace file for the first three layers only. The columns are the time of receipt at the client, frame number, packet byte size (we assume a maximum packet data size of 1,440 bytes, thus allocating 60 bytes for header information), LId, TId, QId, number of packets per frame

and initial transmission time. It can be seen from this example that packet 3 is missing and was lost during transmission, thus Layer two can not be decoded which will lead to a decrease in viewable quality for this frame:

receive_time	frame_num	packet_size	LId	TId	QId	num_packet	trans_time
0.512345	0	1440	0	0	0	0	0.500000
0.517212	0	855	0	0	0	1	0.500000
0.515487	0	1440	0	0	1	2	0.500000
0.513274	0	1440	1	0	0	4	0.500000
0.516239	0	1094	1	0	0	5	0.500000

As we have seen from the myEvalSVC trace file, data were lost from layer two and so a reduced number of layers will be decodable. Assuming the scalable hierarchy as outlined by LId, TId and QId, and assuming no further data are lost from higher layers, then only the base layer, layer 3 and layer 6 are decodable. These specific layers are only dependent on Lld, while all other layers are also dependent on Qld, which was increased by layer two. But if we assume layer N is only viewable if all preceding 1 to $N-1$ layers are decodable, then only the base layer is decodable. The determination of the highest layer that can be decoded is an important concept to consider but which of these two results, the base layer or layer 6, shall we use in our Trace Analysis. The answer is dependent on the initial selection of either Course-Grained Scalability (CGS) or Medium-Grained Scalability (MGS) during encoding. With CGS the scalable hierarchy as outlined by LId, TId and QId mandates the layer dependency required during decoding, while for MGS layer N is only viewable if all preceding 1 to $N-1$ layers are decodable. Prior to our final decision on which encoding option to utilise for our evaluation, let us first reconsider the underlying benefits of SVC. SVC can be viewed as benefiting a single user by providing graceful degradation of quality while also reducing transmission cost when multiple users are viewing the same stream. For a single user the presence of all lower layers provide the graceful degradation required and yields consistency of quality with little variation in decodable layer value over time. Also, in a large real-world deployment, the overall quality of a scalable IPTV stream will be governed by the maximum viewable quality selected, but within the stream different users will decode with dissimilar layers values.

Based on our eight layer example and assuming CGS, the layers that require all lower layers to be available prior to decoding are the:

1. Base layer: this layer is not dependent on any layer and as such is decodable once received.
2. Layer two: requires the base layer prior to decoding.
3. Layer four: as the lowest resolution contains two quality levels, the second quality level in the second resolution can only be decoded once all lower resolutions and lower quality levels have been received.

4. Layer five: as the second resolution contains a third quality level, all lower lower resolutions and lower quality levels must be received prior to decoding.

5. Layer eight: this is the highest layer and by default all lower layers must be available prior to decoding.

While the remaining layers require only a subset of layers to be available prior to decoding:

1. Layer three: as the lowest quality level in the second resolution, only the lowest quality level in the lowest resolution is required prior to decoding. Only the base layer is required prior to decoding.

2. Layer six: as the lowest quality level in the third resolution, the lowest quality levels in all lower resolutions are required prior to decoding. Only the base layer and layer three are required before decoding.

3. Layer seven: as the lowest resolution contains two quality levels, the second quality level in the third resolution can be decoded once the lowest two quality levels for all the lower resolutions have been received. The base layer, layer two, layer three, layer four and layer six are required prior to decoding. Only layer 5 is not required.

As previously stated, based on the findings in Section 3.4.2.1, the CGS layer quantity limitation in JSVM and the benefits provided to single and multiple users by MGS, in our "Trace Analysis" all SVC streams will be MGS encoded and as such layer N is only viewable if all preceding 1 to $N-1$ layers are decodable. Thus in the example outlined above only the base layer is decodable.

3.6.2 Trace Analysis

In this section, we view the steps required to analyse our "myEvalSVC trace file" and determine the effects of network loss on viewable quality. Figure 3.14 provides a high-level overview of the steps taken during trace analysis. We begin by performing some additional processing of the myEvalSVC trace file, which will initially sum the byte cost of each packet per layer, per frame, and then rebuild the "layerTrace-frameno.txt" as "layerTrace-received.txt" but will update the byte-cost length of the packets (column 2) and the time the last packet was received for this layer (column 10):

Start-Pos.	Length	LId	TId	QId	Packet-Type	Discardable	Truncatable	frame_num	trans_time
0x0000018d	18	0	0	0	SliceData	No	No	0	517
0x0000019f	2277	0	0	0	SliceData	No	No	0	517
0x00000a84	1440	0	0	1	SliceData	Yes	No	0	515
0x000012fd	2534	1	0	0	SliceData	No	No	0	516
0x000030d0	5586	1	0	1	SliceData	Yes	No	0	515
0x000046a2	4673	1	0	2	SliceData	Yes	No	0	516
0x000058e3	17006	2	0	0	SliceData	No	No	0	518
0x00009b51	13798	2	0	1	SliceData	Yes	No	0	521

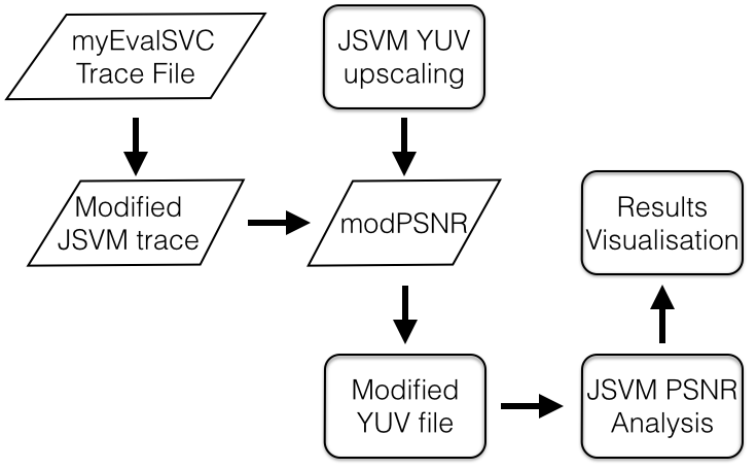


Figure 3.14: Overview of the trace analysis process, from myEvalSVC trace file input to JSVM PSNR analysis and subsequent result visualisation (provided in the next section)

```
0x0000d137   13715    2    0    2     SliceData         Yes          No          0        519
```

As noted, even though the "layerTrace-received.txt" trace file is now a JSVM compatible bitstream, albeit with modified or missing data for some layers, JSVM is unable to decode this file. However, JSVM does contain a library for measuring PSNR between two raw video sequences, called "PSNRStatic". To provide the bridge between the "layerTrace-received.txt" trace file and the PSNR measurements required to illustrate the variation in quality provided by the relevant streaming models, we created a new program called "modPSNR". "modPSNR" utilises the code base of the JSVM program "PSNRStatic" and adds additional functionality so as to create modified YUV files, which are then measured against the original YUV, so as to determine the deviation in quality, or PSNR, of the modified YUV files. The "layerTrace-received.txt" trace files are pre-analysed to determine the maximum, per-frame, stream quality at the client. Each trace is then saved as an achievable quality (AQ) trace file for each streaming model. Using our example, this would determine the frame 0 has a maximum viewable quality of layer 5. If each of the layers are dependent on the previous layer, then the maximum viewable quality would be the base layer (layer 1).

The AQ trace files are utilised to 1) to provide a means of illustrating the transition in frame quality over time and 2) to create the modified YUV files, based on the maximum stream quality per frame, from the original YUV files. In our results, for each model we determine the maximum stream quality per frame based on the highest layer that contains no packet loss and can be fully decoded, thus containing no impairments that are visually observed. It can be surmised, that in a real world system advanced error concealment mechanisms would be used to reduce the visual effects of the lost data. Most advanced techniques are proprietary and as such are unknown. The following examples are possible options that could be used and these include:

1. Simple frame replication: when the amount of data lost in a frame negates the use of the frame, then simply replicating the previous frame may be beneficial to overall viewable quality. This option is dependent on frame rate, frame type (I, P or B), video type and user acceptance of replicated frame data.
2. More advanced replication of individual pixel data contained within a frame. If only partial frame data are lost, then by taking two adjacent frames and determining the changes in frame, may increase viewable quality. This option is dependent on the similarity between adjacent frame. This would not work during scene changes in a clip.
3. Upscaling a selection of the frame pixels within a lower layer to account for the loss of a subset of higher layer pixels. The upscaled pixels may be overly visible, but this would be dependent on frame/clip type and well as the variation in resolution between lower and higher layer.
4. Retransmissions of the lost data. If there is sufficient time and bandwidth, then retransmission of the lost data may be of benefit.

We did not have access to these mechanisms and as such we decode based on the highest layer that contains no packet loss.

As previously stated, a by-product of the JSVM encoding process is the creation of a YUV file for each layer encoded in the stream, we shall call these layer_YUV files. "modPSNR" utilises these layer_YUV files and the AQ trace file to create a modified YUV file. As noted PSNR requires that the resolution of the modified YUV file is the same as the original YUV file, thus we up-sample all lower resolution layer_YUV to the maximum resolution of the encoded stream, *i.e.* in our examples we up-sample from 176x144 (*QCIF*) and 352x288 (*CIF*) to 704x576 (*4CIF*). Some pixelation (upscaling of low quality resolutions thus creating noticeable square shaped single-colour display components on the screen) may occur when the resolution of the maximum achievable quality is less than the maximum viewable resolution.

"modPSNR" scans the AQ trace file and for each frame in the stream will determine the maximum viewable quality. "modPSNR" then extracts the frame of the evaluated quality from the correct layer_YUV file to create the modified YUV file. "modPSNR" supports basic error concealment by which non decodable frames are substituted by duplicating the previous frame. Should the initial one or more frames be undecodable, then the next decodable frame shall be duplicated as the initial frame(s). Part of our initial thesis goal is to view the effects of network loss on existing streaming models, i.e. SVC and MDC. In this regard, more advanced error concealment mechanisms would unfairly increase the viewable quality and consequently the PSNR value of these existing schemes. Thus basic error concealment provides for a more practical reflection of loss on viewable quality. Once the modified YUV has been created,

"modPSNR" will ascertain the PSNR value in comparison to the original YUV file using the JSVM "PSNRStatic" code base.

3.6.3 Metrics and Result Visualisation

Prior to describing how we visualise the viewable quality ascertained from the AQ trace file. We first need to consider what to visualise. As SVC is a layered schema, we need to consider how network loss affects the quality of each frame and illustrating layer variation over time will achieve this. The variation plot provides a coarse overview of viewable quality over time, *i.e.* plots with minor changes in variation provide consistent viewable quality, while large variation in quality especially with differing layer values per adjacent frames mandate visible degradation and thus poor viewable quality. Time is the important metric in this plot, as we view the changes in quality over the duration of the stream.

Variation plots with large transitions do not in themselves provide a sense of the overall viewable quality. A similar plot which determines the percentage of frames viewable for each layer value immediately provides a perception of the overall viewable quality of the stream. In this subsequent plot we are only concerned with the viewable percentage of each layer. We find that streams with large percentage values in the higher layers will mandate increased streaming quality, while conversely large percentage values in the lower layers will provide for reduced streaming quality.

Finally, it is important to view the overall achievable quality of the stream itself and this is achieved using PSNR. PSNR provides us a means of correlating the quality of our received YUV with that of the original YUV, thus illustrating the change in viewable quality over the entire clip.

Gnuplot [140] is a command-line graphical tool for visualising our empirical results. We use Gnuplot primarily to illustrate three main plots and in this section we illustrate an example of all three plots and explain each plot in detail. Gnuplot is used to:

1. Illustrate the variation in viewable quality over time. Figure 3.15 illustrates a ten-second example of the variation in viewable quality for an eight layer SVC encoding of city with a 10% packet loss rate as defined by the AQ trace file. Two items are important to note: 1) depending on where the data are lost, *i.e.* which layer, denotes the maximum viewable quality and 2) how the transitions that occur between adjacent frames can be quite large, illustrating a noticeable change in viewable quality over time. As previous stated, we perform sixteen runs for each loss ratio. The variation plot takes a snap shot of one of the experiments as an example of variation that can occur.

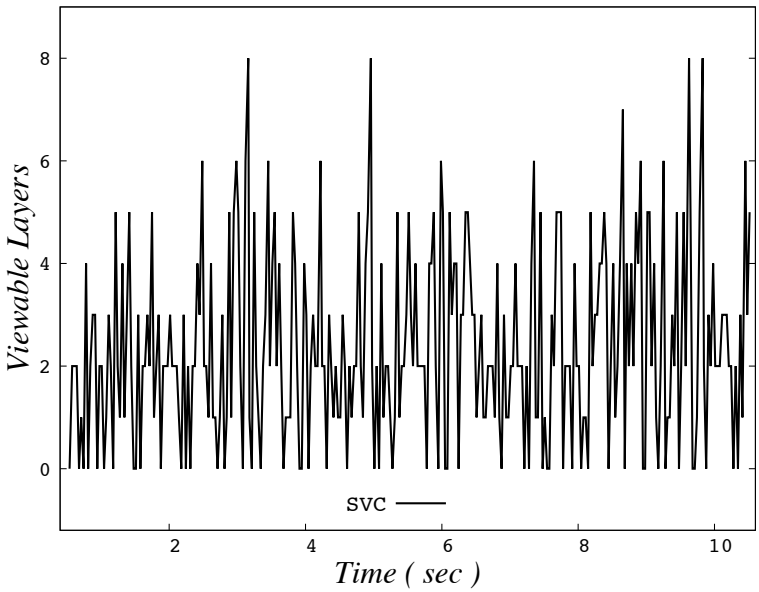


Figure 3.15: The variation in viewable quality for an SVC encoding of city transmitted over a network with a 10% loss rate

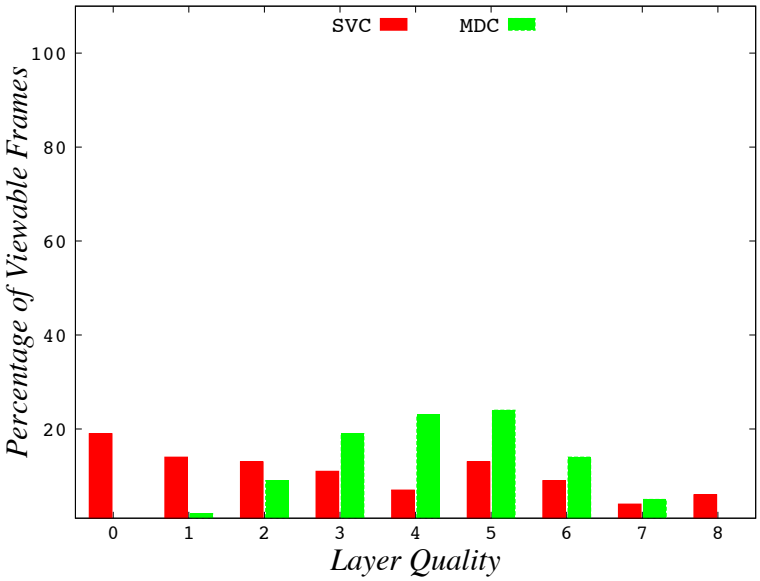


Figure 3.16: The percentage of viewable frames for each of the encoded layers for an eight layer SVC and MDC encoding of crew with a 10% packet loss rate

2. Display the percentage of viewable frames for each layer over the duration of the stream. Figure 3.16 displays the percentage of viewable frames for an eight layer SVC and MDC encoding of crew with a 10% packet loss rate. This plot takes the variation defined by the AQ trace file and determines the percentage of viewable frames for each of the layer quality levels, including for non decodable frames, denotes as layer zero. The percentage of viewable frames plot also takes a snap shot of one of the experiments to illustrate the percentage of viewable frames per layer.

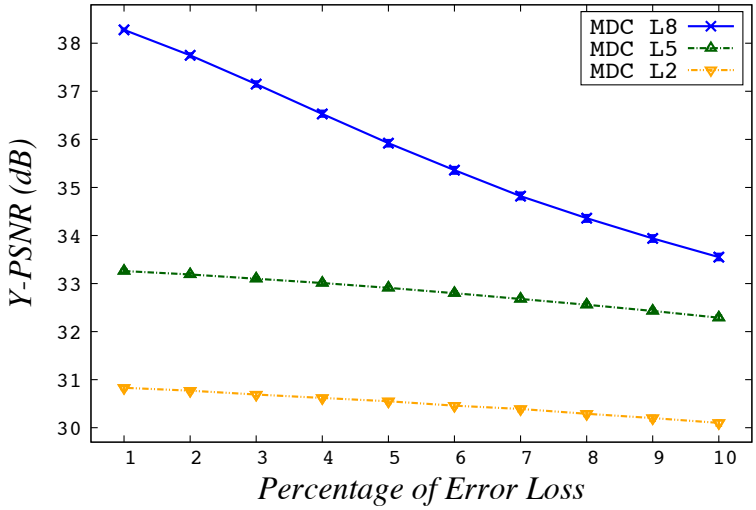


Figure 3.17: The variation in PSNR as network loss increases for three MDC streams

3. Show the variation in PSNR for different network loss rates. Figure 3.17 shows the maximum achievable mean Y-PSNR values, in dB, for three MDC streams. The MDC plots represent the highest quality layer for each resolution in the stream and how their respective PSNR values decrease as packet loss increases. The PSNR plot takes all 16 experiments and averages the PSNR for each plot with respect to the μ selected during simulation. 95% Confidence Interval error bars are added to show the variation in PSNR value that can occur. In some plots the variation in PSNR is so small, the error bars are imperceptible as they are obscured by the legend markers, *i.e.* the 'x', the triangle and the inverted triangle as shown in Figure 3.17.

3.6.4 Evaluation Methodology Conclusion

In this Section, we presented an overview of the methodology utilised to evaluate our research. We outlined our selection of clip type, encoding options, simulation design, trace analysis, and the motivation behind our result visualisation. This Section can be viewed as a precursor to the evaluation sections of later chapters. Should variations to the methodology outlined in this Section be required in later chapters, this shall be clearly outlined.

3.7 SDC Evaluation

In this section, we assess the performance of SDC and SDC-NC and compare it with both SVC and MDC. In our evaluation, we consider an eight layered stream transmitted over a lossy network with different loss rates. Figure 3.18 provides an overview of the adaptive streaming topology that our evaluation simulates. To provide increased

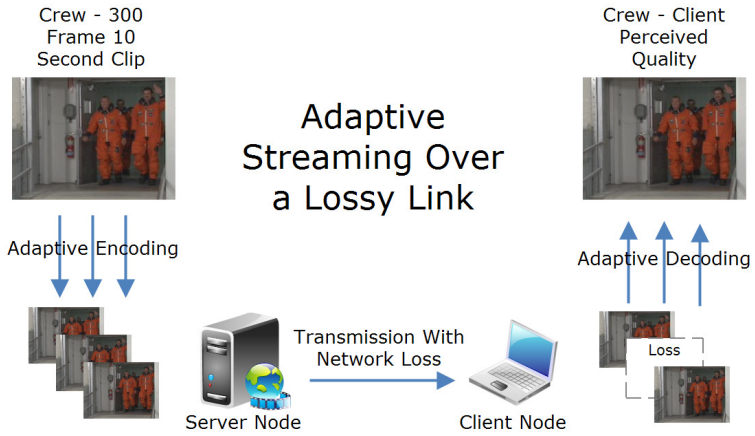


Figure 3.18: Overview of adaptive streaming topology used to determine bandwidth savings and PSNR.

achievable quality in our evaluation results we implement SDC with both packetisation options: section-based description packetisation (*SDP*) and Section Distribution (*SD*), as detailed in Section 3.3.1. SD is utilised by the D_c and D_r descriptions as these contain FEC resiliency to recover from loss, and the loss of a segment from all layers in a single description will not overly affect viewable quality. SDP is used by D_s as no FEC is allocated to the D_s. If SD is used by D_s, then a segment of each layer section in the D_s is allocated to a packet. If any packet for the D_s is lost, then all layer sections in the D_s are undecodable. Using SDP confines the loss of a packet to a specific layer section within the D_s, which reduces the effects of network loss.

Irrespective of bit-rate, a streaming model is defined such that the packets need to arrive within a specific timeframe, so as not to reduce the perceived quality of the stream. The simulation design implemented a layer ID and GOP ID, such that each packet is allocated to a specific layer/description in a specific GOP. This design allows the simulation to track the packets between server and client. Packets arrive at the client in the order they are sent from the server but if packets arrived out of order, then the GOP ID and layer ID are utilised to determine what is decodable by the client.

3.7.1 Experimental Results

Table 3.8 presents the percentage of viewing time at the different quality levels for the different streaming models at a packet loss rate of 10% for our eight layer sample. Table 3.9 outlines a summary of the mean layer value viewed over the duration of the clip, *i.e.* over all 300 frames, and the maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer value viewed, thus illustrating the the variation in quality that occurred during decoding. Note how all models were able to achieve layer 8 decoding. Due to packet loss SVC and MDC were unable to decode some frames, while SDC mandates minimum decoding to layers 3 and 4 dependent on the optimisation technique utilised. SDC

Table 3.8: Example of average percentage of viewing time at a specific layer for eight layers with 10% network loss

	SDC-NC	SDC	MDC	SVC
Layer 8	42%	51%	4%	10%
Layer 7	16%	15%	5%	4%
Layer 6	16%	18%	14%	9%
Layer 5	16%	9%	24%	13%
Layer 4	10%	7%	23%	7%
Layer 3	0%	0%	19%	11%
Layer 2	0%	0%	9%	13%
Layer 1	0%	0%	2%	14%
No Viewable Layer	0%	0%	0%	19%

Table 3.9: Example of the mean layer value and the variation that occurs between maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer viewable quality over the duration of the clip

	Mean	$\min_{qual}$	$\max_{qual}$
SDC-NC	6	4	8
SDC	6	3	8
MDC	4	0	8
SVC	3	0	8

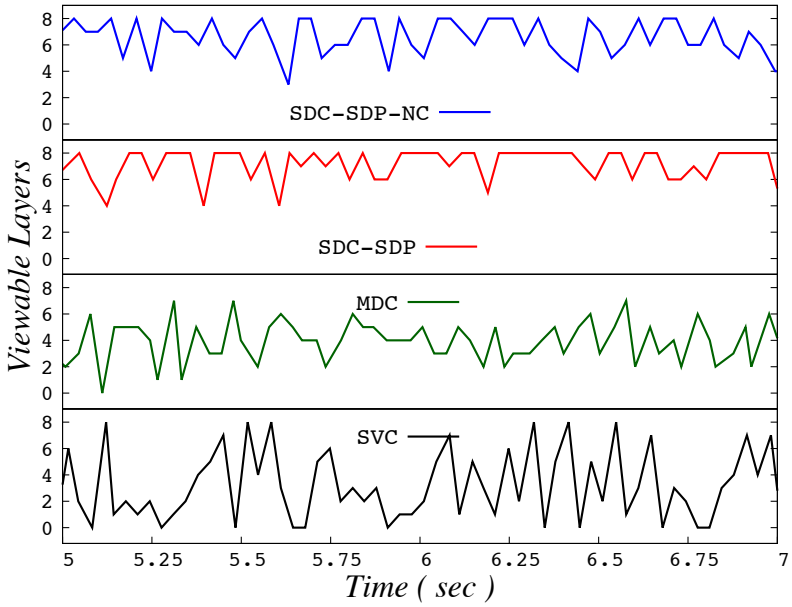


Figure 3.19: Two second example of transitions in viewable quality in SDC-SDP, SDC-SDP-NC, MDC and SVC with eight viewable layers, variable bit-rate and 10% packet loss

also achieved the highest mean viewable quality, layer 6, which is contained within the highest resolution, albeit at the lowest fidelity level. While Figure 3.19 presents a two second snapshot of the data results. It is encouraging to see that the SDC variants provide 42% to 51% of viewing quality at the highest layer and never reduce viewable quality below layer 4, while SVC and MDC have large variations in quality across all

layers. Unfortunately a consistent level of viewable quality is not achieved by SDC. As can be seen in Figure 3.19, SDC-SDP has noticeable levels of maximum viewable quality, layer 8, but large drops in viewing quality, sometimes as low as layer 4, mitigate the benefits of higher quality. The benefits of Network coding provided by SDC-SDP-NC are also unable to provide consistency of viewable quality. These large variations in quality are primarily due to losses in the D_s, rather than specific D_c or D_r.

3.7.2 Measured Impact on PSNR (peak signal-to-noise ratio)

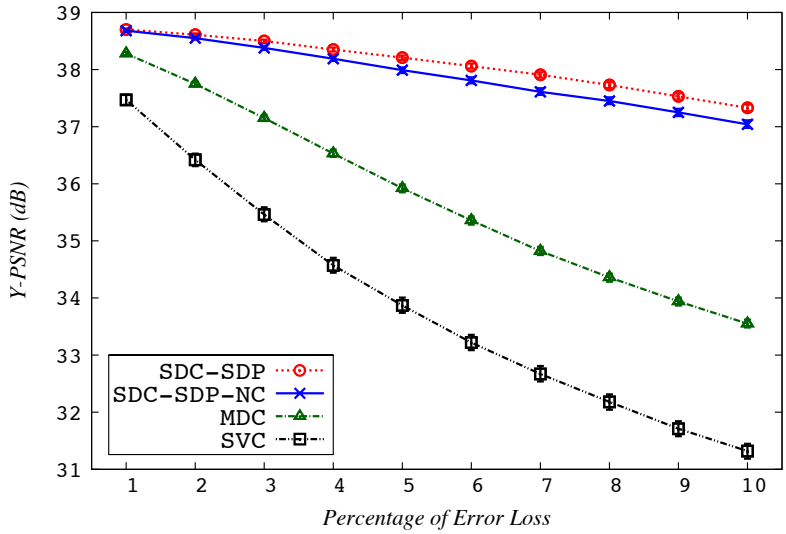


Figure 3.20: Mean Y-PSNR values and 95% Confidence Interval error bars for an eight Layer scheme with variable bit-rate and incremental packet loss.

Figure 3.20 plot the Y-PSNR values for an eight layer scheme with 95% confidence interval error bars, for the streaming models versus percentage of error loss. The major points to note are that MDC has a lower PSNR value as loss rates increase, SVC maintains a similar degradation in quality as loss increases and that SDC-SDP has the highest quality over all loss rates. Finally we note that the Confidence Interval error bars are noticeably small in range, thus implying a closer consistency of PSNR values over different evaluations runs.

We can see that both variants of SDC provide similar PSNR values over all loss rates, a mere .2 dB difference on average. When we compare the PSNR values at 10% loss rate with the percentage of viewing time at a specific layer from Table 3.8, we observe that even though the percentage of viewing time are noticeably different, especially for layer eight, that the PSNR values are very similar. This is due to the similarity in measured pixels rather than a measure of the variation in viewable layer occurring in the stream.

3.8 Conclusion

This chapter presented Scalable Description Coding (SDC), a novel approach for the transmission of redefined MDC-encoded video. SDC shows a noticeable superiority in comparison to MDC in terms of bandwidth requirements and levels of viewable quality, but our goal of consistent high quality for the stream duration is unachieved as some user-perceived quality is sacrificed due to variations in the achievable quality (large changes in layer value being decoded due to loss of the scalable description, as clearly seen in the evaluation results). SDC increases the quality of the stream as most variations occurs in the highest enhancements layers but the variation occurs through the stream duration thus mandating noticeable changes in quality throughout. We believe that selective packetisation is especially significant for mobile networks where bandwidth over-the-air and in the backhaul continue to be insufficient to satisfy the growing demand of video applications.

As we have seen in this chapter, there are benefits to selective packetisation of the layered stream data. The creation of a high prioritised scalable description, while providing a notable increase in viewable quality, mandates large variation in achievable quality when undecodable. Thus separation of the data into high priority and low priority does increase stream quality but is insufficient to provide the consistency of viewable quality required.

The next chapter will take our SDC research and will leverage it to create a new model that will investigate the *equality* of layered data. If all layer data can be packetised so that data loss does not affect individual layers or sections, but all data equally, then viewable quality will be dependent only on the level of packet loss and not on which layer or description the loss was encountered. Thus the goal of the next chapter is to mandate consistency of viewable quality while lowering overall transmission cost.

Chapter 4

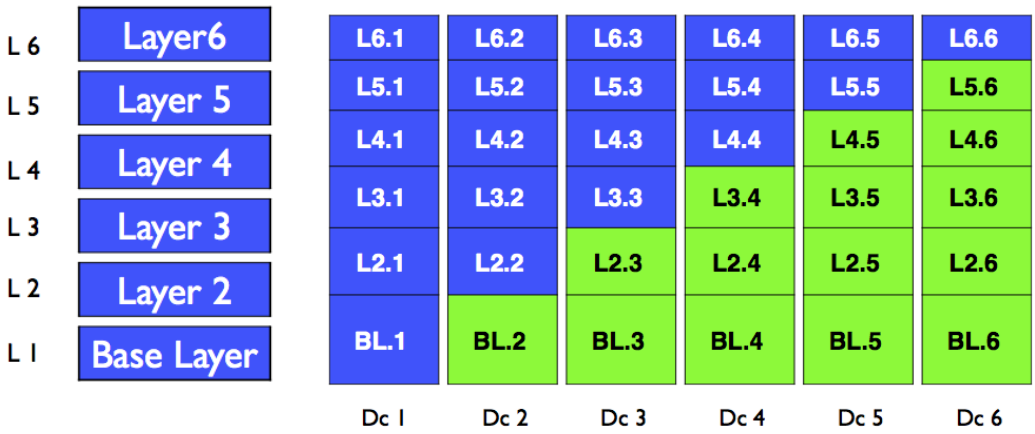
Adaptive Layer Distribution (ALD)

In this chapter we introduce Adaptive Layer Distribution (*ALD*) [15, 16], a novel layered media technique that optimises the trade-off between streaming bandwidth demands and error resiliency. ALD is based on the principle of layer distribution, in which the critical stream data are spread amongst an increased number of descriptions as well as over all packets, thus lessening the impact on quality due to network losses. The proposal of ALD is motivated by two main objectives: reducing the transmission byte-cost overhead of description-based streaming and maintaining a consistent play-out quality over lossy networks, irrespective of GOP size and underlying loss rate. In this context, play-out consistency refers to reducing the frequency of transitions in play-out quality due to packet losses.

ALD introduces the concepts of section thinning and utilises the techniques of improved error resiliency and section-based application packetisation. Our approach focuses on identifying the interrelationship between the various elements of the encoding, packetisation and transmission process. ALD provides a heuristic solution using all the relevant elements at once, such as examining FEC allocation, reducing byte-cost per description and providing packetisation options that mandate consistency of quality over all GOP values. Thus, ALD provides a means of increasing achievable quality, while decreasing overall transmission cost.

4.1 Section Thinning

We begin by introducing the central component of ALD: Section Thinning. This component provides a means of reducing the byte allocation of each layer section per description, while increasing the number of descriptions being transmitted.



(a) SVC **(b)** MDC-FEC

Figure 4.1: An example of (a) a six layered SVC stream encoded as (b) MDC-FEC (blue denotes original SVC data, green - additional FEC data)

Table 4.1: Notation and Definitions

N	The number of SVC layers per Group of Frame (GOP)
$L_{l,x}$	Byte-size of SVC Layer l for frame x
$S_{l,x}$	Layer section byte-size of SVC Layer l for frame x
l	Integer value corresponding to the layer number of L_l
GOP	The number of frames per GOP
STF	Section Thinning Factor
STF_O	STF based on optimal transmission cost
STF_E	STF based on the number of packets per description
D_c	A complete description, containing sections from layers 1 to N
q	Number of MDC descriptions required to decode Layer q
$q+STF$	Number of ALD descriptions required to decode Layer q
IER	Increased Error Resilience for a given layer

We replicate Figure 2.5 and Figure 2.8 from Section 2.2.1 and Section 2.4 respectively, to re-illustrate that the level of additional FEC data in MDC is proportionally high compared to the initial level of SVC data, thus leading to a large increase in transmission byte-cost relative to SVC and highlighting the need for an adaptive mechanism to reduce overall FEC levels.

4.1.1 Section Thinning - Layer Section Allocation

ALD section thinning is motivated to reduce the percentage of FEC data per layer, thus leading to a significant reduction in transmission cost for ALD in comparison to MDC. Section thinning reduces the byte-size of each layer section by increasing the number of ALD descriptions. The formation of the ALD sections follows the same footsteps of MDC section formation, but the section size in each scheme corresponds to a different share of the original SVC layer. In ALD, each section layer-share is scaled by

an additional *section thinning factor* (STF) such that an ALD description section from layer l from frame x, $S_{l,x}$, contains $\frac{L_{l,x}}{l+STF}$ of the layer size. Thus a single complete ALD description from frame x, as shown in Equation (4.1), contains the transmission cost of one section from layers 1 to N, but each section byte-size is smaller. Thus leading to a smaller transmission byte-cost per ALD description:

$$\sum_{l=1}^{N} \frac{L_{l,x}}{l + STF} \tag{4.1}$$

While we view the total transmission cost of one complete ALD description from each frame per GOP as

$$ALD_{D_c} = \sum_{frame=1}^{GOP} \left(\sum_{l=1}^{N} \frac{L_{l,frame}}{l + STF} \right) \tag{4.2}$$

Similar to MDC in Section 2.4.1, a gopFrame variable can be used by ALD to reduce the frame rate. As with MDC, the number of ALD descriptions required to view a specific quality level is based on the transmission cost of a complete ALD description, as per Equation 4.2, times the layer value, or quality layer value (q), requested plus the STF value or "$ALD_{D_c}*$ ((requested layer)+(STF))".

Using the same example as per MDC in Section 2.4.1, where we determine the transmission cost of layer 3 and assuming an ALD STF value of 3, this equates to $ALD_{D_c} * (3+3)$, or six complete ALD descriptions required to decode layer 3. This can be seen in Figure 4.2, where sections L3.1 to L3.6 are required to view layer 3. Thus the total transmission byte cost of ALD required to decode quality layer q is

$$ALD_{D(q)} = ALD_{D_c} * (q + STF) \tag{4.3}$$

While the total FEC transmission cost overhead for ALD quality layer q can be characterised as

$$ALD_{D(q)} - \sum_{frame=1}^{GOP} \left(\sum_{l=1}^{q} L_{l,frame} \right) \tag{4.4}$$

Thus, if STF > 0, the transmission cost of an ALD description is less than the cost of an MDC description, but more ALD descriptions are required to decode the

same quality layer q.

It is important to note that ALD with an STF value of zero equates to the same layer section byte allocation, number of descriptions and transmission byte-cost as MDC. Thus ALD with an STF value equal to zero is exactly MDC. Figure 4.2 illustrates the representation of the six-layer SVC video from Figure 4.1a, using ALD with an STF value equal to three. As shown in the figure, each layer is further extended over the three additional descriptions in comparison to the original MDC.

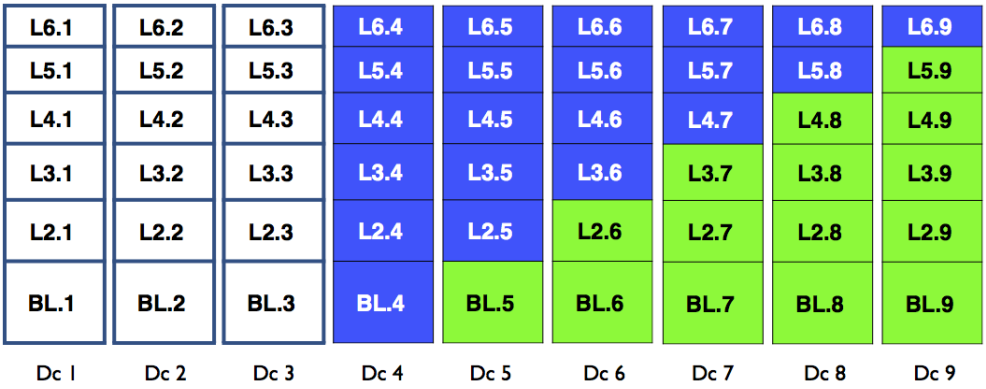


Figure 4.2: ALD GOP for six-layers, with STF = 3

There are a number of points to note when you compare MDC (Figure 4.1b) and ALD (Figure 4.2):

1. As previously highlighted, each MDC description is capable of providing base layer quality, thus mandating MDC to allocate the entire SVC base layer to each MDC description. This can be seen from Equation 2.4 in Section 2.4.1, when we specify $N = 1$, reproduced here for ease of access:

$$\sum_{l=1}^{N} \frac{L_{l,x}}{l} \tag{4.5}$$

Note that the base layer is the first layer and can be defined as layer 1. Hence the allocation cost of a base layer of any frame x to an MDC description is the total cost of the base layer divided by 1, *e.g.* the entire base layer. If we take the example in Figure 4.1b where 6 layers are transmitted, we can see that BL.1 from Dc 1 is the original (blue) SVC base layer, while BL.2 to BL.6, inclusive, are the additional FEC base layer sections. Thus, in this example, leading to six base layer sections being transmitted, or 600% of the original SVC base layer transmission cost. An alternative means of determining the total cost of the base layer in this example is to define the value of q in Equation 2.7 in Section 2.4.1 as 6, thus mandating the total cost of the base layer to be the cost of the original base layer $*$ 6 or 600% of the original SVC base layer transmission cost. Reproduced

here for ease of access:

$$MDC_{D(q)} = MDC_{D_c} * q \tag{4.6}$$

While in Figure 4.2, by utilising *STF*, it can be seen that the original blue (dark) SVC base layer data are distributed over more ALD descriptions, BL.1 to BL.4 in our example, consequently reducing the byte cost of each ALD description base layer section to just 25% of the original SVC base layer. Again this can be determined for the base layer in ALD using Equation 4.1, where we define $N = 1$ and STF = 3. Hence the allocation cost of a base layer of any frame x to an ALD description is the total cost of the base layer divided by $1 + 3$, *e.g.* a quarter of the original base layer. It is important to note that the base layer section in all ALD descriptions in this example contain 25% of the base layer and not just the additional three ALD descriptions above the original quantity in MDC, *e.g.* in all 9 ALD descriptions and not just in the 3 additional ALD descriptions above the 6 descriptions in MDC.

Finally by utilising Equation 4.3 we can determine the total transmission cost of the base layer using ALD. If we define q to be 6 (the maximum layer), STF to be 3 and multiply these by the percentage of the base layer in each description our answer is $(6 + 3) * 25\%$. Thus, in this example, leading to a transmission byte cost of 225% of the original SVC base layer transmission cost, or approximately 38% of the MDC base layer transmission cost. Note that the additional ALD descriptions are shown in white, to illustrate a visual comparison in number of descriptions required by ALD, nine, and MDC, six.

Once this mechanism for section thinning is applied to each layer in the transmitted stream, the transmission byte-cost of ALD is less than MDC. It can be seen that the original blue (dark) SVC data for each layer is shared over more ALD descriptions than MDC descriptions (excluding the highest layer in both schemes where no FEC occurs), thus leading to a reduction in transmission byte-cost, irrespective of encoding rate.

2. The number of FEC sections per layer is consistent between MDC and ALD, but the FEC section byte-allocation in ALD is smaller.

3. A greater number of ALD descriptions, four from Figure 4.2, are required before base layer decoding is achievable. For a device that only needs to view at low-quality this has implications in terms of having to receive more descriptions than with MDC. This is discussed in the next section.

So clearly, the optimal choice of STF is an important design issue that will be

introduced later in this chapter.

4.1.2 Section Thinning - Quality Transmission Cost

Generally, multiple users may be interested in viewing the same video at different qualities, depending on several factors such as the available resources and device capabilities. Section thinning realises significant savings for users interested in receiving high quality video. On the contrary, if a user is interested in receiving low video quality, ALD may result in a larger overhead in comparison to MDC, as additional (STF) ALD-descriptions have to be received in order to decode the base layer. As previously defined, only q MDC descriptions, Equation (2.7) from Section 2.4, are required to decode quality layer q in comparison to $(q + STF)$ ALD descriptions, Equation (4.3), to realise the same video quality.

Hence, the difference in the amount of transmitted data, or total relative overhead $D(q)$, for a single client between ALD and MDC for video quality q, can be calculated as

$$D(q) = ALD_{D(q)} - MDC_{D(q)} \tag{4.7}$$

Note that negative total overhead implies that ALD is more bandwidth efficient than MDC for the selected quality level q.

4.1.3 Section Thinning - Optimal STF Selection

As previously mentioned, multiple users may be interested in viewing the same video at different qualities, thus ALD provides a mechanism for optimal STF ($\overline{STF_O}$) in streaming scenarios for both unicast, single user with one quality requirement, and multicast [141], numerous users with possibly differing requirements. Multicast provides two options for ALD transmission:

i) Each quality layer q is transmitted as a separate entity, thus implementing a multi-bitrate scheme (this option overly increases transmission cost)

ii) Each ALD description is transmitted as a single multicast stream, thus allowing users to subscribe to (q + STF) descriptions to receive the required q quality layer (this option reduces transmission cost, as only the maximum requested quality layer, (max[q] + STF) descriptions, are transmitted thus permitting multiple users access the same descriptions for their respective q' quality layers).

Let p_q denote the percentage of clients interested in viewing a video with quality level q. In a unicast scenario, this would be based on the requirements of a single client, while in multicast, would consider the needs of numerous clients and their varied demands. Thus, the expected total overhead can be estimated as

$$E\{D(q')\} = \sum_{q=1}^{q'} p_q D(q). \tag{4.8}$$

In our design, we choose an $\overline{STF_O}$ value that minimises the expected total overhead and can be expressed as

$$\overline{STF_O} = \arg\min_{STF} E\{D(q')\} \tag{4.9}$$

In this equation, arg min can be defined as the argument of the minimum or the value that for a given argument for the given function attains the minimum value. arg min returns the argument, *i.e.* STF, that returns the minimum value of $E\{D(q)$ rather than the value of $E\{D(q)$ itself. Note that the optimal $\overline{STF_O}$ would vary depending on different factors including the number of layers and the size of each layer. The following is a snippet of the ALD code utilised in Matlab to determine $\overline{STF_O}$:

```
STF=0:frameMaxSTF;
for ii=1:length(STF)
    printAll( printValue,(sprintf('\n*** STF value of %d ***\n', ii-1)));

    % Cumulative ALD transmission cost of each layer section
    Dc=0; %description size
    for jj=1:nLayers
        Dc=SVC(jj)/(STF(ii)+jj)+Dc;
    end

    % Cumulative transmission cost of an ALD description
    ALD=ones(1,nLayers+STF(ii)) * Dc;
    cumALD=cumsum(ALD);

    qualityALD(ii,:)=cumALD(STF(ii)+1:length(ALD));
    overhead(ii,:)=qualityALD(ii,:)-qualityMDC;
end

% determine the cumulative transmission cost of each STF value
totalTransmissionCost=sum(overhead,2);

% determine the percentage difference between cumulative transmission cost of each STF value
percentageSTF = maxSTF;
currentTransmissionCost=totalTransmissionCost(1);
previousTransmissionCost=totalTransmissionCost(1);
for jj=1:length(totalTransmissionCost)
    currentTransmissionCost=totalTransmissionCost(jj);

    if ((currentTransmissionCost-previousTransmissionCost) >
        (previousTransmissionCost*percentageDifference));
        percentageSTF=(jj-1);
        break;
    end
    previousTransmissionCost=currentTransmissionCost;
end
```

```
% get the minimum cumulative transmission cost of each STF value
[minTotal optSTF]=min(totalTransmissionCost);
```

4.2 Improved Error Resiliency (IER)

The main objective of this technique is to enhance the streaming quality by ensuring a smooth play-out with fewer quality transitions. Clearly, the FEC overhead of higher layers in MDC is inversely proportional to the layer-level. For the top-most layer, no FEC is considered. Hence, packet loss rates that cause partial or complete MDC description(s) loss result in an immediate downgrading of the stream quality. Similarly, further proportional reductions in the stream quality for the same GOP is dependent on the cumulative loss of additional descriptions.

Improved Error Resiliency (*IER*) [15, 16] reduces the number of non-redundant sections of layer data, *i.e.* the number of sections containing the original SVC data, by distributing the layer data over a number of reduced sections allowing for one or more additional FEC sections. IER can be applied to any layer, or number of distinct layers, where additional error resiliency is required. However, it is typically applied to the top-most layers to reduce the incurred FEC overhead. While IER changes the level of FEC for a given layer, which could be defined as being an example of Adaptive FEC [100], no previous work is presented in the literature which proposes varying the FEC allocation based on defined description-based layer section sizes.

An example is used to illustrate the concept of IER. MDC in Figure 2.8 from Section 2.4 consists of 6 descriptions, where each description contains a segment from each SVC layer. Each SVC layer is distributed over the MDC descriptions using $\frac{L_l}{l}$, where l denotes the SVC layer index and the remaining MDC sections are populated with FEC data for the respective layers, as previously illustrated in Section 2.4. The SVC layer 6 is distributed over all six descriptions, $\frac{L_6}{6}$, and does not contain FEC, as such any packet loss will reduce viewable quality. To counteract this reduction in quality, we will improve the error resiliency of layer 6 by providing one section of IER, $IER_{6,1}$. This is accomplished by distributing the layer 6 data over five descriptions, $\frac{L_6}{5}$. Determining the reduced distribution of the SVC data provided by IER is undertaken during the initial SVC partitioning, prior to FEC allocation, as previously illustrated in Figure 2.7 in Section 2.4. The remaining layer 6 section is then populated with FEC data. Thus IER mandates an increase in transmission cost as well as providing increased error resiliency. Figure 4.3 illustrates the final compositions of the modified MDC description structure.

Based on equation 2.3 for SVC layer distribution to an MDC description structure from Section 2.4.1, the reduction in divisor provided by IER can be generalised to:

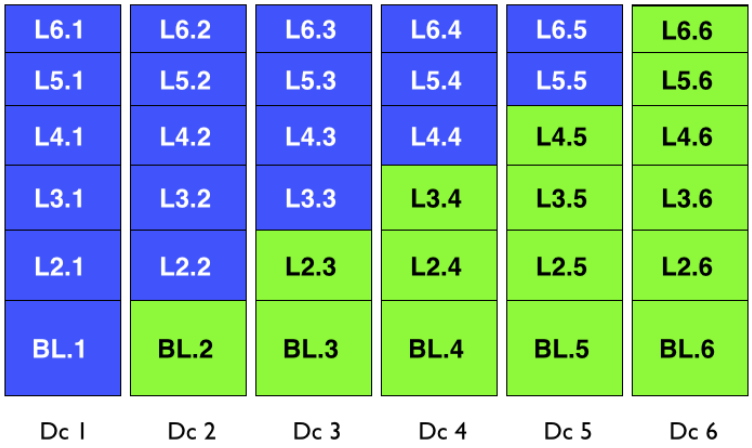


Figure 4.3: One section of IER allocated to Layer 6 of MDC in Figure 2.8 from Section 2

$$\frac{L_l}{l - IER} \tag{4.10}$$

Where l exists between 1 and N, *i.e.* the total number of SVC index values, and IER is a positive numerical value for reducing the l index value. If IER is zero, then equation 4.10 defaults to the standard MDC distribution. Assuming the description structure of Figure 4.3, where the base layer is repeated in every description, IER can not be allocating to the base layer. The layer index must be larger than the level of IER being allocated to the specific layer, *i.e.* for layer 2, a maximum of one additional section of IER can be allocated, while for layer 6, a maximum of five additional section of IER can be allocated. This does not mandate the maximum levels must be imposed, but that a maximum level exists that can not be exceeded. This concept can be generalised as: for layer l, the maximum level of IER applied to layer l can not exceed the index of layer l:

$$IER_{l,x} : l > x \tag{4.11}$$

Finally, there is no optimal level of IER to implement by default. The choice of layer and the level of IER is user or provider specific and may reflect loss rates within the network or the prioritisation of a specific layer within the encoding hierarchy. Figure 4.3 can be viewed as an example where maintaining the quality of the maximum layer is important. As stated the level of IER required is dependent on the level of network loss and in this example layer 6 can incur approximately 16% packet loss prior to a degradation in viewable quality. The 16% threshold is determined based on the additional FEC section. Of the six sections of layer 6 transmitted only five sections are required, thus $\frac{1}{6}$, or 16%, of the transmitted data for layer 6 can be lost before layer 6 is undecodable. As each lower layer in Figure 4.3 contains either an equal amount, *i.e.* layer 5, or higher levels of resiliency, 16% of the transmitted stream can

be lost before there is a reduction in viewable quality. This *loss rate* threshold over all transmitted data is achieved due to the packetisation options of SDP and SD as presented in Section 3.3. While SDP and SD cannot be used simultaneously as both are packetisation options, IER is applied pre-packetisation, thus can be implemented by the streaming model prior to utilising SDP or SD.

In this chapter the ALD STF also needs to be considered and as such the layer share per an ALD description would increase from $\frac{1}{(l+STF)}$ to $\frac{1}{(l+STF-ER)}$, where ER represents the added IER factor. IER may be applied to any number of the higher layers at the expense of additional FEC overhead for the selected layers. However, it is typically applied to the few top-most layers to reduce the incurred FEC overhead. In this section IER-1 can be viewed as providing one additional section of FEC to the highest layer, while IER-2 provides one additional section to the top two highest layers, such that in general IER-n allocates one additional section to the n highest layers.

4.3 Section Packetisation

This component reduces the impact of packet loss on any description-based scalable video, such as MDC and ALD. We provide a brief recap of the two packetisation mechanisms, called section-based description packetisation and section distribution, as previously introduced in Section 3.3.

4.3.1 Section Packetisation - Packetisation Mechanisms

Two packetisation mechanisms are proposed and these are:

i) **Section-based Description Packetisation (*SDP*)**, as introduced in Section 3.3.1, is a description-based layered coding technique, uses sections as application transmission units instead of the entire description. We presented two options for applying SDP: 1) transmitting the data of each section per description as a single transmission unit, which will in general increase the number of packets being transmitted and 2) grouping the data of two or more sections per description as a single transmission unit, so as to reduce the number of packets being transmitted.

Keeping these packetisation options in mind, we also define an $\overline{STF_E}$ value that maintains a level of error resilience per ALD description and can be expressed as a lower bound on the number of packets per description. When the underlying bitrate of the GOP is low, which can occur with low quality clips or when there is a large number of frames per GOP (due to the increased frame interdependency) the total transmission cost of a single description can be less than the underlying

packet payload. When this occurs the loss of a packet mandates the loss of a complete description which can lead to noticeable variation in the viewable quality. For example in our evaluation results, $\overline{STF_E}$ is chosen such that a minimum of two packets are packetised for each description. Hence, the loss of one of these packets would not completely affect an entire description. This approach would sustain high levels of video quality. Hence, the chosen $\overline{STF}$ value can be defined as

$$\overline{STF} = \min\{\overline{STF_O}, \overline{STF_E}\} \tag{4.12}$$

As with IER there is no default value for $\overline{STF_E}$, the granularity in the minimum number of packets mandated per description is dependent on the bitrate of the stream and the levels of network loss. However there is a trade off between the improvement in viewable quality mandated by the increase in the number of packets and the elevation in transmission cost due to the greater number of packet headers.

i) **Section Distribution (*SD*)** as previously stated in Section 3.3.2, creates packets which contain a segment of each section per description, *e.g.* a piece of data from each layer per description. Per GOP these packets mandate "equality" of priority and "equality" of byte-size. SD is utilised by ALD, thus limiting packet loss to only a segment of each ALD section per GOP.

4.3.2 Section Packetisation - Transmission Unit Stream Quality Loss Rate

In this section we present an example of how the achievable quality of a single frame changes as we vary the number of packets that are lost. If we again assume the single GOP example, from the widely-used video clip known as crew.yuv encoded as a six-layer SVC stream, we utilised to illustrate our packetisation options in Section 3.3, illustrated here for ease of access. Table 4.2 shows the byte-size of each layer for a selected frame, which was obtained from the JSVM trace file, "layerTrace.txt", as detailed in Section 3.4.2.

Table 4.2: GOP SVC Layer sizes

Layer	1	2	3	4	5	6
Layer Size	1442	1577	1601	1546	1255	3372

In Table 4.3, we take the byte cost of each layer and we show the transmission cost, in terms of bytes, for SVC, MDC, both options of MDC-SDP and by utilising Eq (4.12) an ALD with an STF of 3. Thus increasing the number of ALD descriptions

to nine. Table 4.3 also presents the number of packets per frame and highlights the best case (B-C) and worst case (W-C) maximum viewable layer based on the loss of a specific number of packets. It is worth noting that the SVC data transmission byte-cost for all versions of MDC are equal, but the total transmission byte-cost for each scheme will vary, dependent on the number of packets being transmitted and the increased byte-cost of packet headers. In this section to provide a simplified example, we evaluate the SVC data element only.

Table 4.3: Example transmission byte-costs for SVC, MDC, MDC-SDP (both options) with viewable quality as packet loss increases

Scheme	SVC	MDC-FEC	MDC-SDP opt1	MDC-SDP opt2	ALD
Transmission Cost	10,793	23,790	23,790	23,790	15,273
# of Packets	11	18	36	18	18
One Lost Pk (B-C / W-C)	5 / 0	5 / 5	6 / 5	6 / 5	5 / 5
Two Lost Pk (B-C / W-C)	5 / 0	5 / 4	6 / 4	6 / 4	5 / 5
Three Lost Pk (B-C / W-C)	5 / 0	5 / 3	6 / 3	6 / 3	4 / 4
Four Lost Pk (B-C / W-C)	4 / 0	4 / 2	6 / 2	6 / 2	4 / 4
...	...	...	...	...	...
Six Lost Pk (B-C / W-C)	3 / 0	4 / 0	6 / 0	6 / 0	3 / 3

As the number of lost packets increases, and dependent on which packet is lost, the quality of the stream can remain high or degrade significantly. As can be seen, SVC is severely affected by packet loss. The worst case (W-C) for all four lost packets, highlights the loss of a packet from the base layer, while the best case (B-C) is based on consecutive losses from the highest quality layer down, *e.g.* layer six is composed of three packets, such that B-C will remain at quality level layer 5 until the fourth packet is lost, when the quality reduces to quality layer four.

MDC-FEC is similar in that each description is composed of three packets, such that the B-C remains consistent over three packet losses, and reduces quality to layer four when the fourth packet is lost. W-C is based on the loss of a single packet from distinct descriptions, thus incrementally reducing quality for each additional packet lost. The increase in viewable quality is consistent with the level of additional error resilience added to the original SVC data, but this increase in viewable quality requires an additional approximately 13,000 bytes of transmission bandwidth.

Consistent with MDC-FEC, both options of MDC-SDP achieving the same W-C viewable quality, again based on a single lost packet from distinct descriptions. Both options of MDC-SDP achieve the maximum B-C over all four lost packets, as loss can be confined to the green FEC section packets. Thus highlighting the benefits offered by section based description packetisation.

As previously stated, ALD employs the section distribution (*SD*) technique for packet packetisation, thus achieving *packet equality*. As highlighted in Table 4.3, this equality produces a uniformity in the B-C and W-C achieved by ALD. As each of the nine ALD descriptions is composed of two packets, achievable viewable quality is

incrementally reduced once two additional packets are lost.

A loss rate of six packet is illustrated to highlight that with the loss of six packets, the transmission cost of ALD over the network is less than the transmission cost of SVC with no packet loss. This offers a comparison of the B-C and W-C quality achieved by SVC and ALD for comparable transmission byte-cost. It is important to note that while the B-C of ALD is less than SVC, the W-C of ALD is better, thus highlighting the balance offered by ALD between transmission cost and achievable consistent quality.

ALD offers one additional option by which B-C and W-C quality can be increased. By implementing the previously highlighted IER technique, ALD-IER, on the highest layer, *e.g.* layer six, one additional FEC section is allocated to layers 6, while the existing SVC data is re-allocated over the remaining 5 sections. In this manner the the achievable viewable quality for both B-C and W-C achieved by ALD-IER for the loss of one or two packets is six, *i.e.* the maximum achievable quality. Thus maximum quality can be achieved for a very minor increase in transmission byte-cost, 47 bytes per ALD description.

4.3.3 Section Packetisation - Transmission Structure of the SVC Data

In our evaluation, we combine both the SD and SDP components with MDC to illustrate simple mechanisms to increase viewable quality, while not increasing SVC data transmission cost. It is important to note that packet equality may mandate a minor increase in overall transmission cost, as some byte rounding up may occur when subsections are divided by R.

Also as each packet now contains a subsection of each layer, *e.g.* subsections of NALs rather than a NAL for a specific layer as defined by SVC, the subdivision of each packet, *e.g.* the specific bytes for each layer subsection, per GOP must be identified to the receiving decoder. Possible options to provide this information are

i. For each GOP, provide a file which details the structure of each packet for each GOP or for all GOPs in the stream, similar in structure to a media streaming manifest file, e.g. DASH [5]. As each packet per GOP contains the same structure only one manifest file is required per GOP. An issue with this option, is that the GOP manifest file may be lost during transmission. One manner in which to reduce this issue is to provide the manifest file during stream setup, thus removing the issue of manifest loss during stream delivery.

ii. Include the packet structure as an additional header within each GOP packet. An issue with this option, is 1) an increase in overall transmission cost as each packet per GOP will contain the header information and 2) repetition, as the additional header is the same in each packet per GOP.

iii. By utilising R and the byte cost of decoding the base layer, $ALD_D(1)$, we can determine the minimum number of packets, $\min(P_k)$, required per GOP to decode the lowest layer. If we divide the byte size of a single instance of the additional header outlined in item ii. by $\min(P_k)$, we can determine the minimum additional byte allocation per packet required by SD so as to determine the structure of each packet per GOP once the minimum number of packets required by the base layer have been received. FEC is utilised to extend this minimum byte size over all packets per GOP. The reason $ALD_D(1)$ is utilised to determine $\min(P_k)$, is that a lower number of packets will not permit decoding of the base layer, so the manifest file is not required, while an increase in packets may provide an increase in viewable quality and the structure of the layer subsections per packet is required for all layers, *e.g.* BL to N.

4.4 Performance Evaluation

The evaluation steps in this chapter are based on the Evaluation Methodology as outlined in Section 3.4. In these following sections we provide results for two different scalable evaluations. The first set of results are based on a GOP value of one, so as to focus on a one dimensional overview of how loss affects scalable media. This GOP value eliminates the interdependence of I, P and B frames. In this manner, the maximum quality of a frame is dependent only upon itself, and is not affected by the quality of neighbouring frames.

The second set of results are based on larger GOP values. In this manner, the effects of a larger GOP value would limit the achievable quality of a frame to the maximum quality of its dependent frame. This set of results illustrates the inherent interdependence of frames within a larger GOP encoding.

4.4.1 Single GOP value Evaluation

In this section, we assess the performance of ALD with respect to SVC, MDC and MDC-SDP (option 1), which we will furthermore refer to as MDC-SDP. Transmission is simulated over a lossy network and the maximum stream quality level achievable at the device is measured. The goal of our evaluation is to determine how viewable video quality, for each streaming scheme is affected by defined levels of packet loss. Thus we do not implement a retransmission mechanism in our evaluation, as in any case to do so would add unwanted delay and network overhead. We begin by outlining the evaluation framework utilised to encode the media, simulate packet loss, and generate the results.

4.4.1.1 Evaluation Framework - Media Encoding

As efficient encoding is the initial step in providing a quantitive evaluation, Joint Scalable Video Model (*JSVM*) [131] software is utilised for our encoding requirements. In this regard, JSVM v9.19, based on eight-layer and one frame per GOP, is utilised to create a multi-layered SVC-compliant stream. This stream is encoded from a widely used raw ten-second 4CIF 30fps media clip, crew.yuv, which consists of a number of astronauts walking down a corridor while waving. All well-known YUV clips utilised in the evaluation section have a total viewing time of ten seconds and were obtained from the well known Leibniz Universität Hannover video library [127].

Table 4.4: SVC structure, defined by JSVM, for an eight-layer, 30 fps, scheme with a GOP of 1 frame

Layer	Resolution	Bitrate	DTQ
0	176x144	345.60	(0,0,0)
1	176x144	826.80	(0,0,1)
2	352x288	1381.70	(1,0,0)
3	352x288	2069.00	(1,0,1)
4	352x288	2755.00	(1,0,2)
5	704x576	4293.00	(2,0,0)
6	704x576	5345.00	(2,0,1)
7	704x576	6796.00	(2,0,2)

Table 4.4, highlights the JSVM SVC structure command line output for the eight-layer, single frame GOP, encoded stream. This example JSVM encoding yielded a maximum PSNR of 38.52, a maximum cumulative bitrate of 6,796 kbit/sec and an encoding time of approximately 13.89 seconds.

JSVM contains a mechanism for extracting actual trace data from the encoded SVC stream, which provides a means for us to determine the transmission cost and packet requirements per frame, for SVC, MDC, MDC-SDP and ALD. Table 4.5 utilises the trace data and highlights the transmission cost, average size of each layer or description, number of layers/description per-frame, the number of packets required and percentage byte size with reference to SVC for each of the adaptive schemes. This trace data is utilised by ns-2 in the simulation section.

By applying Eq (4.12), based on the transmission costs of SVC in Table 4.5, an $\overline{STF}$ value of 6 can be determined. The STF value specifies that six additional descriptions are required, thus increasing the number of ALD descriptions to fourteen. Note that the transmission byte-cost of MDC compared with SVC is 197%, while the ALD transmission byte-cost compared to SVC is 127%, a reduction of approximately 36% when ALD is compared with MDC.

Note also that the number of packets for MDC-SDP has increased to 20,728 while

Table 4.5: Transmission byte-cost as per the eight layer JSVM streaming trace data

Scheme	SVC	MDC	MDC-SDP	ALD
Transmission (bytes)	8,487,766	16,681,768	16,681,768	10,760,302
Layer or Description $\overline{Size}$	3,536	6,950	6,950	2,561
Transmission Units (TU)	Layer	Description	Description	Description
TU Quantity	8	8	8	14
# Packets	6,993	12,456	20,728	9,380
Value compared to SVC	100 %	≈ 197 %	≈ 197 %	≈ 127%

the transmission byte-cost (in terms of payload) has remained the same.

4.4.1.2 Evaluation Framework - Simulation

The transmission of the encoded videos is simulated in Network Simulator 2 (*ns-2*) [138] using myEvalSVC [139], an open source tool for evaluating JSVM video traces for SVC. myEvalSVC presents a means of determining transmission costs, based on the JSVM trace data, and simulating real-time packetisation, over a lossy network, in ns-2. Modifications are made to myEvalSVC scripts to simulate MDC, ALD and their respective variants. These modifications include implementing packetisation based on description structure, section size, SD and STF value, as well as modifications to the sent and received trace files, so as to determine packet loss.

4.4.1.3 Evaluation Framework - Result Generation

As illustrated in the "Trace Analysis" section of Section 3.6.2, the output trace files from myEvalSVC are saved as an achievable quality (AQ) trace file for each streaming model. The AQ trace files are utilised to 1) determine the percentage of viewable frames for each layer, 2) to provide a means of illustrating the transition in frame quality over time and 3) to generate PSNR results for each of the adaptive schemes, thus providing a means of evaluating the performance of each streaming model. PSNR [71] is a widely used pixel quality differentiation mechanism and is utilised to correlate the values in the AQ trace file to the quality of the original YUV media clip. To determine the effects of loss on the quality, the AQ trace file values are first converted to a YUV file. myEvalSVC and JSVM do not contain a reliable mechanism for this form of YUV modification, so we created a new program called modPSNR.exe which is based on the original JSVM source code. Our program "modPSNR", utilises the code base of the JSVM program "PSNRStatic" and adds additional functionality so as to create modified YUV files, which are then measured against the original YUV, so as to determine the deviation in quality, or PSNR, of the modified YUV files.

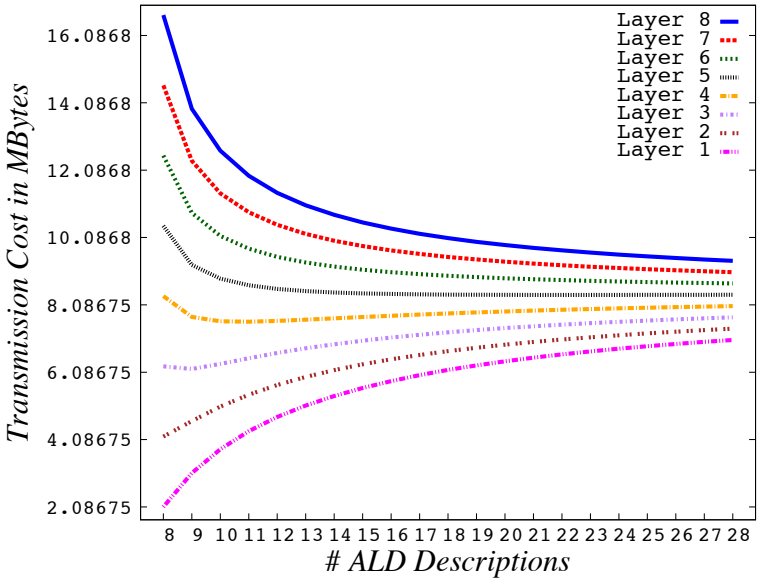


Figure 4.4: Transmission cost of the crew media clip for each distinct media quality, as the STF value and associated number of ALD descriptions increase

4.4.2 Simulation Results

4.4.2.1 Effects of Section Thinning Value Selection

We shall begin by highlighting how the choice of STF affects both the transmission cost of each quality layer and maximum achievable PSNR for the highest quality layer.

Transmission Cost: Figure 4.4 illustrates the transmission cost of each of the different achievable video qualities, as the STF value increases. It can be seen that as the number of descriptions increases, the transmission cost of streaming the higher quality videos decreases, while the transmission cost of streaming lower quality videos increases. The increase in the number of ALD descriptions affects the transmission byte-cost of both the lower quality and higher quality videos in opposite manners. As mention in Section 4.1.1, ALD with an STF of zero equates to MDC. Thus in Figure 4.4, ALD with a description number of 8, illustrates the per layer transmission cost of MDC.

As the value of STF increases, more descriptions are required to decode the base and lower layer quality sub-streams, thus more bandwidth is consumed for the lower quality video. The opposite is true for the higher quality layers, as STF value increases, the bandwidth cost is reduced. As the value of STF increases, this mandates a modification in the transmission byte-cost of the stream. In a scenario where all quality layers are transmitted and fidelity is determined at the device by the number of descriptions required to decode a specific quality layer, an increase in STF delivers an overall reduction in transmission bandwidth costs, due to a reduction in the number of bytes per section, as well as a decrease in the

overall percentage of FEC per layer. Thus Figure 4.4 can be viewed as means of highlighting the transmission trade off between the different quality videos for a number of STF values, namely 0 to 20.

Also the increase in transmission cost of the base quality (minimum quality video) is slower than the drop of the transmission cost of the maximum quality. It is important to note that in description-based streaming scheme (such as ALD and MDC), sections of other layers are transmitted with the layer that is requested. Such that in the case of the base layer quality, not only is the base layer section transmitted, but also sections from all other layers.

Achievable PSNR: Figure 4.5 illustrates the adjustment in achievable PSNR, for the highest quality video, as the value of STF changes from six to five. This adjustment increases the number of ALD descriptions to fourteen. It can be seen that there is a direct correlation between the reduction in transmission byte-cost and maximum achievable PSNR as STF increases.

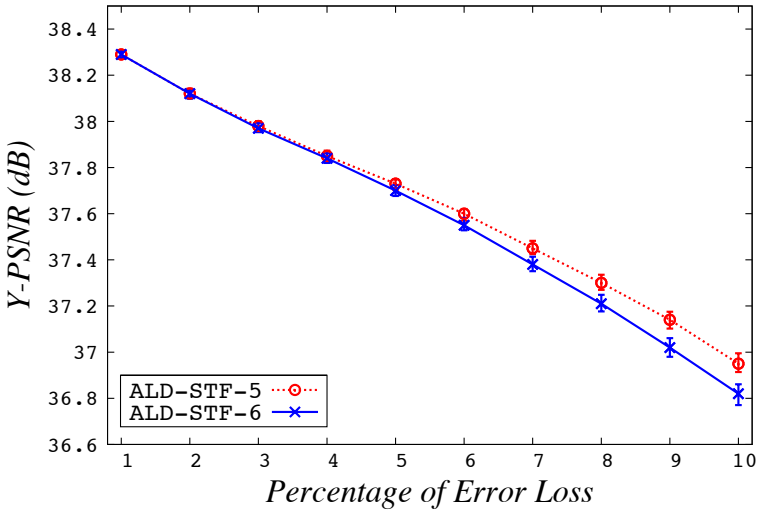


Figure 4.5: Mean Y-PSNR values and 95% Confidence Interval error bars with incremental packet loss for crew with a revised STF value of 6 and the initial value of 5

The adjustment in STF increases the transmission byte-cost by approximately 0.3MB, while also increasing PSNR by approximately 0.1dB at 10% packet loss. Thus STF provides a mechanism for determining the optimal balance between transmission byte-cost and achievable quality.

It is important to note that the highest levels of transmission byte-cost reduction are achieved with low values of STF, as shown in Figure 4.4. Once a slow gradation in transmission cost is achieved, as occurs at ALD description number fifteen in Figure 4.4, further transmission cost reduction supports little additional benefit to achievable stream quality.

Table 4.6: Example of the mean layer value and the variation that occurs between maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer viewable quality over the duration of the clip

	Mean	$\min_{qual}$	$\max_{qual}$
MDC-SDP	7	4	8
ALD	6	0	8
MDC	4	0	8
SVC	3	0	8

4.4.2.2 Impact of Section Thinning

To better understand the degree of variation that can occur in adaptive media streaming, Figure 4.6 shows a two-second snapshot of the ns-2 simulation and the viewable quality transitions that were analysed for SVC, MDC, MDC-SDP and ALD with a packet loss rate of 10%.

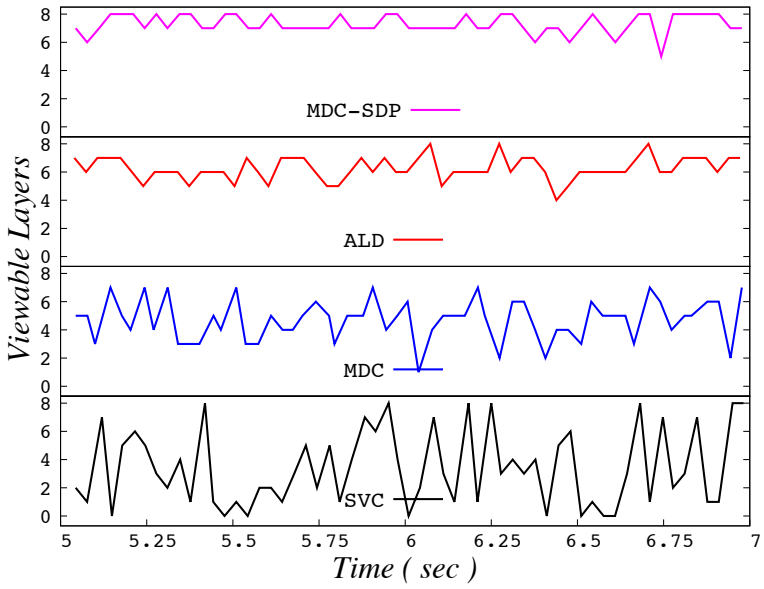


Figure 4.6: Two second example of viewable quality transition for eight-layer SVC, MDC, MDC-SDP and ALD with 10% packet loss.

Table 4.6 outlines a summary of the mean layer value viewed over the duration of the clip, *i.e.* over all 300 frames, and the maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer value viewed, thus illustrating the the variation in quality that occurred during decoding. Note how all models were able to achieve layer 8 decoding, while the mean layer viewable by MDC-SDP and ALD is contained with a resolution level above MDC and SVC. MDC and SVC also mandate the undecidability of some frames within the stream.

From Figure 4.6, it can be seen that SVC and MDC feature the highest frequency of variation and as such would provide a media stream with frequent variation in video quality. MDC-SDP and ALD also contain less variation. More importantly these

variations are limited to the higher quality layers. MDC-SDP has the least amount of variation, with a minimum reduction in stream quality to layer five. ALD does consist of a slight increase in the level of variation, but with a predominantly minimum reduction in stream quality also to layer five. The impact of these fluctuations are reflected in the PSNR values as shown in Figure 4.7.

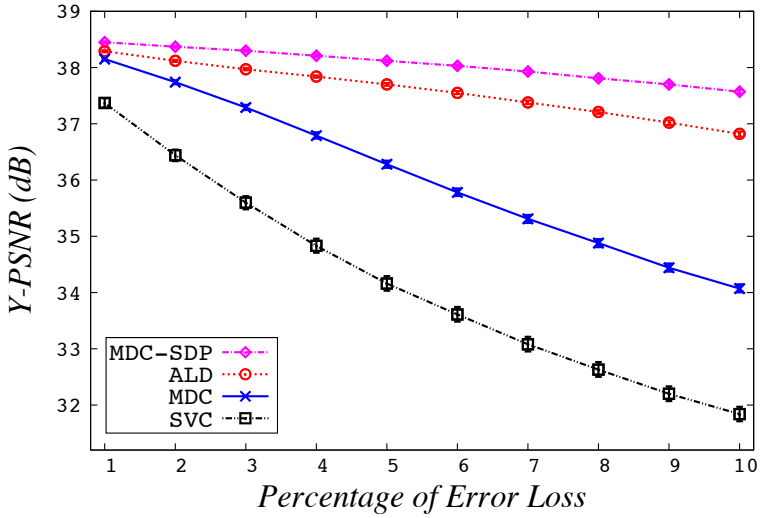


Figure 4.7: Mean Y-PSNR values and 95% Confidence Interval error bars with incremental packet loss

Figure 4.7 plots the Y-PSNR values for the simulated streaming schemes versus the percentage of packet loss. The values are consistent with Figure 4.6, as MDC-SDP performs best, ALD produces a slight reduction in quality, with MDC and SVC having the worst quality. It is important to note that while the PSNR quality of the ALD stream is slightly worse than MDC-SDP, approximately 0.75dB or 2%, the transmission overhead of ALD is approximately 36% less than MDC.

4.4.2.3 Improved Error Resilience

In this section, we investigate the impact of IER on the performance of ALD, as was suggested in section 4.2. Generally, the top most layer in a description based scheme is distributed over the total number of descriptions, thus providing no FEC error correction for the highest quality layer. This can lead to degradation of the maximum stream quality, unless all descriptions are received. Figure 4.6 illustrates that, similar to MDC-SDP, the viewable quality level of ALD continuously moves between the higher quality layers, which can lead to noticeable video variations for a viewer. To increase the level of consistency in the media stream, we propose to increase the level of error resilience for the higher quality layers. In this manner, we distribute the highest layer data over a reduced number of ALD descriptions, thus we marginally increase the byte-size of each higher layer section.

Let us use layer eight, as an example. Currently layer eight is distributed over each of the fourteen descriptions. To facilitate an additional section, without increasing the number of description, we take the existing byte size of layer eight and distribute it over one description less, *e.g.* distribute over thirteen descriptions. Thus increasing the byte-size of layer eight per description section. The remaining section, description fourteen, is then populated with layer eight FEC error resilience, which is consistent with the objective of ALD. Generically, an incremental increase in IER extracts the layer byte size currently divided by N descriptions, decrements N to N-1 descriptions and add one section of FEC error resilience to the N^{th} description.

In this manner, we increase the resilience of the higher quality layers, thus increasing stream quality consistency, with a minor increase in bandwidth transmission. Table 4.7 highlights the increase in transmission byte-cost, number of packets per ALD (IER) scheme and byte-size when compared with SVC. Notice the slight increase in transmission byte-cost for the IER schemes. This is due to the highest layers being distributed over the greatest number of descriptions and containing the least amount of error resiliency.

Table 4.7: Increasing transmission byte-cost for ER, with reference to Table 4.5

Scheme	ALD	ALD-IER-1	ALD-IER-2
Transmission (bytes)	10,760,302	10,899,756	11,017,664
Layer or Description $\overline{Size}$	2,561	2,595	2,623
# Packets	9,380	9,492	9,562
Value compared to SVC	$\approx$ 127%	$\approx$ 128 %	$\approx$ 130%

Table 4.8: Example transmission byte-cost of the crew media clip for each quality layer, for each adaptive scheme. Number of layers or descriptions required to received the respective quality is shown in brackets. Transmission costs of MDC and MDC-SDP are equal, with only MDC being shown.

Layer	SVC	MDC	ALD	ALD-IER-1	ALD-IER-2
8	8,487,766 (8)	16,681,768 (8)	10,760,302 (14)	10,899,756 (14)	11,017,664 (14)
7	6,674,789 (7)	14,596,547 (7)	9,991,709 (13)	10,121,202 (13)	10,230,688 (13)
6	5,360,940 (6)	12,511,326 (6)	9,223,116 (12)	9,342,648 (12)	9,443,712 (12)
5	3,440,447 (5)	10,426,105 (5)	8,454,523 (11)	8,564,094 (11)	8,656,736 (11)
4	2,584,949 (4)	8,340,884 (4)	7,685,930 (10)	7,785,540 (10)	7,869,760 (10)
3	1,728,843 (3)	6,255,663 (3)	6,917,337 (9)	7,006,986 (9)	7,082,784 (9)
2	1,036,455 (2)	4,170,442 (2)	6,148,744 (8)	6,228,432 (8)	6,295,808 (8)
BL	434,947 (1)	2,085,221 (1)	5,380,151 (7)	5,449,878 (7)	5,508,832 (7)

Figure 4.8 presents the variation in stream quality transitions, as IER is first increased on layer eight (ALD-IER-1) and then increased on both layer eight and layer seven (ALD-IER-2). Note that while the consistency of the stream quality at ALD-IER-2 is not continuous, there is a noticeable increase in view-ability for the higher quality layers, which is compatible to the MDC-SDP stream replicated. Note that

Table 4.9: Example of the percentage of decodable frames per quality level for each of the adaptive schemes, based on the crew media clip with 10% packet loss

Layer	SVC	MDC	MDC-SDP	ALD	ALD-IER-1	ALD-IER-2
8	10.67%	1.67%	34.00%	6.33%	23.00%	21.67%
7	6.67%	10.67%	46.33%	40.67%	26.33%	48.00%
6	4.00%	17.33%	15.00%	40.00%	39.00%	18.67%
5	14.00%	26.33%	3.67%	11.33%	10.33%	10.66%
4	9.67%	25.67%	1.00%	1.67%	1.34%	1.00%
3	12.00%	13.00%	0.00%	0.00%	0.00%	0.00%
2	9.67%	4.33%	0.00%	0.00%	0.00%	0.00%
BL	20.00%	1.00%	0.00%	0.00%	0.00%	0.00%
Un-viewable	13.33%	0.00%	0.00%	0.00%	0.00%	0.00%

such improvement is achieved at a minor increase in transmission byte-cost as shown in Table 4.7.

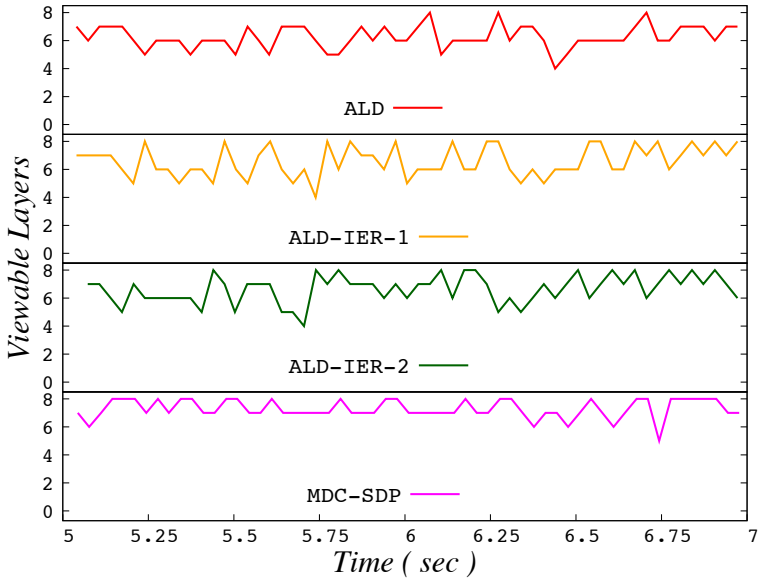


Figure 4.8: Two second example of viewable quality transition for ALD and MDC-SDP, from Figure 4.6, and ALD-IER-1 (layer eight) and ALD-IER-2 (layer eight and layer seven)

Table 4.10 outlines a summary of the mean layer value viewed over the duration of the clip, *i.e.* over all 300 frames, and the maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer value viewed, thus illustrating the the variation in quality that occurred during decoding. Note how all models are able to achieve layer 8 decoding and the mean of all models are contained within the highest resolution level. IER-1 improves the minimum layer level of ALD to 4 while IER-2 improves the mean layer value to 7, thus facilitating higher levels of viewable quality with minimum increases in overall transmission cost.

Figure 4.9 shows the increase in PSNR for ALD-IER-1 and ALD-IER-2, with reference to ALD and MDC-SDP with up to 10% packet loss. As can be seen, at low error loss levels, the PSNR curve for both IER-1 and IER-2 provide the best performance, while as the percentage of loss increases, the PSNR curve for both IER-1

Table 4.10: Example of the mean layer value and the variation that occurs between maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer viewable quality over the duration of the clip

	Mean	$\min_{qual}$	$\max_{qual}$
ALD	6	0	8
ALD-IER-1	6	4	8
ALD-IER-2	7	4	8
MDC-SDP	7	4	8

and IER-2 will converge with ALD due to the reduction in benefits yielded by only one additional section at the highest layer(s). Future works will determine the optimal number of additional sections required by IER to maintain quality consistency as loss increases.

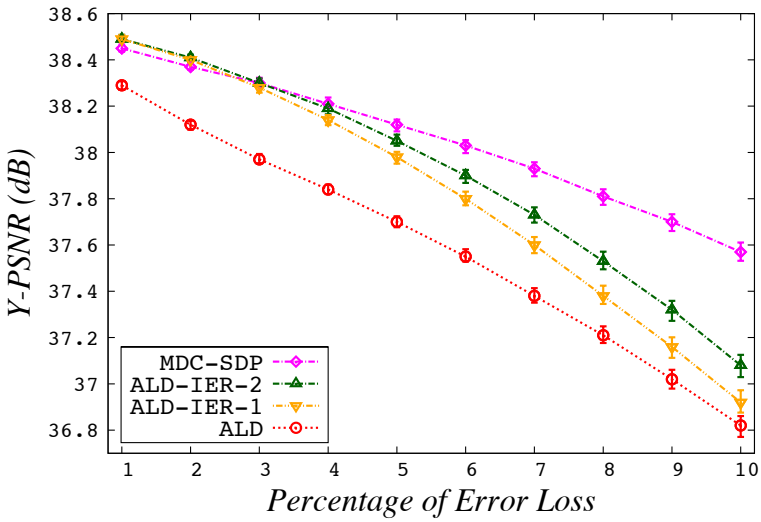
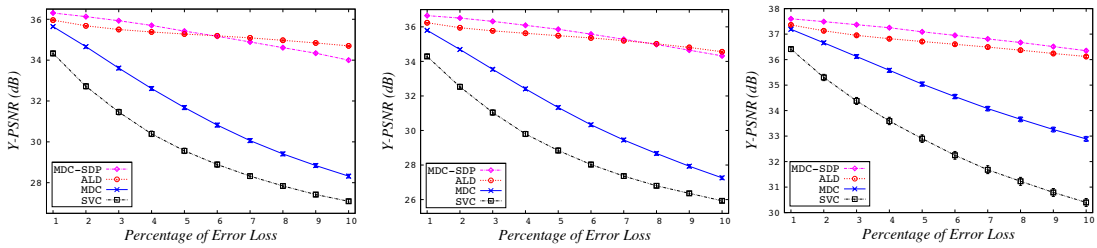


Figure 4.9: The PSNR increase in ALD with IER on layers eight and seven

Table 4.8 gives an overview of the transmission byte-cost for each of the streaming schemes evaluated. It can be seen, that the SVC transmission byte-costs per layer are lowest. ALD delivers a dramatic reduction in transmission byte-cost for higher quality layers when compared with MDC, but as mentioned in Section 4.1.1, ALD has a higher transmission cost, than MDC, for users requiring lower layer quality. These results are consistent with Figure 4.4.

Table 4.9 outlines an example of the percentage of decodable frames per quality level for each of the adaptive schemes. It can be seen that MDC-SDP consists of a large percentage of decodable frames for quality layers six to eight, thus providing a consistent high level of viewable quality. ALD also contains large quantities of layers six and seven but the inclusion of the low bandwidth cost IER, re-allocates these high percentage levels over layers six to eight. Thus highlighting the increased benefits provided by IER.



(a) City stream, $\overline{STF}$ 2 **(b)** Harbour stream, $\overline{STF}$ 5 **(c)** Soccer stream, $\overline{STF}$ 3

Figure 4.10: PSNR results for additional stream evaluation. Note that for clarity, the plots use different scales on the Y-axes

4.4.2.4 Additional Evaluation

Three additional streams were evaluated using the same JSVM SVC encoding as the Crew stream and their evaluation results are given in this section. The streams are City, a slow moving aerial shoot of a city, Harbour, a stationary shot of a harbour with a number of moving boats, and Soccer, a fast moving shot of a game of soccer. All additional clips are available from the Hannover repository [127] and a single frame from all four evaluated streams is given in Figure 4.11. By utilising Eq (4.12) and based on the SVC per layer transmission byte-costs of each additional media clip as presented in Table 4.11, the $\overline{STF}$ values for city is 2, the $\overline{STF}$ value for harbour is 5 and the $\overline{STF}$ value for soccer is 3.

Figure 4.11: Single frame image from each of the four evaluated streams: crew, city, harbour and soccer

Table 4.12 highlights the relevant maximum PSNR and transmission byte-cost per adaptive scheme for each of the additional streams; note the varying percentage increase in transmission cost for MDC and ALD.

Table 4.11: SVC transmission byte-cost for each quality layer, of the additional media clips

Layer	City	Harbour	Soccer
8	14,797,807	16,311,564	10,189,375
7	11,179,340	13,017,572	7,665,548
6	8,409,220	10,412,045	5,851,889
5	4,731,469	6,585,181	3,607,645
4	3,514,773	5,197,057	2,636,439
3	2,118,932	3,469,141	1,639,147
2	980,230	1,396,047	781,908
BL	358,483	577,249	319,650

Figure 4.10 outlines the achieved PSNR for each of the evaluated adaptive schemes per media clip. It is important to note that while the maximum PSNR and transmission byte-cost for each of the adaptive schemes vary, the comparability between the different levels of PSNR is consistent across all three media clips. Also observe the convergence of ALD and MDC-SDP in both the city and harbour streams. This is consistent with the initial low PSNR values and high transmission byte-cost for MDC for both streams.

Table 4.12: Additional stream maximum PSNR and transmission byte-cost per adaptive scheme c/w percentage value compared to SVC

Scheme	City	Harbour	Soccer
PSNR	36.47	36.78	37.72
SVC	14,797,807 (100%)	16,311,564 (100%)	10,189,375 (100%)
MDC	24,811,448 (168%)	30,466,240 (187%)	17,823,312 (175%)
ALD	20,392,250 (138%)	21,091,057 (129%)	13,637,558 (134%)

4.4.3 Larger GOP value Evaluation

In this section, we evaluate how the consistency of viewable quality of SVC, MDC and ALD, and their respective variants, vary as the number of frames per group of picture (*GOP*) increases. Our simulations show that as the number of frames per GOP increases, ALD, and its packetisation and transmission techniques, can offer consistency of viewable quality for longer periods of time, while the interdependence of layers and individual frame types, *e.g.* I, P and B frames, can further limit the achievable quality of existing scalable streaming models, *e.g.* SVC and MDC. Additionally, our results show that for larger GOP values, our models can increase the consistency and quality of scalable media for all users while leveraging the benefits of overall network transmission

Table 4.13: QP value per layer for all clip types and GOP values

Resolution	QCIF		CIF			4CIF		
Layer	BL	2	3	4	5	6	7	8
QP Value	34	28	33	30	28	35	32	30

Table 4.14: Maximum achievable PSNR value (dB) per clip type for layer 8 with a GOP value of one

Layer	City	Crew	Harbour	Soccer
PSNR	36.7	38.75	37.02	38.03

cost reduction offered by SVC with larger GOP values (approximately 90% reduction when comparing a GOP value of 1 to a GOP value of 32).

4.4.3.1 Modifications to Evaluation Framework for Larger GOP Values

In this Section, we present the modifications to our performance evaluation framework for larger GOP value evaluation. Our GOP evaluation is based on the well-known 10 second *city* video, an aerial view of a building landscape, obtained from the Leibniz Universität Hannover video library [127]. These videos are recorded at 30 frames per second totalling 300 frames per video. The videos are encoded using JSVM [131] to eight layers with spatial and quality scalability, using medium grain scalability (*mgs*) and quantizer parameter (QP) values as per Table 4.13.

As illustrated, we consider three resolutions (QCIF, CIF and 4CIF) with respective 2, 3, and 3 quality levels, *e.g.* two fidelity levels in the lowest resolution and three fidelity levels in each of the higher resolutions. For larger GOP values, we use the same QP values for all encodings and only vary the GOP value. The QP values in Table 4.13 provide a means of demonstrating how the bitrate of the encoded clip and associated GOP value can mandate variation in the maximum achievable quality of the individual SVC layers. Table 6.2 highlights the changes in maximum achievable PSNR for layer eight for each of the clip types with a GOP value of one. In [142], the authors define that a typical choice of QP values in AVC and HEVC encodings to be 22, 27, 32 and 37 based on the software reference configuration specified by [143]. While these QP values are sufficient when comparing clips of a defined quality and a single resolution, *e.g.* clips containing only a single layer, for scalable video a separate QP value is required for each individual layer in the SVC encoding so as to determine a quality level for each layer. The QP values utilised in our encodings provide a means of mandating that the lower the layer value, the lower the underlying bitrate of that specific layer, *e.g.* the base layer will have the lowest bitrate, layer eight will have the highest bitrate and the layers in-between will have incrementally higher bitrates. This provides for a gradual increase in transmission cost as the viewable quality increases.

Table 4.15: Transmission Megabyte-cost for quality layer 8 for each adaptive scheme, for each of GOP values and maximum achievable PSNR for Layer 8. All MDC variants, MDC, MDC-SDP and MDC-SD, have the same transmission cost, so only MDC is shown

Layer	PSNR	SVC	MDC	ALD	ALD-IER-2
GOP1	36.7	18.49	32.79	24.90	25.75
GOP8	35.5	4.56	7.79	6.05	6.25
GOP16	35.0	2.83	4.91	3.79	3.90
GOP32	34.5	1.93	3.41	2.61	2.68

For GOP we evaluate six streaming models, SVC, MDC, MDC-SDP, MDC-SD (where MDC description data are packetised using section distribution), ALD and ALD-IER-2 (ALD with one additional FEC section for the two highest layers, L7 and L8, thus providing increased protection for the maximum viewable quality), over four distinct GOP values, *e.g.* number of frames per GOP, namely 1, 8, 16 and 32.

Similar to our GOP of one simulations, ns-2 and myEvalSVC are used to create modified evaluation scripts which are used to simulate real-time packetisation over a lossy network. In our modified evaluation scripts, we packetise the various models based on their respective encapsulation unit, *e.g.* layer, description, section, thus providing clear distinction between the various units during transport. In this manner the loss in one unit will not effect the quality achievable from any other unit. For SVC, each individual layer per frame is partitioned in one or more packets. With MDC each description per frame is packetised separately. For MDC-SDP each section per layer is packetised individually. While in ALD, ALD-IER-2 and MDC-SD, each packet contains a segment of each layer per description (using SD). In ALD, ALD-IER-2 and MDC-SD, this would lead to a segment from every layer per packet. As can be seen once we begin to control the structure of the packetisation we can reduce the interdependence of the units right down to packet level. Thus lessing the effects of network loss on viewable quality.

For ALD and MDC-SD, as GOP value increases, SD mandates that each packet transmitted contains not only a segment of each layer per description per frame, but also a segment of each layer per description from each frame per GOP, thus mandating packet equality for all frames per GOP. The remainder of the scripts are similar to the GOP value of one evaluation.

4.4.3.2 Evaluation Results for an Increased Number of Frames per GOP

The purpose of evaluating an increase in the number of frames per GOP is to determine the effects of inter-frame dependency on viewable quality for scalable video. As the GOP value increases, the overall transmission cost is reduced but the inter-frame dependency increases. This increase typically affects the viewable quality. The de-

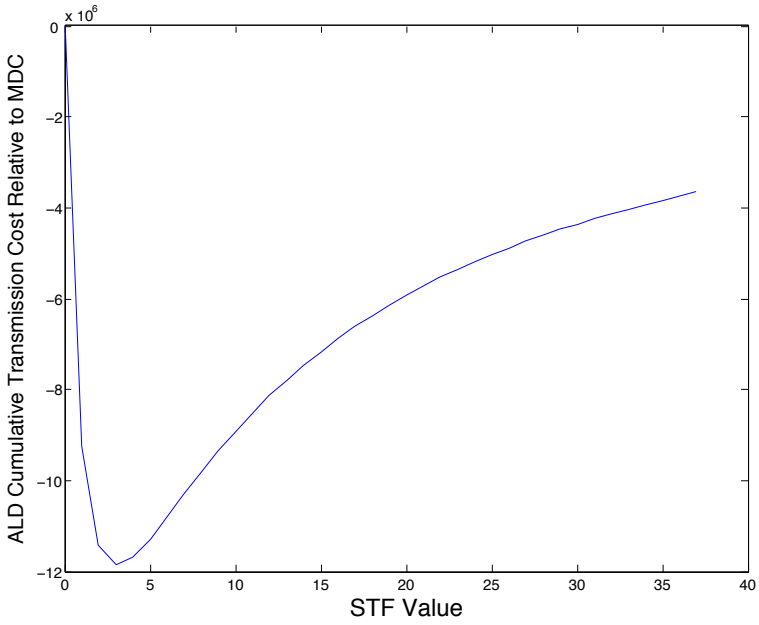


Figure 4.12: City stream with a determined STF_o value of 3 for a GOP of one

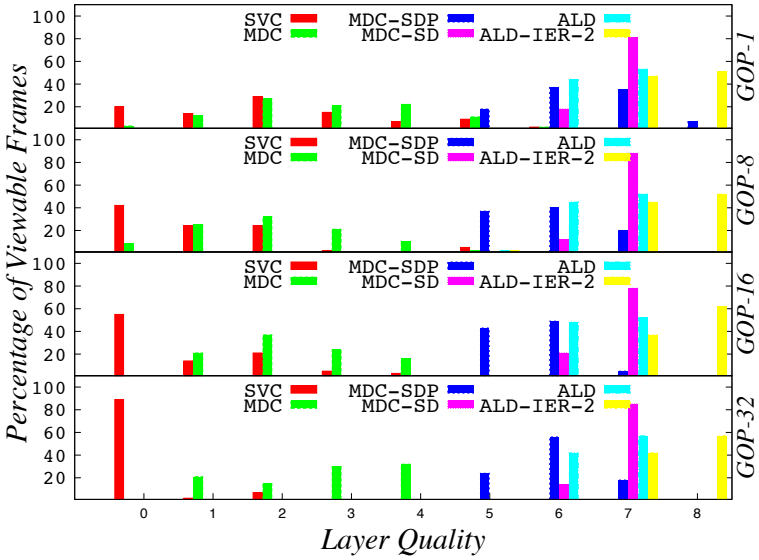


Figure 4.13: An example of the percentage of viewable frames, with a 10% packet loss rate, for each of the six streaming models for each of the four GOP values for the city clip

veloped techniques in ALD benefit from the reduced transmission cost of larger GOP values while maintaining consistent levels of viewable quality.

As GOP increases, as illustrated in Table 4.15, the transmission cost of MDC and ALD changes. As STF is based on the cumulative transportation cost of ALD relative to MDC, this has the potential to create different STF values for differing GOP values. For our evaluation, this created STF values of 3 for a GOP of one and a GOP of thirty two, and an STF value of 2 for a GOP of eight and a GOP of sixteen. To provide consistency of STF value used in the evaluation over all GOP values, we

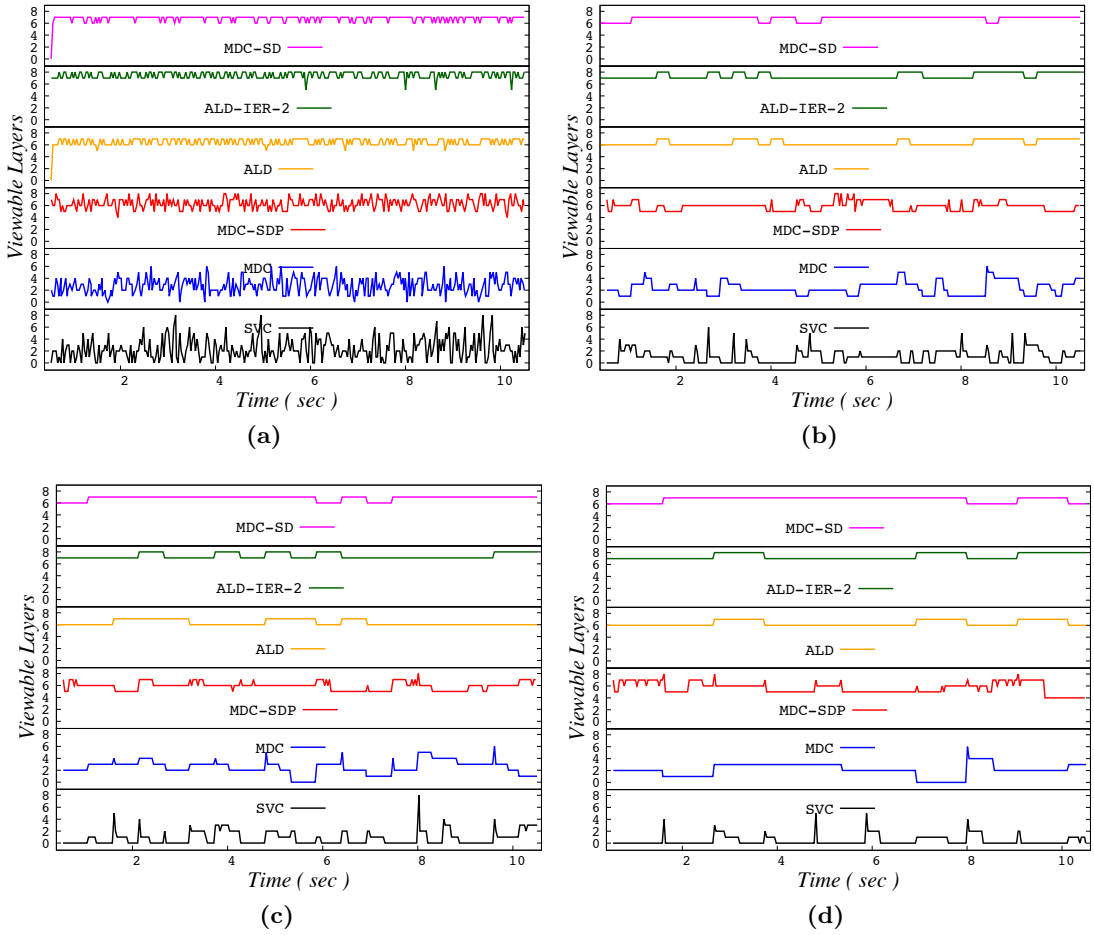


Figure 4.14: Ten second examples of the stream quality transitions for each streaming model of the city clip with a GOP value of 1 (a), 8 (b), 16 (c) and 32 (d) with a 10% packet loss rate

use the same value of STF, *e.g.* 3, as defined for a GOP of one, for all ALD and ALD-IER-2 simulations. The increase in STF value for a GOP of eight and a GOP of sixteen reduces their overall transmission cost by 294Kb and 189Kb respectively, by reducing their levels of FEC allocation. The STF is defined as per the developed optimisation framework shown in Section 4.1.1. Figure 4.12 provides the optimal STF value for the city clip type with a GOP of one. For each GOP value we evaluated packet loss rates from 1% to 10%. We only illustrate results for a 10% packet loss rate, but evaluation results for packet loss rates from 1% to 9% provided similar conclusions.

Table 4.15 displays the transmission megabyte-cost of layer 8 for each streaming model, for each of GOP values and maximum achievable PSNR for Layer 8. All MDC variants, MDC, MDC-SDP and MDC-SD, have the same transmission cost, so only MDC is shown. Note the approximate 90% decrease in transmission cost between GOP1 and GOP32. Thus illustrating the benefits provided by a higher number of frames per GOP, in scenarios where congestion and large burst loss may occur. Also note that mandating the same QP and encoding values, maximum achievable PSNR decreases,

illustrating the link between encoding, transmission cost and viewable quality.

Figure 4.13 plots the percentage of viewable frames for each of the six streaming models for the city clip with a packet loss rate of 10%. Each plot illustrates a different GOP value. Higher quality is illustrated by larger percentage values in the higher layers. Note how

i. only MDC-SD, ALD and ALD-IER-2 provide the same approximate percentage rates for the higher layer values for each of the GOP values, thus providing consistency of higher quality decoding as GOP values increase.

ii. only ALD-IER-2 provides this consistency of higher quality decoding at the highest level, layer 8, as the GOP value increases.

iii. once SVC is encoded with 32 frames per GOP, over eighty percent of the frames are undecodable due to packet loss.

iv. the simple packetisation options of SD and SDP greatly increase the viewable quality of MDC, without increasing MDC data transmission cost.

v. the decodable quality of ALD-IER-2 never drops lower than layer 7.

Figure 4.14 plots ten-second examples of the stream quality transitions for each streaming model of the city clip with a GOP value of 1 (Figure 4.14a), 8 (Figure 4.14b), 16 (Figure 4.14c) and 32 (Figure 4.14d) with a 10% packet loss rate. For each value of GOP it can be seen that SVC and MDC feature the highest frequency of variation and as such would provide a media stream with frequent variation in video quality. The other models contain less variation and more importantly these variations are limited to the higher quality layers, thus mandating higher achievable video quality. The plots also illustrate how a simple mechanism which re-packetises MDC data (MDC-SDP mandating section packetisation and MDC-SD where each packet contains a segment of each layer per description) can produce such considerable increases in viewable quality at no increase in transmission cost. Note how as GOP increases, the detrimental effects of inter-frame dependency decreases achievable stream quality for some of the streaming models, *e.g.* SVC, MDC and to some extent MDC-SDP. Furthermore for ALD, ALD-IER-2 and MDC-SD, the minimum transitions that can occur is consistent with the number of frames per GOP, *e.g.* for GOP32, the minimum number of frames for a given layer is 32. Finally, as ALD and MDC-SD do not contain FEC error resilience on the maximum layer, *e.g.* layer 8, only ALD-IER-2 can provide maximum achievable quality and in the plots for GOP16 to GOP32, ALD-IER-2 only varies between the highest two layer, *e.g.* Layer 7 and 8.

Figure 4.15 illustrated the frame interdependency of our 8 frame GOP encoding. For layers one and two, the JSVM encoding creates P frames for each frame, while for layers three to eight, the encoding implements a *IBBBPBBB* design. The arrows

in Figure 4.15 present the frame interdependency, with the arrow point denoting the dependent frame. As can be seen, the loss or a low quality value of an I or P frame will mandate low quality streaming for all frames which are dependent on it. GOP16 and GOP32 contain the same structure of one I and one P frame per GOP.

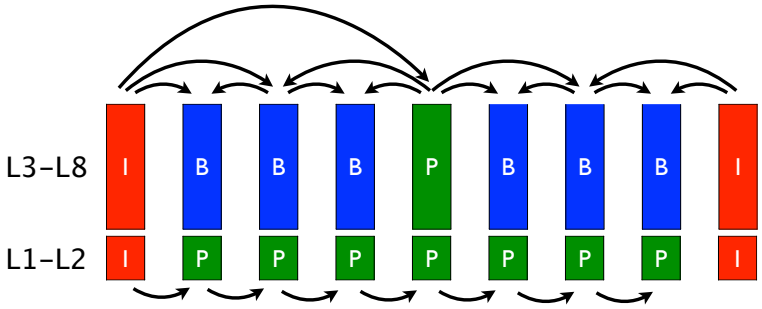


Figure 4.15: Inter-frame dependency for our 8 frame GOP encoding

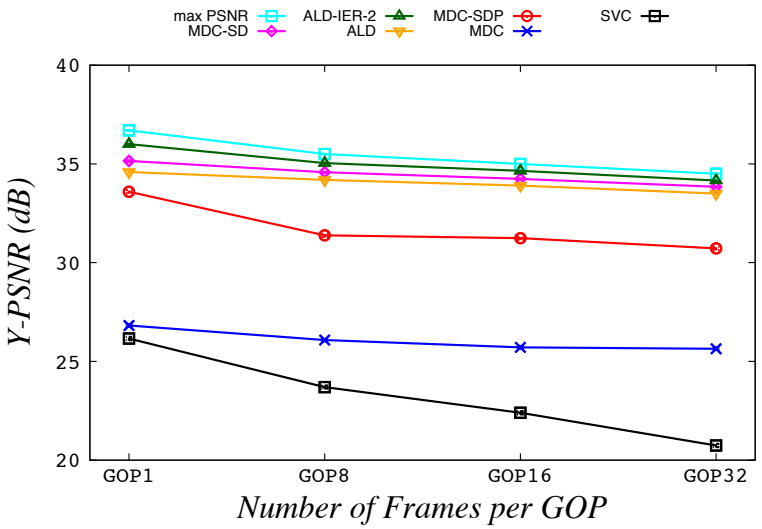
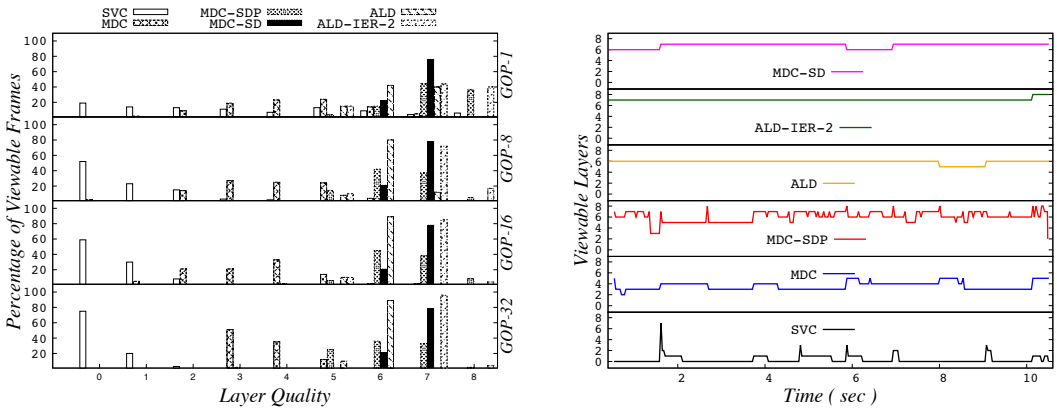


Figure 4.16: A plot illustrates maximum achievable PSNR, shown as "max PSNR", and PSNR values for each of the streaming models with a 10% packet loss rate, as GOP increases

Figure 4.16 illustrates the maximum achievable PSNR, shown as "max PSNR", and the changes in PSNR value for each of the streaming models with a 10% packet loss rate, as GOP increases. PSNR provides a numerical representation of the achievable viewable quality of a model. As can be seen, the streaming models that deliver the highest layers from Figure 4.15, achieve the highest PSNR values. SVC and MDC provide low quality overall. As was seen in Figure 4.13, over 200 frames of SVC were undecodable with a GOP of 32, thus the evaluated PSNR value is primarily composed of duplicated frames with low fidelity. It is only the minor changes in background imagery, that mandate such a high PSNR value for SVC with a GOP of 32. MDC-SDP provides an increase in dB, relative to MDC and SVC, of between 6dB (GOP1) and 10dB (GOP32). ALD and MDC-SD provide a further noticeable increase in viewable quality, while ALD-IER-2 provides near maximum achievable PSNR for all GOP values.

Table 4.16: STF for each of the clip types with a GOP of one

Layer	City	Crew	Harbour	Soccer
STF	3	6	7	3



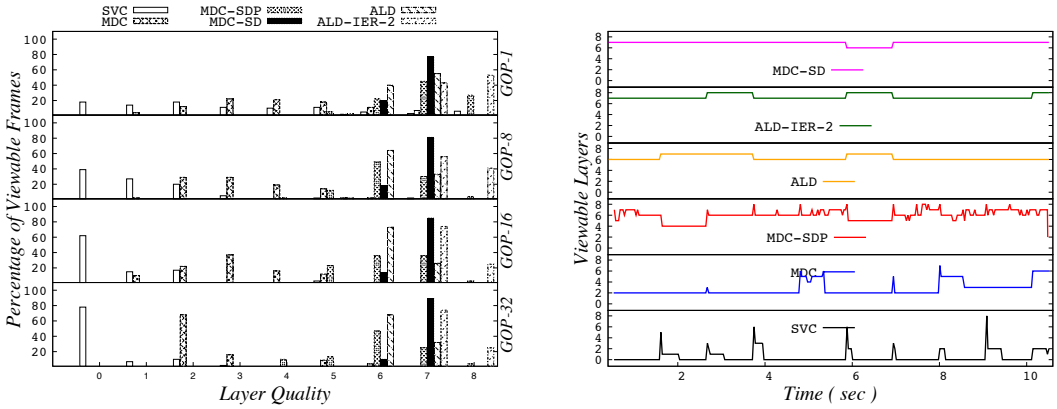
(a) **(b)**

Figure 4.17: (a) An example of the percentage of viewable frames for each of the six streaming models for each of the four GOP values for the crew clip and (b) Ten second example of the stream quality transitions for each streaming model of the crew clip with a GOP value of 32. Both with a 10% packet loss rate.

Thus the evaluation of a higher number of frames per GOP illustrates the benefits of selective packetisation, improved error resistance and adaptive FEC allocation provided by our techniques.

Three additional video streams, *crew*, *harbour* and *soccer*, were also assessed over all GOP values, using the same evaluation framework as *city*. The developed optimisation framework shown in Section 4.1.1 defined STF values of 6 (*crew*), 7 (*harbour*) and 3 (*soccer*) for their respective ALD and ALD-IER-2 simulations, as shown in Table 4.16.

To further confirm the ability of the ALD framework to realise similar gains for different videos, we present a sample of the results for the crew and soccer clips. Figure 4.17a presents an example of the percentage of viewable frames for each of the six streaming models for each of the four GOP values for the crew clip while Figure 4.17b illustrates a ten second example of the stream quality transitions for each streaming model of the crew clip with a GOP value of 32. Both figures with a 10% packet loss rate. Figure 4.18a and Figure 4.18b provides the same plots for the soccer clip. It can be seen that the results presented for crew in Figure 4.17b and for soccer Figure 4.18b are consistent with the results seen for city in Figure 4.14d. MDC-SD and ALD are viewable in layers 7, 6 and 5, ALD-IER-2 in layers 7 and 8, with the other streaming models containing large variations in viewable layer value. One time to note in Figure 4.17b is that ALD-IER-2 is viewable primarily in layer 7, while ALD drops to layer 5 once during the duration of the stream. This reduction in viewable quality



(a) **(b)**

Figure 4.18: (a) An example of the percentage of viewable frames for each of the six streaming models for each of the four GOP values for the soccer clip and (b) Ten second example of the stream quality transitions for each streaming model of the soccer clip with a GOP value of 32. Both with a 10% packet loss rate.

can also be seen in Figure 4.17a where there is a reduction in the viewable quality for the defined layer values of ALD and ALD-IER-2 for increasing values of GOP.

The reason for this reduction in quality is due to the selection of STF for crew. The STF values for each of the tested clips are illustrated in Table 4.16. It can be seen that city and soccer have the same STF value, thus would have similar levels of FEC resiliency. While crew and harbour have an increased STF which leads to an increase in the number of ALD descriptions required to decode the base layer and a decrease in the level of FEC resiliency and respective transmission cost. Even though the ALD variants in crew have a lower level of viewable quality in comparison to city and soccer, ALD-IER-2 still outperforms MDC-SD and has the highest levels of viewable quality of all streaming models for crew.

In our evaluation the percentage of viewable frames, quality transitions over time and maximum achievable quality per model for crew and harbour were consistent, while for soccer these results were similar to city. This highlights how the selection of STF and the underlying level of network loss mandates the maximum level of achievable viewable quality. Similar variations in the maximum achievable PSNR and transmission costs per model were noted as GOP values decreased. Thus further illustrating that our techniques and models are able to provide levels of consistent high quality viewing, with lower transmission cost irrespective of clip type.

While the results shown thus far use well-known low resolution test clips, we also undertook High Definition evaluation of our techniques and models. A sample of the HD results are provided in the next section.

4.4.4 Evaluation Results for HD Content

In this section, we present a sample of our high definition (*HD*) evaluation results. For our HD evaluation we use a trailer for the "Sintel" movie [128]. Sintel is an independently produced animated short film, initiated by the Blender Foundation, containing both slow and fast moving sequences. The Sintel trailer is 52 seconds in duration and contains 24 frames per second. Similar to our low resolution evaluation, for Sintel we encode an eight layer SVC stream with 1,253 frames in HD using three resolutions 854x480 (*480p*), 1280x720 (*720p*) and 1920x1080 (*1080p*); using a 2,3,3 quality to resolution ratio and QP values as per Table 4.17.

Table 4.17: Sintel QP values per layer for all GOP values

Resolution	QCIF		CIF			4CIF		
Layer	BL	2	3	4	5	6	7	8
QP Value	34	28	33	30	28	35	32	30

The streaming models, SVC, MDC, MDC-SDP, MDC-SD, ALD and ALD-IER-2, are simulated. ALD and ALD-IER-2 are allocated an STF value of 6 for each of the four GOP values. Table 4.18 illustrates the maximum achievable PSNR, the transmission megabyte-cost for each adaptive scheme for quality layer 8 and the encoding time ($Encoding_T$) in hours for each GOP value. As can be seen, as the GOP value increases, overall PSNR decreases. There is a noticeable increase in encoding time from GOP of one to GOP of eight, but minor increases in encoding time for higher values of GOP. There is an approximate 80% decrease in transmission cost for SVC as GOP increases from a value of one to thirty two. There is an average of 150% increase in transmission cost from SVC to MDC, with an increase of approximately 40% from SVC to ALD. ALD-IER-2 mandates a 2% increase in transmission cost above ALD.

Table 4.18: For each of GOP values: maximum achievable PSNR for Layer 8 (in dB), transmission megabyte-cost for the Sintel HD clip at quality layer 8 for each adaptive scheme and the encoding time ($Encoding_T$) in hours. All MDC variants, MDC, MDC-SDP and MDC-SD, have the same transmission cost, so only MDC is shown

GOP	PSNR	SVC	MDC	ALD	ALD-IER-2	$Encoding_T$
1	49.6	49.9	137.3	69.8	70.8	1.16
8	49.1	16.9	44.0	23.3	23.7	17.37
16	48.3	12.5	32.3	17.1	17.4	20.33
32	47.4	10.4	26.8	14.2	14.4	21.25

Figure 4.19 presents an example of the percentage of viewable frames for each of the six streaming models for each of the four GOP values for the Sintel HD clip. Over all GOP values there are noticeable levels of SVC and MDC in the lower layers, with high levels of SVC, 60%, in the non-decodable range (layer 0) for a GOP value of thirty two. As GOP value increases, MDC-SD, ALD and ALD-IER-2 increase their percentage of viewable frames in the higher layers, indicating an increase in viewable quality. While

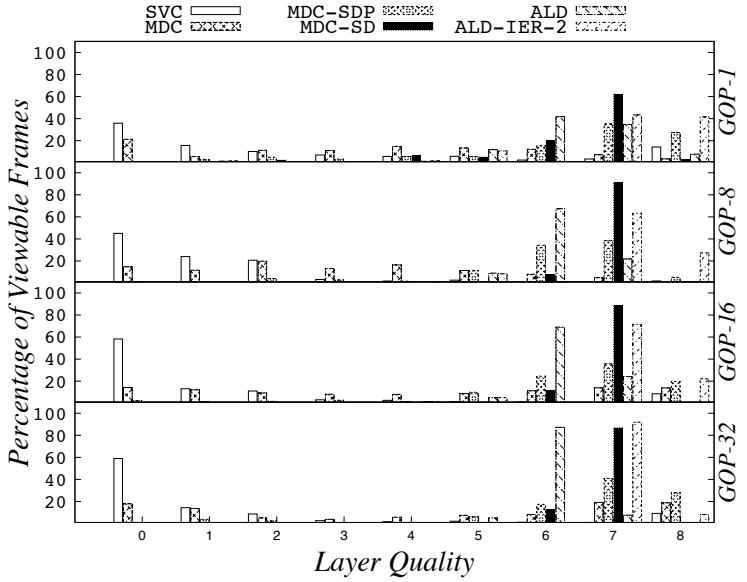


Figure 4.19: An example of the percentage of viewable frames for each of the six streaming models for each of the four GOP values for the Sintel HD clip with a 10% packet loss rate.

these increases demonstrate improved quality, we must first determine the transitions occurring between layers of adjacent frames to fully evaluate the benefits provided by our techniques. These plots provide similar results to our low resolution evaluations, where our techniques IER and SD mandate a greater percentage of viewable frames in the higher layers.

Figure 4.20 plots ten-second examples of the stream quality transitions for each streaming model of the sintel HD clip with a GOP value of 1 (Figure 4.20a), GOP value of 8 (Figure 4.20b), GOP value of 16 (Figure 4.20c) and GOP value of 32 (Figure 4.20d) with a 10% packet loss rate. For a GOP value of 1, there are noticeable levels of variation for all streaming models, but the range of transitions is higher for SVC and MDC, with the remaining models mandating lower transition ranges predominately in the higher layers. For GOP values of 8 and higher, the effects of inter-frame dependency result in increased degradation to the achievable quality of SVC and MDC. While the packetisation benefits of SD mandate increased consistency and stability in the level of viewable quality for longer periods of time. MDC-SD, ALD and ALD-IER-2 have near continual achievable quality over the duration of the clip for a GOP of thirty two. 92% of the viewable frames of ALD-IER-2 are at layer 7, with the remaining 8% at maximum viewable quality (layer 8), consequently out performing all other models.

It is important to note how the incremental increase in viewable quality provided by ALD-IER-2 mandates only a 2% increase in transmission cost as illustrated in Table 4.18. While ALD provides a reduction in transmission cost relative to MDC of approximately 46% for each of the respective GOP values. Thus validating the results seen for our low resolution evaluations and confirming that the benefits provided by ALD and our optimisation techniques are beneficial irrespective of clip type, encoding

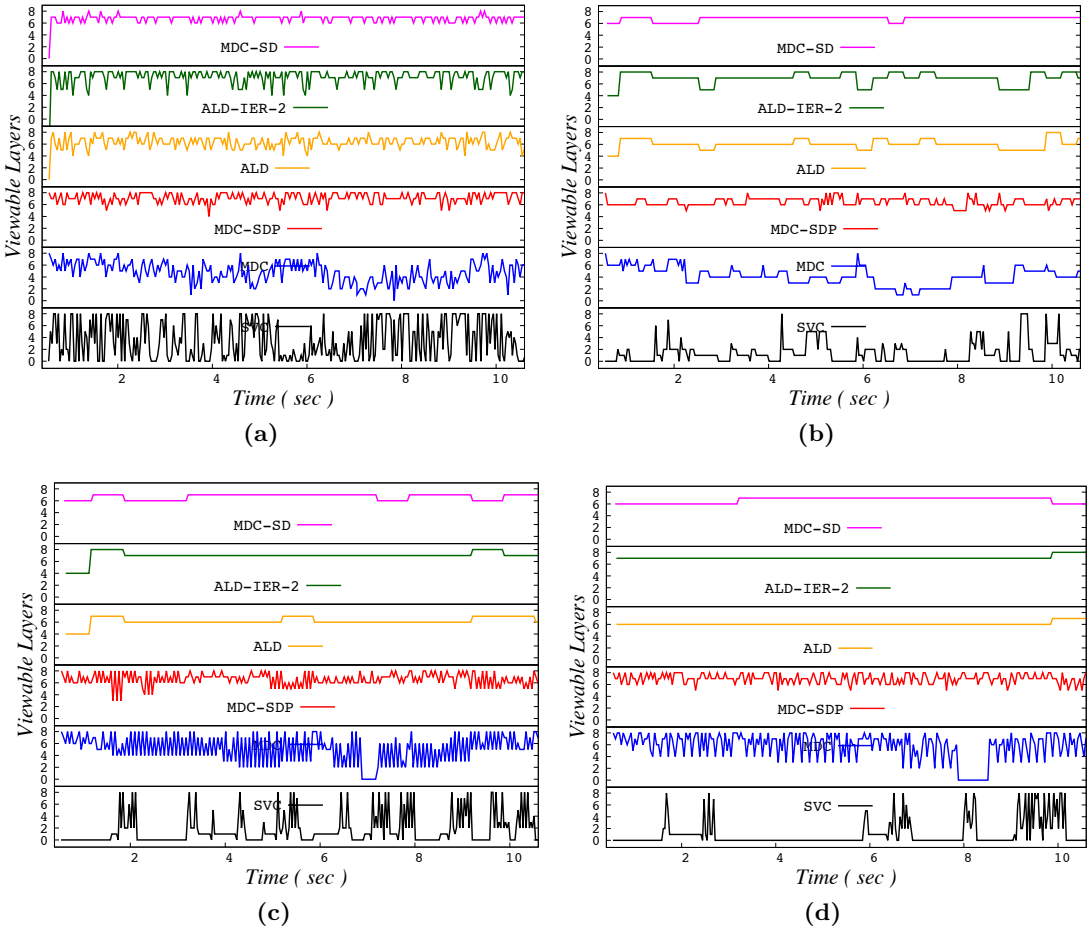


Figure 4.20: Ten second examples of the stream quality transitions for each streaming model of the Sintel HD clip with a GOP value of 1 (a), 8 (b), 16 (c) and 32 (d) with a 10% packet loss rate

demands or underlying resolution requirements.

4.5 Conclusion

In this chapter, Adaptive Layer Distribution (*ALD*) is proposed as a novel multifaceted approach to media streaming optimisation. ALD section thinning enables the reduction of the total streaming overhead while IER and section distribution improve ALD error resiliency to loss. Hence, ALD strikes a balance between stream quality and bandwidth efficiency. In our evaluation results we have seen how by combining MDC and our packetisation techniques (SDP and SD) that viewable quality can be noticeably improved without increasing the transmission cost of MDC. We noted how ALD sustains this increase in viewable quality while reducing MDC overhead. And finally, we viewed how, as GOP values increase, our techniques provided near continuous levels of maximum viewable quality when faced with high levels of network loss.

In chapter 5, we will investigate two outstanding issues with ALD:

1. By utilising STF, ALD provides a means of defining acceptable levels of FEC error allocation, based on the lowering of the cumulative overall transmission cost. But STF is a static metric and is calculated independent of the level of measured transmission network loss. Because of this, and dependent on STF value, high levels of network loss will negate the benefits of STF. Thus metrics which also consider the level of network loss will need to be evaluated.
2. As mentioned in Section 4.1.1, ALD has a higher transmission cost, than MDC, for users requiring lower layer quality. A model for reducing lower layer transmission cost will need to be considered.

Chapter 5

Streaming Classes (SC)

As we have seen, each of the scalable schemes contain known design issues. While the low transmission overhead is a benefit of SVC, the prioritised hierarchy and its dependency on the base layer is its greatest weakness. As we have highlighted, network transmission issues can affect all packets, and lower layer loss in SVC is detrimental to stream quality. MDC mitigates the effects of loss by overly increasing transmission cost, especially for users interested in lower layer video quality. More importantly, this transmission overhead represents a huge burden on users with limited bandwidth or device capabilities. This overhead is even more overwhelming when videos are encoded with a large number of layers to accommodate the existing diversity in mobile device capabilities [144]. ALD provides the framework to achieve the high levels of adaptable stream quality promised by SVC, but the transmission byte-cost of devices requesting lower layer decoding is dependent on stream encoding and the ALD selection value for STF. However ALD mandates a high level of transmission cost for devices requiring lower layer streaming.

To alleviate these issues we now propose Streaming Classes (SC), a novel transmission framework by which streamed media, either layered or description-based, are selectively grouped together to provide a means of reducing lower layer transmission byte-cost, while maintaining high levels of viewable quality for all users, irrespective of the layer requirements and original streaming model. In this manner, SC selects a number of layers from the underlying media stream, which best suit user requirements, and groups these layers together into *classes*. Figure 5.1 illustrates where the six layer SVC example illustrated in Figure 2.5, from Section 2.2.1, is grouped as three hierarchical SC classes, where C1 denotes class one and contains the lowest layers, C2 denotes class two and contains the mid-range layers and C3 denotes class three and contains the highest layers. Similar to SVC, the SC classes contain a prioritised hierarchy, such that classes containing lower layers are needed to decode classes containing higher layers. For layered coding, SC will introduce a minor increase in the transmission cost

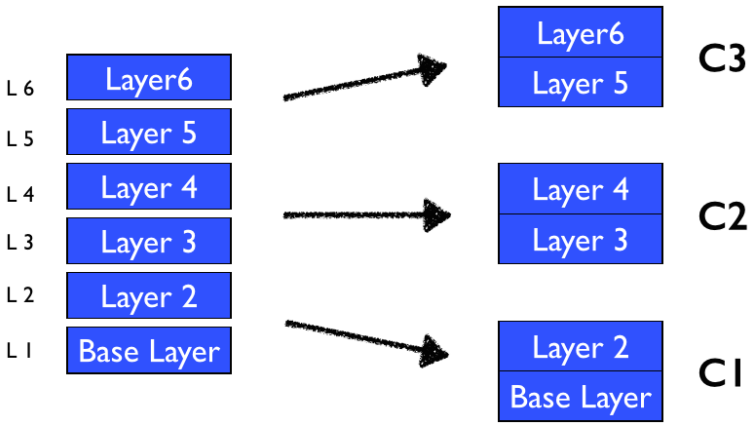


Figure 5.1: Example of a six layer SVC stream grouped as three hierarchical classes.

to provide a sufficient level of resilience for all layers, thus adequately negating the effects of packet loss. While for description-based models, SC will provide a marked reduction in transmission cost by selectively reducing the error resilience of the specific layers that are requested. For ALD this will provide a means of reducing the lower layer transmission cost, as highlighted as an outstanding issue of ALD in Section 4.5. In description-based encoding, the main transmission unit is the description, while for layered models the transmission unit is a layer. SCs reduce the transmission overhead by redefining the transmission unit as a selected set of section segments from different layers. As we have seen, selective packetisation of streaming data provides control over the effects of network loss on achievable quality, especially for the higher layers. With SC we introduce this control to lower layers in the stream especially for devices, as perviously mentioned, with limited bandwidth or device capabilities.

SC is a hybrid mechanism, which uses the description structure of description-based models, and the prioritised hierarchy and layer intra-dependency of layered models to reduce transmission cost and maintain the viewable consistency of stream quality. SC is utilised to re-distribute the grouping mechanism of description-based models, i.e. one section from every layer in a description, and provides a mechanism to group layers and sufficient levels of error correction, to provide acceptable levels of transmission cost for all layers, irrespective of scalability. Thus SC provides a means of grouping layers together to mitigate network loss and to reduce transmission cost for devices requesting a lower level of achievable quality. In this manner, and by utilising the prioritised hierarchy of layered coding, SC provides classes based on the underlying combination of temporal, spatial and fidelity scalability, as well as the layer intra-dimensionality that is removed from description-based adaptive stream models.

5.1 Design Principles

As we have seen in the design process of our subjective testing, Section 6.1, the number of design options to consider can greatly increase the complexity, and the underlying decisions, mandated during encoding, packetisation, transmission, decoding and subsequent viewing (we call the process from encoding to subsequent viewing as a *stream flow*, while each of the individual steps from encoding to subsequent viewing we call *stream elements*). Examples of these design options include encoding requirements (number of layers, QP values, GOP size, etc.), expected network loss rates, video clip types, user quality selection and specific design options for the underlying streaming model (STF for ALD as an example), as well as the interaction and interdependency between all these design options. It is important to understand that decisions in one stream element will impact other stream elements, *e.g.* encoding decisions determine the stream bitrate, which changes the number of packets being transmitted, which in turn affects transmission issues, such as congestion and the loss rate, which influences the decodability of the stream and subsequently the level of viewable quality of the stream at the user's device.

Based on what we have learned from the evaluation of our initial stream models, SDC and ALD, as well as the feedback from our subjective testing, we view SC as more than just a means of reducing transmission cost for lower layer streaming, but as an architecture to allow one to appreciate that the achievable quality of a stream is dependent on so many factors. One design based on a number of defined variables can not solve all stream flow issues that can occur. Thus the goal for SC is to consider all aspects of the stream flow and provide adaptive mechanisms for determining and supporting consistent high levels of viewable quality for all users.

We begin by providing a brief overview of the five primary design principles for streaming classes:

1. Based on the observed levels of network loss, the *Class Packet Loss Rate* principle is used to determine an expected loss rate value, LR_{C_i}, for each class, C_i. The LR_{C_i} is used by each of the following four design principles. The goal of this principle is to mandate consistency of quality over time, *e.g.* reduce the frequency of quality change that occurs between adjacent frames, or GOPs, due to variations in network loss rate.

2. The *Layer Allocation and Hierarchy* principle defines how to group individual SVC layers together based on the interdependence that exists between different classes because of the inherent SVC layer prioritisation. The composition, or structure, of the individual classes is dependent on the initial SVC encoding, the GOP value selected, user requirements, and the defined combination of spatial, temporal and quality scalability. One example of layer allocation is to group lower

layers together into a single class, so as to reduce transmission cost for devices requesting lower quality streaming or for devices on constrained networks.

3. The principle of *Error Resiliency* is utilised by each class to add sufficient levels of error resilience to every layer to combat defined levels of network loss. To reduce overall transmission cost we implement an inverse relationship between layer level and error resilience, i.e. the higher the layer level the lower the error resilience. This relationship provides the more important layers per class with higher levels of resiliency.

4. Similar to layered and description-based models, SCs are created with a hierarchical nature such that the more SCs received by the user, the better the quality of the decoded video. The principle of *Streaming Class Structure* defines each group of layers as a prioritised class, i.e. classes containing lower layers are more important to viewable quality than classes containing higher layers, as higher layers in the SVC hierarchy are predisposed to be dependent on lower layers. Thus the level of viewable quality is dependent on the number of classes received, as well as the number of decodable layers per class.

5. The principle of *Class Packetisation and Granularity of Packet Data Byte-size* is utilised so as to provide layer equality per class, i.e. by utilising Section Distribution (*SD*), each packet per class contains a segment of every layer per class, consequently viewable quality is dependent on the number of packets lost rather than on the contents of the packets that are lost.

Thus these design principles provide a selection of options by which both layered and description-based models can increase viewable quality. These options range from:

1. Encoding decisions such as GOP value, and spatial, temporal and quality scalability.

2. Transmission choices such as selective packetisation, the allocation of layers to classes and intuitive adaptation techniques which adjust to network conditions *e.g.* packet loss and error resiliency allocation.

3. User requirements such as the quality of the media clip requested, and the corresponding maximum layer value and associated lower layer dependencies required to decode the requested quality.

In the following sections, each of the five primary design principles shall be examined in further detail.

5.1.1 Class Packet Loss Rate

The principle of "Class Packet Loss Rate" is utilised to determine the loss rate of

Table 5.1: Notation and Definitions

n	The number of SVC layers, L, per Group of Picture (GOP)
m	The number of SC classes per GOP
l	Integer value corresponding to the layer number of L_l
GOP_{number}	The number of frames per GOP
GOP_i	The i^{th} GOP
μ	Mean or average network loss rate
μ_{GOP_i}	Loss rate for the i^{th} GOP
μ_X	Loss rate for the duration of the stream. Also known as the Loss rate from GOP_1 to GOP_N, where N denotes the maximum number of frames per GOP
LR_{C_i}	Loss rate for Class i
PL_{C_i}	Packets lost for Class i
T_{C_i}	Packets transmitted for Class i
$\mu^{\max}_{GOP_i}$	Determined maximum loss rate for the i^{th} GOP

each class, LR_{C_i}, based on the overall loss rate in the network. The packet loss rate for media streaming is primarily based on the number of packets unavailable during decoding of a specific frame or GOP. The unavailability of a packet during decoding can be caused by numerous issues, these include packet delay [145] (due to data transmission over multiple network links), buffering [146] (bottlenecks in the network routers) and congestion; both rate control [147] (slowing throughput to reduce loss) and packet dropping [148] (non transmission of packets to reduce loss). In general the percentage of packets lost in the network will not equate to the same percentage of lost transmitted data, as the transmitted data tends to be composed of different sized packets and the percentage of lost transmitted data is dependent on the distinct packets that are lost. It is important to consider this relationship when we examine the effects of network loss.

In the literature, numerous mechanisms exist to determine the current network loss rate, example of these include Server-side such as Realnetwork Helix Mobile Media Server Rate Control [149], Web Server TCP packet loss determination for QoS [150] and IBM Unix network Performance analysis [151], Client-side such as Quality-Oriented Adaptation Scheme (QOAS) [152], Application-level Estimation [153], and NetPolice: loss rates in ISPs [154], and from nodes within the network, such as iPlane: an information plane for distributed services [155], RON: resilient overlay networks [156], and Queen: estimating packet loss between arbitrary internet hosts [157].

For our adaptation to network loss, we need to consider two issues

i. How the variation in network loss over the duration of a media clip influences the loss rate for any given individual GOP:
   If we were to assume that the network loss rate is consistent over the duration of a media clip, then by simply transmitting a percentage of redundant media

proportional to the average network loss rate this would enable consistency of viewable quality. But in reality, variations in network loss rate will occur. Each GOP provides adequate statistical data to infer the current loss rate per GOP as well as the overall variation of viewable quality. Thus for each GOP, GOP_i, we first determine the expected network loss rate, μ. μ is the mean or average loss rate over all previously transmitted GOP. This mean value does not mandate that all future GOP would encounter the same μ value but infers that future GOP shall include redundancy that accommodates to μ. The value of μ is affected by both the loss of an individual packet and the loss of groups of adjoining packets, known as bursty loss (we can define all loss as bursty loss if we include the loss of a single packet as a burst rate of 1) and as such the loss rate of the next GOP can range from 0% to 100% depending on the level of bursty loss and the number of frames per GOP.

In a binomial distribution, the variation that occurs from the mean value is limited in range. In network loss, the instantaneous GOP loss rate can range from 0% to 100% but when averaged out and taking minimum and maximum values, the variation is also limited in scope (between adjacent GOP). It is important to note that some outlying minimum and maximum values can occur, but these values are dependent on a number of factors, such as number of frames per GOP and the levels of traffic over the transmission network. In our evaluation we illustrate how the variation in packet loss that can occur per GOP is dependent on the number of frames per GOP. Over the duration of the media clip, we determine the minimum and maximum packet loss value per GOP and show how the interval between these packet loss values decreases as the number of frames per GOP increases.

If we assume that the variation that can occur is limited in scope and is similar in nature to a binomial distribution, we can utilise σ_μ, which equates to the square root of the variance to cap the level of error correction required. In this manner, μ and σ_μ are used to define a statistical maximum threshold on the expected network loss rate for the current GOP, which we define as the maximum expected loss rate, $\mu^{\max}$, for a given GOP_i or $\mu^{\max}_{GOP_i}$. $\mu^{\max}_{GOP_i}$ limits the increase in transmission cost while providing an acceptable level of adaptive error resiliency.

ii. How the loss rate for an individual GOP determines the loss rate per class:
The viewable quality of a single non scalable stream, *e.g.* one resolution, one frame rate, and one quality level, is based on the average loss rate μ over a predetermined length of time, *e.g.* from the start of the stream to the current point in time. While for scalable streams, *e.g.* layered/description-based models, the viewable quality is dependent on the average loss rate, μ, over all transmission units (layer, description, section, packet - dependent on the underlying streaming

model), *e.g.* the distribution of μ over each of the transmission units and how this allocation of loss affects the decoding of the underlying layered data. Thus, the defined loss rate for each individual transmission unit is based on the percentage loss rates of all other transmission units and μ. This is also true for each class in SC. The μ is mean over all classes and does not mandate the loss rate of each individual class will equal μ.

Hence the overall loss rate, with respect to each class, can be defined as the summation over the number of packets lost per class divided by the number of packets transmitted per class divided by the number of classes or

$$\mu = \frac{\sum_{i=1}^{m} \frac{PL_{C_i}}{T_{C_i}}}{m} \tag{5.1}$$

It can be seen that for a given loss rate over a number of classes, the average loss rate per class can vary from zero loss, to the transmission cost of all classes multiplied by the overall network loss rate, *e.g.* assuming three classes of equal transmission cost, a μ of 10% could mandate an individual class loss rate, LR_{C_i}, of between 0%, minimum loss rate $LR_{C_i}^{\min}$, and 30%, maximum loss rate $LR_{C_i}^{\max}$. If we again assumed three classes and 10% loss, where C_1 has a transmission cost of 100 bytes, C2 has a 1,000 byte cost and C3 has a 10,000 byte cost. Then the loss rates range from a minimum of 0% for all classes to a maximum of 11.1% for C_3 and 100% loss for both C_1 and C_2. Total transmission loss at 10% is 1,110 bytes and this is larger than C_1 and C_2 combined. The effects of individual class loss rates are investigated in the *error resiliency* principle and examples of the variation that can occur between classes of the same GOP are illustrated in the evaluation section of this chapter.

For each GOP, GOP_i, the loss rate of each class, LR_{C_i}, is passed as an argument to the other four primary design principles. μ can be determined from these loss rates or can be passed as a separate argument.

i. For layer allocation, this value will assist in determining the layer distribution to classes during periods of high loss rate or bursty loss.

ii. For error resiliency, this value will determine the forward error correction (FEC) allocation rate for the highest layer in each class.

iii. For class structure, this value will assist in determining the number of classes being transmitted. During moments of high network loss, classes containing higher byte-cost layers may be dropped, so as to reduce congestion.

iv. For class packetisation, this value will assist in determining the byte-size and layer

Table 5.2: Encoder output for a general encoding scheme for the crew video, composed of two resolutions, 2 quality levels and three frame rates.

Layer	Resolution	Frame Rate	(D,T,Q)
0	176x144	7.5	(0,0,0)
1	176x144	15	(0,1,0)
2	**176x144**	**30**	**(0,2,0)**
3	176x144	7.5	(0,0,1)
4	176x144	15	(0,1,1)
5	**176x144**	**30**	**(0,2,1)**
6	352x288	7.5	(1,0,0)
7	352x288	15	(1,1,0)
8	**352x288**	**30**	**(1,2,0)**
9	352x288	7.5	(1,0,1)
10	352x288	15	(1,1,1)

segment allocation per packet.

Each of these items will be further explained in their corresponding section.

5.1.2 Layer Allocation and Hierarchy

The definition of streaming classes would vary depending on the scalability techniques adopted during video encoding. To illustrate the design options of defining SCs, we consider a generally encoded video using a combination of spatial, temporal, and quality scalability for the well-known crew video [127].

The output of the encoder is shown in Table 5.2 in which the (D,T,Q) tuple, (D for dependency_id - spatial resolution dependency), T for temporal_level and Q for quality_level) represents the level of dependency in the three scalability dimensions. A higher index in any of these fields indicates that decoding the corresponding layer requires receiving all previous layers with a lower index in the same field. For example, decoding layer 7 (1,1,0) implies that we should receive all layers including lower values in the D and T fields; i.e. the streaming client should also receive layers 6 (1,0,0), 1 (0,1,0), and layer 0 (0,0,0), assuming that inter-layer dependency is enabled in the encoding process (CGS selection). It is worth noting that if scalability is performed in one dimension (or MGS encoding selection), the reception of a higher quality video mandates the reception of all lower layers. For example, layers 0, 1, and 2 may be considered as temporally scaled video with QCIF resolution (176x144) and one quality level. Hence, for videos encoded with a single scalability dimension, there exists one option for layer aggregation, which is grouping subsequent lower layers.

The presence of two types of scalability in the encoded video creates more design options as shown in the previously presented combined quality-resolution scalability. If these two dimensions include a temporal dimension, the layer grouping can be per-

formed in different ways. To illustrate, consider layers 0 to 5 in Table 5.2. This subset of layers can be considered an example for a combined temporal-quality encoded video at QCIF resolution. In this case, two basic design options are envisioned for grouping:

- SCs are created from layers having the same quality index. For example, layers 0 to 2 and layers 3 to 5 represents two streaming classes respectively.
- SCs are created based on frame rate (temporal dimension). For example, grouping layers having 7.5fps to form the lowest SC, layers played at 15fps as a second streaming class and layers played at 30fps as a third streaming class.

Which option to choose would depend on the underlying bitrate of the video and the individual user's requirements. The transmission cost of both options is the same, as an equal number of layers are transmitted by both options. Option one can only vary the frame rate once all classes are received as layer allocation is based on quality levels, while the second option is more adaptive to users requesting lower frame rates, as each frame rate is contained within a distinct class.

Similarly, a video encoded with temporal and resolution dimensions would have two basic design options for creating the SCs. To illustrate, let layers 0 to 2 and layers 6 to 8 represent a video encoded with temporal and resolution scalability. Hence, the two design options would be aggregating based on frame rate or based on video size. The former would also be favourable because the latter would produce a high data rate in the lower streaming classes. Additionally, the former would be more appealing as the higher frame rate in the aggregated lowest class would result in a better quality video. Last but not least, consider an encoded video over the three scalability dimensions as shown in Table 5.2 and note the many SC aggregation options that would be possible. However, as established in our presentation, aggregating layers with different resolutions contradicts with the principals of SC (increasing transmission cost for lower layer streaming). Hence, aggregation over the temporal and quality scales remain the two possible options.

To this end, the layer aggregation criteria may include more than one rule. For example, the aggregation of several layers over one of the scalability dimensions can be accompanied with another constraint on other factors such as the resultant transmission byte-cost of the aggregated layers. Another possibility is that the aggregation criterion considers target bit-rates matching existing bit-rates in real networks. To further illustrate, if the aggregation of the three quality layers for a single resolution would result in a data rate exceeding a target bit-rate, one can alternatively use more than one streaming class for this set of layers by allocating the lowest of them as one streaming class and the remaining two layers as a second streaming class. The proposed one-two splitting is suggested to create a balanced set of streaming classes.

As illustrated in the adaptation to network loss section, we can also consider the

underlying loss rate in the network while determining the layer allocation per class. The effects of congestion can be reduced, in some part, by a reduction in stream bit-rate. Thus by removing the allocation of high byte-cost layers, we can reduce the level of network loss. Also careful consideration is needed when we consider the prioritised hierarchy of the classes, *e.g.* how important is the content of this class, and as such the maximum achievable quality of this class, to all other classes.

Thus to recap, the number of layers and layer allocation per class can be based on a number of options, these include but are not limited to:

i. *adjacent layers*: an i number of adjacent layers. Subsetting the total number of SVC layers into classes, where each class contains an equal number of layers.

ii. *temporal, spatial or fidelity layering*: creating classes based on distinct frame rates, resolutions, quality levels, or combinations of same. Combining higher layer data with lower layer data, may be counter productive to the reduced transmission cost objective of SC.

iii. *layer intra-dependency*: the intra-layer dependency of requested layers.

iv. *stream layer selection per requesting domain/network location*: the layers being requested per individual domain or co-located networks. Classes can be created for specific domains, or network regions, which reduce overall transmission cost to the specific layers requested by these geographical locations.

v. *the number of distinct quality layers being requested*: the number of distinct layers being requested. Classes can be created which are based on the specific layers required to decode quality layer q. Consequently removing the cost of transmitting un-required layers.

vi. *the number of users per quality layer*: grouping the most requested layers together. Creating classes based on overall user requirements. The greater the requests for a given layer, the higher the class prioritisation, or lower class number, for that layer.

vii. *based on bit-rate selection*: thresholds of available bandwidth and cumulative transmission byte-cost. Creating classes based on network throughput, constrained links or overall congestion rates. While not referencing scalable video specifically, but commenting on the reliable transmission of TCP traffic but using layered video to illustrate adaptation of quality dependent on available bandwidth, the TCP Rate Adaptation Protocol (RAP) proposed in [158] is a seminal paper in this regard. The authors of [158] investigate the rate-adaptation of layered media dependent on the timescale of round-trip times but maintain viewable quality for longer periods of time by using buffering to accommodate mismatches between transmission and consumption rates.

viii. *replication of layer data*: it is not infeasible that numerous classes may benefit from the allocation of the same layer, *e.g.* allocation of the base layer to numerous classes, with each of the classes based on a different spatial, temporal or quality scalability. It may be more cost effective, with respect to transmission cost for individual users, to transmit the individual layer as a single class, or allocated to numerous classes, so as to reduce the receipt of higher layers from unwanted dimensionality.

As previously stated, the goal of SC is to consider all aspects of the stream flow and then choose which option(s) maximise overall viewable quality. Thus which "Layer Allocation and Hierarchy" option(s) to choose at runtime will be governed by the effects of the chosen option(s) on the other design principles and overall stream quality.

5.1.3 Error Resiliency

Irrespective of transmission unit, the transmission cost overhead between layered and description-based models is fundamentally the level of error resilience allocated. Thus the goal of the error resiliency principle is to provide balance between transmission cost and achievable quality. As previously described, section distribution (*SD*) offers a means of delivering packet equality for description-based models, such as MDC and ALD. It is this packet equality and the incremental levels of error allocated per SVC layer, as presented in Section 4.3.2, that provides consistency of viewable quality and mandates that viewable quality is dependent on the number of packets lost rather than on the contents of the packets that are lost. By implementing SD in the streaming class model, we can mandate that the number of decodable layers for class i, C_i, is dependent on the level of error allocation per layer and the corresponding class loss rate, LR_{C_i}.

We have already seen how the loss rate of class i, LR_{C_i} is dependent on both the μ and the loss rates of all other transmitted classes. Hence the level of error resiliency per layer must take into consideration

i. the priority of the layer in the original SVC hierarchy. Lower layers per class contain higher level of error correction. This allocation of error correction provides an incremental increase in quality, dependent on the level of network loss, which is consistent with description-based streaming models and also a graceful degradation in viewable quality as network loss increases.

ii. the variation that can occur in both μ and the loss rates of all other transmitted classes. As seen, the variation that can occur between classes for each GOP is relative to the loss rate of all classes, thus this variance, or deviation, must be taken into consideration.

iii. the maximised consistency of quality for this class and for all classes which are dependent on this class. In SVC, higher layers are dependent on the achievable quality of lower layers and SC is similar in this regard. Dependent on the allocation structure of the classes, higher classes will predominately be accessible only when all layers in a lower dependent class are decodable. To benefit from this dependency, all layers per class must contain error resiliency, and higher levels of error resiliency may be required in lower classes to maximise quality for all classes.

iv. the bitrate of the allocated layers. Low bit rates per layer, which can occur with larger GOP values and for lower frame rates, may require more than the minimum error resiliency allocation to recover from loss, as the byte allocation per packet may be too small.

v. the addition of increased error resiliency will impact on transmission cost, viewable quality and network loss. As we determine error resiliency based on predetermined loss rates, will increasing the error resiliency per layer impact on the current levels of network loss. We can consider three examples which illustrate the varying levels of error allocation, relative to current network loss rate.

    i. Define the error allocation rate for the highest layer in class i to be equal to the current level of network loss, i.e. LR_{C_i}.

    This option will in some instances only covers lower loss rates than the current rate, *e.g.* assuming a loss rate of 10%, if we add 10% additional data to the layer contents, we effectively increase the transmission cost to 110%. Assuming the loss rate does not increase with increased traffic, then the effects of the same 10% loss rate will equate to 11% of the 110% being transmitted, consequently only 99% is decodable and the highest layer is unviewable. Note, that this disparity will be dependent on the packetisation rate, as we shall see later. If the increased error rate can be allocated to the same number of packets as the original layer data, then even with 10% of the packets lost, the layer is still decodable.

    ii. Define the error allocation rate for the highest layer to be equal to the current level of network loss plus the standard deviation that can occur for this loss rate, i.e. $\lceil LR_{C_i} + \sqrt{LR_{C_i}} \rceil$. We define this as the allocated loss rate value for the maximum viewable quality or $LR_{\max}$. This would equate to 10% + 3.17% respectively or a rounded up value of 14% FEC for the highest layer, assuming a 10% packet loss rate.

    This higher level of error allocation covers some of the packet loss rate variation over time but may still be susceptible to loss during high bursty conditions.

iii. Define the error allocation rate for the highest layer to be upper bounded by a maximum threshold based on the expected network loss rate required to recover from the current level of network loss, which we will define as $LR_{C_i}^{\max}$. This loss rate will need to consider both the initial loss rate, LR_{C_i}, and the loss rate that will affect our newly added FEC, $\sqrt{LR_{C_i}}$. An algorithm to determine this value is provided in Eq 5.2. $LR_{C_i}^{\max}$ denotes the current determined loss rate for the maximum viewable quality $\lceil LR_{C_i} + \sqrt{LR_{C_i}} \rceil$.

$$LR_{C_i}^{\max} \geq \frac{LR_{\max}}{1 - LR_{\max}} \tag{5.2}$$

Finally as per our example, if we assume $LR_{\max}$ is 14%, we can determine the minimum additional FEC value required to recover for this level of maximum packet loss, as shown in Eq 5.3.

$$\begin{aligned} LR_{C_i}^{\max} &\geq \frac{14\%}{1 - 14\%} \\ LR_{C_i}^{\max} &\geq \frac{14\%}{86\%} \\ 17\% &\geq \frac{14\%}{86\%} \end{aligned} \tag{5.3}$$

Such that for a packet loss rate of 10%, a standard deviation of 4% and a minimum recovery rate (based on packet loss + standard deviation) of 3%, we can define a $LR_{C_i}^{\max}$ of 17% FEC.

This option covers both the variation in packetisation rate as well as minor variations in network loss rate.

vi. the original streaming model. For SVC the transmission unit is a complete layer and the packetisation options outlined above take into consideration the entire layer plus allocated error resiliency. Thus small incremental increases in error resiliency are easily accommodated into the packetisation option. For description-based models, the transportation unit is either a description (MDC), a section (SDP) or a packet (ALD), such that the packetisation is defined based on the size of the underlying unit. Additional error resiliency is then based on the addition of a predetermined section size based on the value of a layer index (plus STF for ALD). These section sizes are static in nature and the addition of a section adds a pre-defined level of increased transmission cost to a class, *e.g.* for ALD, assuming

5 layer and an STF of 3, this defines that each section in the highest layer, layer 8, contains 12.5% of the overall transmission cost of that layer. As ALD adds additional sections to define the structure of increased error resiliency, each increase in section for layer 8 will mandate an increase of 12.5% of the layer byte size. Minor increases can be achieved by using Improved Error Resiliency (*IER*), but even then the increase in overall cost can be quite high, and possibly higher than the required LR_{C_i} percentage. A minor increase in error resiliency can be allocated, but as ALD uses sections sizes to define SD packetisation, this increase must be distributed over all sections or must equal the defined SD packetisation sizes.

To sum up: the allocation of error correction must take into consideration the structure of the original stream model. With layered streaming the increase in error correction can directly equate to the expected network loss rate. While for description-based models, the description structure is defined with default levels of error resiliency, *e.g.* based on predefined section sizes, which may overly increase error correction levels and thus elevate transmission cost.

So to recap, for each class; each layer must contain a minimum level of error resiliency, consistent to the LR_{C_i} determined by the "Class Packet Loss Rate" principle. Lower layers per class must contain incrementally higher levels of resiliency, to counteract the variation that can occur due to loss rates in other classes, thus providing graceful degradation in viewable quality as loss increases. Examples of allocation rates for $LR_{\max}$ and $LR_{C_i}^{\max}$ based on a packet loss rate of 10% are presented in Appendix C.

Once the error allocation rate for the highest layer per class is determined, the incremental increase in error allocation per lower layer must be determined. This can be defined on a fixed incremental increase, a variable level of increase (dependent on external factors, such as packetisation rate, layer allocation, network conditions, user requirements to name but a few), or can be based on the allocation rate of the adjacent higher layer plus a determined level of standard deviation, as used in our SVC evaluation results in Appendix C.

5.1.4 Streaming Class Structure

Similar to the layer structure in SVC, SC imposes a prioritised hierarchy based on the importance of the layers contained within each class. This class hierarchy defines which subset of classes is required to decode a requested layer. In this section, we explore how variations in class composition can be utilised to maximise viewable quality per class. Dependent on layer allocation, the viewable quality of higher classes are dependent on the higher prioritised lower classes. Such that the loss of a lower layer will impact the viewable quality of all or a subset, dependent on encoding dimensionally, of the higher

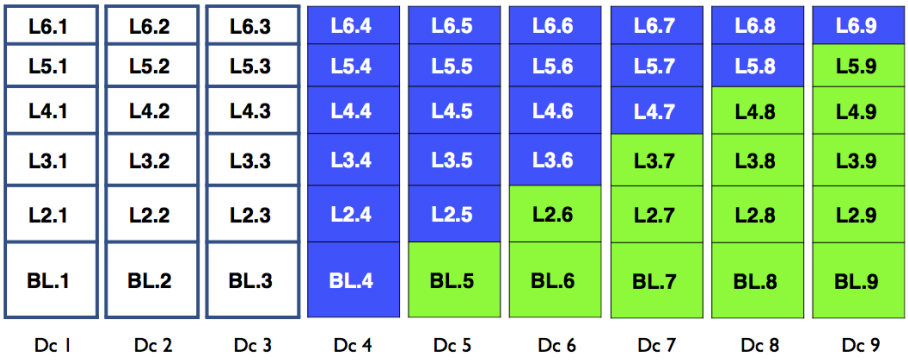


Figure 5.2: ALD with six-layer and an STF of 3

layers for all classes. Thus the goal of this principle is to maximise quality in all lower classes.

As we have seen, the loss rate per class can vary dependent on the overall loss rate, μ, and the loss rate of the other classes. While error resiliency provides a mechanism to allocated acceptable levels of error correction to individual layers, there exists a balance between sufficient error resiliency and increases in overall transmission cost.

We offer two class composition options by which to maximise viewable quality per class. Both options illustrate the balance between error resiliency and transmission cost and are based on the interdependence of the classes. For ease of illustration we present an example based on ALD, note that the same class composition option would also hold true for SVC and MDC (remember ALD with STF=0 is MDC). Figure 4.2 from Section 4.1.1 (reproduced here in Figure 5.2) illustrates a six layer ALD stream with an STF of three. We shall create three classes, based on layer adjacency, such that class 1 (C1) is composed of the base layer and layer 2, class 2 (C2) is composed of layer 3 and layer 4, and class 3 (C3) is composed of layer 5 and layer 6. The two class composition options, based on the defined class hierarchy, can be defined as:

i. *Independent Class Composition (ICC)*: With ICC, layer data of a class is contained only within the class. This is illustrated in Figure 5.3, where we can see that the layer data for each class is contained only within the defined class. In this instance the error allocation rate, as determine by the "Error Resiliency" principle, governs the overall effects of variations in the network loss. A higher level of error resiliency in the lower classes may be sufficient to recover from loss. ICC maintains packet equality per class, mitigates loss by allocating sufficient levels of error resiliency and maintains acceptable levels of transmission cost per class. Examples of ICC based on a packet loss rate of 10% are presented in Section 5.1.5.1 and Appendix C.

ii. *Increased Class Interdependency (ICI)*: ICI is an enhanced version of ICC, where additional lower class data is allocated to the higher classes to increase the viewable quality of the lower classes, and thus increase overall quality of the higher

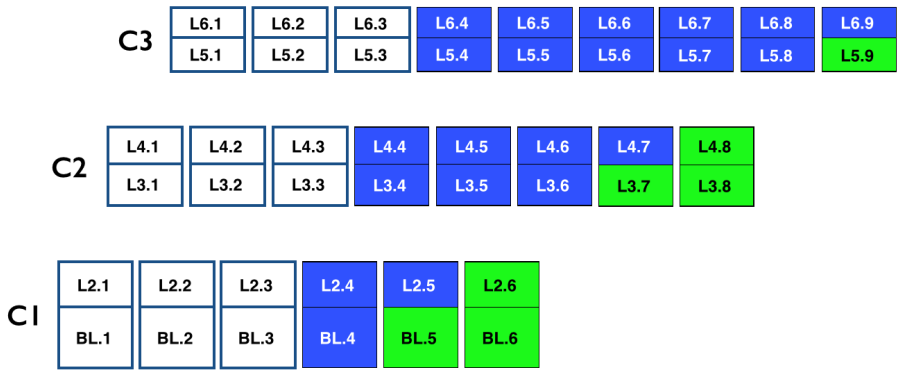


Figure 5.3: ALD streaming classes using the ICC class composition option - C1 denotes class one, C2 denotes class two and C3 denotes class three

classes. The initial error allocation rate as determine by the "Error Resiliency" principle, is utilised by ICC and then ICI assigns additional unallocated layer data from lower classes to the higher classes. This is illustrated in Figure 5.4, where we can see that two sections from the Base Layer and layer 2 are allocated to class 2 (C2), while a single section from the four lower layers are allocated to class 3 (C3), thus allocating all of the remaining FEC sections from Figure 5.2 across the higher classes. Note how the number of FEC sections allocated to the higher classes is consistent to the number of FEC sections allocated to the lowest layer of each higher class. While this example illustrates FEC allocation for description-based streaming, the same steps would be used for layered streaming, but the allocation rate would be based on the percentage of FEC in the lowest layer rather than a defined number of sections. Note how the FEC and layer data allocation in C1 in Figure 5.4 (ICI) and Figure 5.3 (ICC) are identical, as no additional data are allocated to the lowest class. ICC is beneficial when variation in the network leads to loss rates greater than the error allocation rate for a given lower class layer. In this manner, the availability of lower layer data in a higher class, may provide sufficient additional data to maximise the given lower layer. In this option, in addition to the lower layer data allocated to the higher classes, a lower level of error resiliency allocated to the lower classes may be sufficient to recover from expected network loss. A minor increase in transmission cost, relative to the level of additional lower class data, is to be expected.

Finally, the level of lower layer data allocated to the higher classes permits the higher classes to experience an equivalent level of loss in the lower classes while still permitting complete decoding of the higher class. The level of lower layer FEC allocated to the higher classes is adaptive to the needs of the stream flow. Examples of this FEC level include adaptation based on the levels of loss in the network, it may reflect the priority of a given lower layer or may be dependent on consistent decoding of a given higher layer.

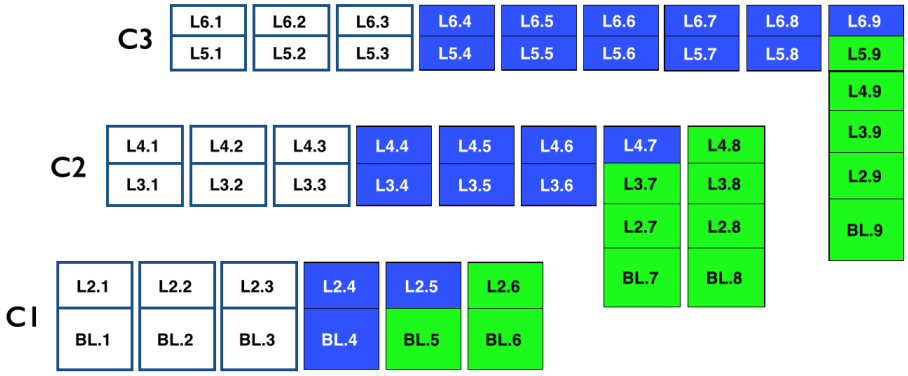


Figure 5.4: ALD streaming classes using the ICI class composition option - C1 denotes class one, C2 denotes class two and C3 denotes class three

Packet equality is maintained in the higher classes, as the lower layer data are distributed over all transmitted packets. For lower class subscribers, one additional benefit from the ICC composed higher classes, is that if the achievable quality in a lower class is marginally less than maximum, then the lower class user can subscribe to a subset of the higher class packets. Thus improving quality by receiving segments of lower layer data, while marginally increasing transmission cost by receiving segments of un-required layers. As ICC mandates that all higher classes contain lower class data, the lower class user can select packets from all higher classes, therefore benefiting from real-time data acquisition of lower layer data, without the delay of retransmission, which is highly beneficial to media streaming.

If the error resiliency allocation rate of ICI is sufficient in all classes, then the ICC allocation of lower class data to higher classes is an unacceptable and unrequired increase in transmission cost.

5.1.5 Class Packetisation and Granularity of Packet Data Byte-size

As we have seen, Section Distribution (*SD*) was initially designed to extend the concept of equal importance from description to the packet level per frame. SD allocates a segment of each section per description to a packet, as a result limiting packet loss to a portion of each description section, rather than being SVC layer or MDC description specific. In this manner SD creates a coping mechanism for both single packet loss and burst loss models. Thus achievable quality per frame is based on the cumulative number of received packets, with achievable quality directly reflecting the level of packet loss during transmission.

One additional benefit of SD is that all packets per frame are of equal byte size. This equality is provided in both packet byte-size and packet priority. Also as the number of frames per GOP increases, SD will provide data equality for all frames

within the GOP. In [45], the authors highlight that packets of dissimilar processing times, produce dissimilar transmission times. Such that by maintaining such packet byte-size equality, the order of packet delivery is improved. Hence, SD packet equality improves consistent delivery in network transmission.

While SD provides packet equality based on an equal allocation of segments for each layer per class, to provide increased resiliency to network loss we also consider the byte-size of each packet to determine if a byte-size threshold is beneficial to achievable quality. Let us assume for simplicity that a single frame contains eight SVC layers and each layer is the same byte-size, *e.g.* 540 bytes. Total transmission cost for this frame is 4,320 bytes, which would equate to a minimum of three packets, assuming a data size of 1,440 bytes per packet and the layers are allocated in increasing order until the packet data-sizes are full. Let us also assume a network loss rate of 10%, which in our example would equate to one lost packet or over 2.5 lost layers. Best case (B-C) scenario is where the data from packet three, *e.g.* layers 6, 7 and 8 were lost and layer 5 can be decoded, while worst case (W-C) scenario is where packet one, *e.g.* the base layer, layer 2 and 3 were lost and the frame is undecodable. With the current packet data byte-size allocation, 1,440 bytes, we incur the same level of layer loss for all packet loss rates from 1% to 33% inclusive.

By increasing the number of packets transmitting the layers, we can reduce the effects of the layer loss relative to the current loss rate. By increasing the number of packets to eight, we incur a minor increase in transmission cost, relative to the increased number of headers, while the effects of the 10% network loss is reduced to one packet, *e.g.* one specific layer. Each packet now contains 540 bytes, or one complete layer, and a single packet can incur a loss rate of between 1% to 12.5%. B-C for this simple decision increases viewable quality to layer 7, while in W-C the frame is still undecodable. With eight packets, an optimal threshold for number of packets has been reached. An increase in the packet numbers will not increase viewable quality, as each packet will subsequently contain either segments of multiple layers or a segment of a single layer, and the loss of a segment will negate decoding of the entire layer.

Consequently, we present two packetisation options for SCs, namely reduced overhead packetisation and improved resiliency packetisation.

1. *Reduced Overhead Packetisation (ROP)* In this scheme, the data belonging to the same layer (including FEC sections) in each SC are aggregated to create one super section per layer. For SVC this would combine layer and FEC data, while for description-based models individual description sections and associated FEC sections would be merged. These super sections are then packetised using the *SD* mechanism, where each packet contains a portion of the super section from each layer in the streaming class. ROP tends to create packets with large data byte-size content. However, the loss of any packet typically increases the

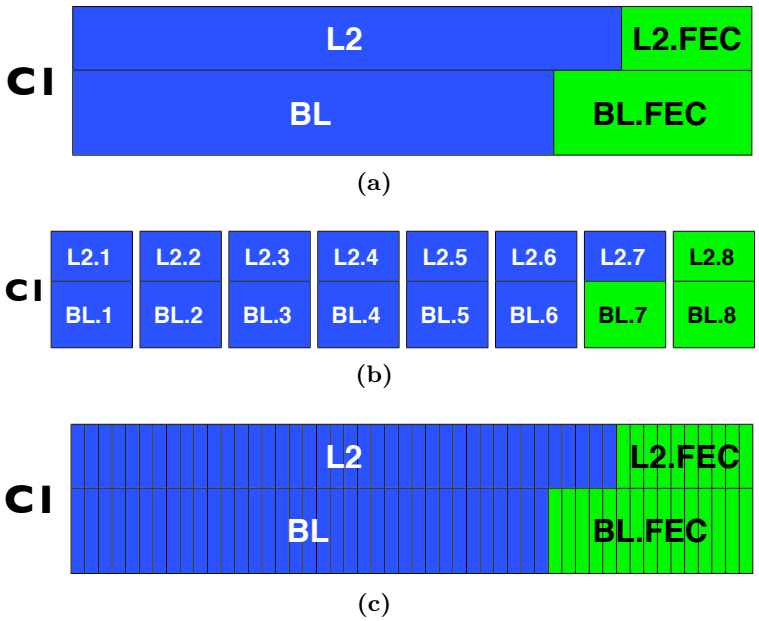


Figure 5.5: Examples of two layers allocated to class one (C_1) for (a) SVC plus FEC and (b) description-based models plus FEC. (b) illustrates the section structure of MDC utilised by IRP, while (c) illustrates an example of the IRP packetisation of the SVC class in (a).

probability of large amounts of multiple layer data being lost, thus reducing the availability of layer data by which stream quality is maximised. Example: C_1 shown in Figure 5.5a illustrates two SVC layers and associated FEC allocated to a single class. Let us assume that the total transmission cost is 7,200 bytes. With ROP, we treat each layer per class as one super section and packetise using the minimum number of packets, *e.g.* 5 packets at 1,440 bytes. Thus the loss of a single packet equates to 1,440 bytes of lost data from layers BL and L2. Should we have used a description-based model for this examples, as per Figure 5.5b, ROP would have combined the individual sections illustrated in Figure 5.5b, so as to represent Figure 5.5a and packetisation would be identical to the previous example.

2. *Improved Resiliency Packetisation (IRP)* For description-based streaming we are able to use the section sizes of the underlying descriptions to define the governing threshold byte-size allocation of our packetisation. As per Figure 5.5b, SD is applied to the individual sections of both layers. Noting that the number of packets depends on individual section sizes in comparison to the super section sizes in ROP, more packets are typically generated on using this scheme.

   For SVC plus error resiliency, we have no easily selectable segmentation markers. For this we use the packetisation threshold as previously defined, by which we can specify our SD packetisation requirements. Figure 5.5c illustrates the packetisation of Figure 5.5a based on a threshold value which determines the maximum

combined byte-size of both layers allocated to a single packet.

As can be seen by using IRP, the loss of one packet would affect a smaller portion of the class data. Note that both ROP and IRP would benefit from SD but IRP has a better granularity. The main drawback of IRP is the additional overhead associated with every packet. Example: again let us assume that the total transmission cost of C_1 shown in Figure 5.5b is 7,200 bytes. With IRP, and for description-based models, we maintain the descriptions structure, *e.g.* 8 descriptions, and packetise based on the byte size of each description, *e.g.* 900 bytes. Thus the loss of a single packet equates to 900 bytes of lost data from layers BL and L2, thus reducing the effects of packet loss on achievable quality by increasing the number of packets transmitted. An example of IRP packetisation of SVC in Figure 5.5a is shown in Figure 5.5c. Note the data between the black vertical lines denote the packetisation segments of each layer.

Using the IRP packetisation example, we note that typically the byte-cost allocated per packet is less than the maximum byte-size of the packet data, circa 1,440 bytes. Thus, one additional option to increase viewable quality is to allocate levels of error resiliency to the layers, such that the total layer cost is less than or equal to the maximum byte-size of the packet data. Therefore we do not increase the number of packets transmitted but only the data content of the packets.

As an example, if we assume the transmission of eight packets and an initial error resiliency allocation rate of 14% for the highest layer. Assuming a 2% increase in FEC for each lower layer, this provides an error resiliency allocation rate of 14% to 28%. If we then take the eight layers and their respective error resiliency rates and packetise using SD. Now each packet contains a segment of every layer, or approximately 12.5% of each layer plus a percentage of the initial layer error resiliency allocation. As the highest layer contains 14% error resiliency and each subsequent layer contains a higher level of error resiliency, both B-C and W-C with a network loss rate of 10% mandates layer 8, or maximum layer, decodability.

This is a simple example, but the underlying concept holds true for layers with differing byte-sizes, for larger GOP, for increased numbers of packets and for varying error resiliency allocations. Once the threshold for packet sizes/number, as well as error resiliency allocation rates, is defined, we can then begin to offset optimality against overall transmission cost to define a balance between viewable quality and the effects of network loss. As illustrated by the error resiliency allocation rates, we can always implement IER to further increase the resiliency of select layers.

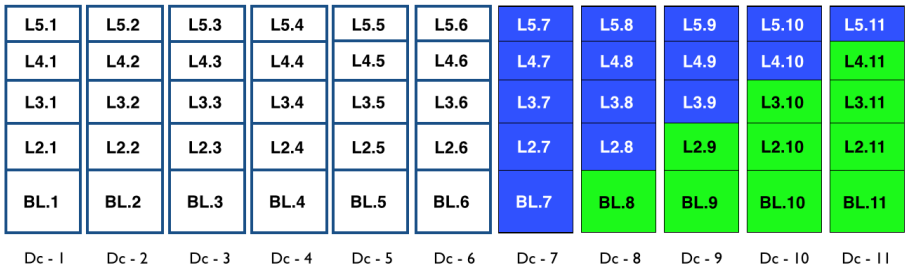


Figure 5.6: ALD with five-layer and an STF of 6

5.1.5.1 Streaming Classes - Class Packetisation Evaluation Results

In this Section, we illustrate a simple ALD example where two classes are packetised using ROP and IRP and transmitted over a link with a 10% loss rate. We begin by allocating the layers to the classes. In this example, we will use the crew media clip encoded as a two-resolution, five-layer stream. We allocate the layers to the classes based on resolution, such that 2 layers (BL and L2) are allocated to C1 and 3 layers (L3, L4 and L5) allocated to C2. A class composition of Independent Class Composition (ICC) is used. Table 5.3 illustrates the JSVM output for the first frame.

Table 5.3: Encoder output for a single frame for the crew video

Length	LId	TId	QId
18	0	0	0
1424	0	0	0
1577	0	0	1
2186	1	0	0
2461	1	0	1
2388	1	0	2

Table 5.4: Transmission cost for SVC, MDC, ALD and ALD-SC

Layer	SVC	MDC	ALD	Class	ALD-SC
L5	10,054	20,270	12,232	Class 2	12,232
L4	7,666	16,216	11,120		
L3	5,205	12,162	10,008		
L2	3,019	8,108	8,896	Class 1	4,032
BL	1,442	4,054	7,784		

Table 5.4 shows the per layer transmission cost for SVC, MDC, ALD with an STF of 6 (shown in Figure 5.6) and ALD-SC (ALD using a Streaming Class model) with ICI class composition. In our example we incorporate three FEC sections of the base layer and two FEC sections of layer two to class 1, while also allocating one FEC section of both the base layer and layer two to class two, as illustrated in Figure 5.7.

The reasoning for the FEC section allocation in the lower class is that the $LR_{C_i}^{\max}$ as defined by Equation 5.3 in Section 5.1.3 for a 10% packet loss rate is a 17% FEC rate for the highest layer in the class. Because of an STF of 6, eight sections of layer

Figure 5.7: Two streaming classes created from ALD with five-layer and an STF of 6. C1 denotes class one and C2 denotes class two

two are required to decode layer two, thus each layer two section equates to 12.5% of layer two. The allocation of a single FEC section for layer two would only equate to 12.5%, which is less than the $LR_{C_1}^{\max}$ of 17% mandated, such that 2 sections totalling 25% are required to eleviate $LR_{C_1}^{\max}$. We view this as an FEC mapping ratio of (3,1) which denotes that the lowest layer in class 1 contains 3 FEC sections, while the lowest layer in class 1 contains 1 FEC sections. This can be generalised to (<number of FEC section in the lowest layer of class 1>, <number of FEC section in the lowest layer of class 2>, ..., <number of FEC section in the lowest layer of class N>). In a streaming class that utilises all FEC sections from the original streaming model, adding all the values in the FEC mapping will equal $N-1$, where N denotes the total number of layers in the original encoding.

Table 5.5 presents the percentage of FEC per layer allocated to each class. As we have not varied the transmission cost of ALD-SC with respect to ALD, the cumulative FEC value per layer, *e.g.* by adding the FEC percentage over both classes for a given layer, are consistent with ALD.

Table 5.5: Per layer FEC percentage per class for ALD-SC

Layer	C1	C2
L5	0%	0%
L4	0%	10%
L3	0%	22.24%
L2	25%	12.50%
BL	42.87%	14.29%

As illustrated in Table 5.4 and Figure 5.7, in our simple example we do not increase the transmission cost of ALD and as such do not increase the error resiliency of C2 to the same level of $LR_{C_i}^{\max}$.

Table 5.6 illustrates the ALD-SC packet sizes and number of packets per packetisation option, ROP and IRP, based on the single frame example in Table 5.4. It can be seen that by creating one super section using ROP that the number of packets required per class is noticeably lower than IRP, but understandably the degradation in viewable quality is greater during moments of packet loss. Figure 5.8 illustrates the percentage of viewable quality for ALD-SC over the duration of the clip for both

packetisation schemes, with a loss rate of 10%. Note how a simple reduction in packet payload creates a noticeable increase in the viewable quality. It can also be seen that due to the incremental levels of FEC per layer within a class, different loss rates per GOP mandate the decodability of different layers per class.

Table 5.6: ALD-SC Packet sizes and number of packets per packetisation option

	ROP		IRP	
Class	Packet size	# of	Packet size	# of
Class 2	1299	8	800	13
Class 1	1344	3	404	8

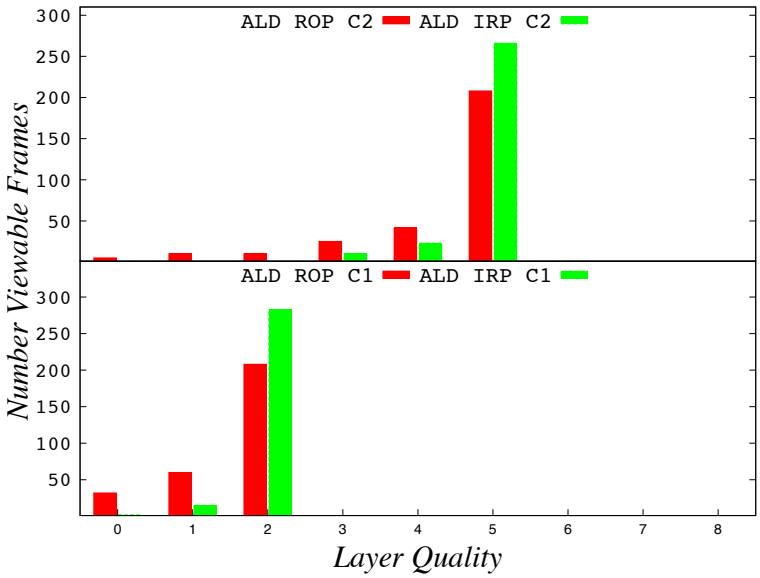


Figure 5.8: Viewable quality of ALD-SC for both packetisation schemes, ROP and IRP, with a loss rate of 10%

We extend this example in Appendix C and we provide examples of defining the packet byte-size allocation based on fixed values of 1,440 bytes, 1,000 bytes and 500 bytes for each class, rather than the defined section size of the underlying description model. Defining the packet byte-size allocation based on fixed byte sizes is consistent with the packet byte-size allocation required for SVC. These values define that all packets per GOP, over all classes, are the same byte size, but not of equal importance (lower class packets are still of higher priority). We also defined a model where a 1,440 byte threshold is allocated to the highest class, 1,000 byte threshold to the middle class and 500 bytes to the lowest class, thus illustrating bytes sizes relative to the underlying byte cost of the respective classes. While a final option determines the number of packets for the highest layer using a 1,440 byte allocation, and mandated that all lower classes utilise the same number of packets, as a result forcing an optimal packet threshold for the lower classes based on the number of packets for the highest class. The results provided in Appendix C for SVC-SC (SVC using a Streaming Class model) illustrate that as we increase the GOP value, different levels of allocated FEC can

provide continuos levels of maximum stream quality for all packet byte-size allocation schemes illustrated above. Thus illustrating both adaptation in FEC allocation and packet byte-size allocation by which viewable quality can be preserved during moment of packet loss. For the examples in this section and Appendix C, a class composition of Independent Class Composition (ICC) is used.

5.2 Evaluation Framework

The evaluation steps in this Chapter are based on the Evaluation Methodology as outlined in Section 3.4. The results provided in Appendix C illustrate an example where by the Streaming Class design principles are utilised to adapt an SVC stream using SVC-SC (SVC using a Streaming Class model) based on ICC. For the remainder of this chapter we present examples of description-based streaming classes, namely MDC and ALD, based on ICI (note we do not increase the transmission cost of MDC or ALD, thus no additional FEC is added to layer 8). It is important to note that the usage of the *Streaming Class Structure* principle for layered or description-based models is interchangeable. The usage of ICC and ICI in our examples is for illustration purposes only, and their allocation to the underlying models demonstrates their usage rather than mandating that ICC is only for SVC-SC and ICI is only for description-based SC.

Prior to presenting our evaluation results, we provide a brief overview of our evaluation framework. Our evaluation is based on the widely-known 10 second *crew* video. The video is encoded using JSVM [131] to eight layers with spatial and quality scalability, using medium grain scalability (*mgs*), quantizer parameter (QP) values for BL to L8 of 34, 28, 33, 30, 28, 35, 32, 30 respectively, and a GOP value of one. We consider three resolutions (QCIF, CIF and 4CIF) with respective 2, 3, and 3 quality levels, *e.g.* two fidelity levels in the lowest resolution and three fidelity levels in each of the higher resolutions, which are allocated to three Streaming Classes based on underlying resolution. QCIF is mapped to C1 (maximum 38.2dB), CIF is mapped to C2 (maximum 39.3dB) and 4CIF is mapped to C3 (maximum 38.7dB).

The transmission of the encoded video is simulated in Network Simulator 2 (*ns-2*) [138] using myEvalSVC [139], an open source tool for evaluating JSVM video traces for SVC. Modifications are made to myEvalSVC scripts to simulate MDC, ALD and SCs. In SVC, each layer per frame is packetised individually, in MDC, each description per frame is packetised separately, while in ALD, ALD-SC and MDC-SC, each packet contains a segment of each layer per description (using SD). In ALD, this would lead to a segment from every layer per packet, while in the SC models, each class would be packetised separately, but within each class, a packet would equate to a segment of each layer.

The simulated network topology is shown in Figure 5.9 in which we vary the

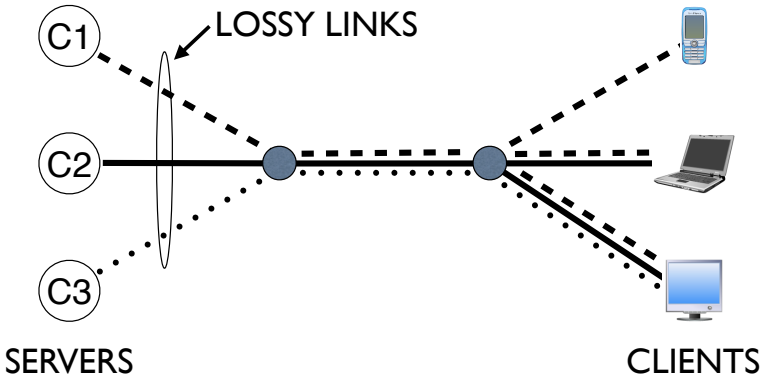


Figure 5.9: Simulated network topology

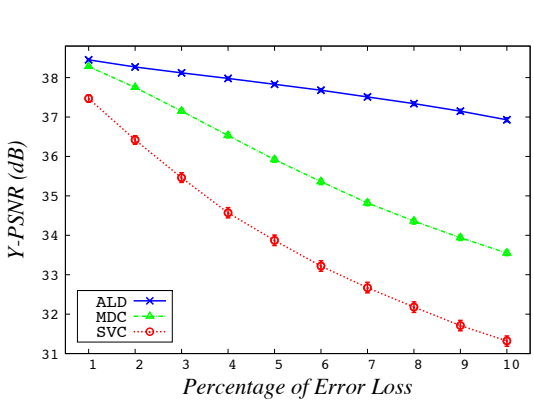
average packet error rate, μ, from 1% to 10% to test the streaming performance of different schemes over lossy links. We use an ns-2 Errormodel to define a total packet error rate with a uniform distribution, such that the level of loss per frame varies from less than or equal to, to greater than μ, but with total average stream loss equal to μ.

For each of the simulated schemes, sixteen iterations are run to create the ns-2 output traces, which are analysed to determine the average maximum stream quality per-frame at the client. Each trace is then saved as an achievable quality (AQ) trace file for each streaming scheme. The AQ trace files are utilised to 1) to provide a means of illustrating the transition in frame quality over time and 2) to create the received YUV files, based on the maximum stream quality per frame, from the original YUV files. PSNR is then calculated as previously described.

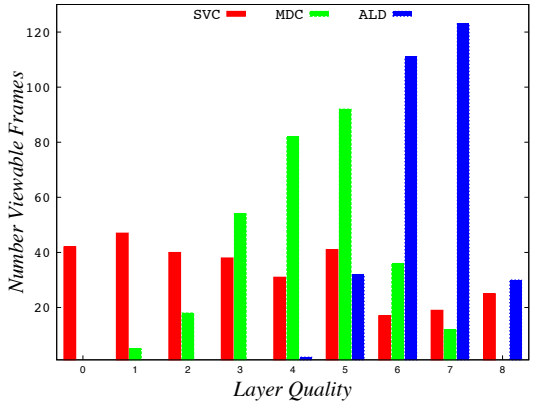
5.3 Simulation Results For SVC, MDC and ALD Without Using The SC Framework

The following results are provided as an example of the SVC, MDC and ALD evaluation determined so far and provide a base case comparison to the SC result shown later in this section. Figure 5.10a plots the Y-PSNR (Y-PSNR is the measured PSNR for the Y-component of YUV) values versus the percentage of datagram loss over the communication link for SVC, MDC and ALD when the user is streaming the highest video quality (4CIF). In this section SVC is shown for comparison purposes only. The results indicates that ALD shows the best performance followed by MDC and then SVC. Typically, MDC is better than SVC due to the included FEC. The further improvement achieved by ALD are due to the increase in the number of descriptions, reduction in the byte-allocation per description section and SD. SD disperses the loss impact over several sections instead of a single datagram loss affecting only one layer (SVC) or one description (MDC).

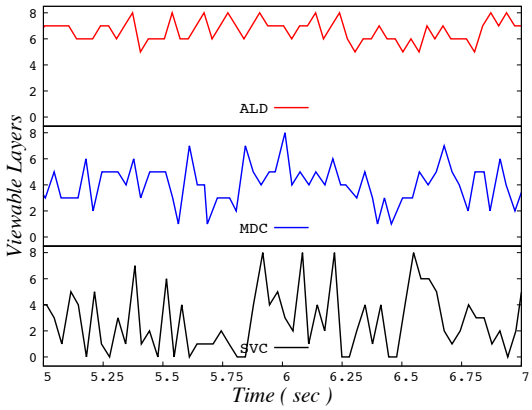
Figure 5.10b confirms these results by showing the number of frames viewed at



(a) Versus packet loss ratio



(b) Viewable video quality at 10% loss.



(c) 2 second sample of viewable quality transitions

Figure 5.10: Performance of scalable video encoding over lossy links

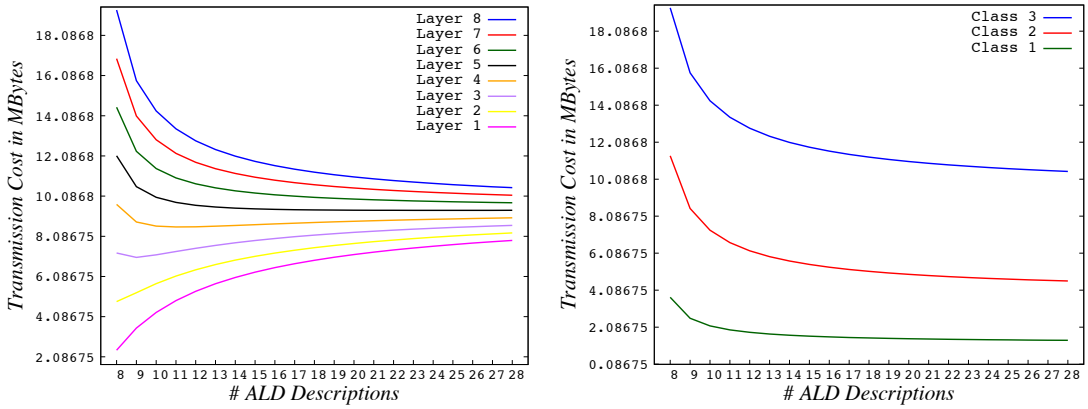
every quality level for the crew video at a datagram loss rate of 10% when SVC, MDC and ALD are used. This figure demonstrates the severe impact of packet loss on SVC performance, where approximately 40 frames were completely lost due to the loss of the base layer. Additionally, SVC shows frequent quality-level shifts where each layer is viewed between 17 and 47 times. On the contrary, users using MDC video enjoyed a better streaming experience where more than 80% of the frames are viewed between qualities 3 and 6. Using ALD further improves the streaming experience where more than 80% of the video is shown at the highest resolution (4CIF).

Figure 5.10c takes the frames viewed at every quality level, for each of the models, and plots a two second sample of the frequency of layer switching. This result again illustrates the high variation in quality for SVC, minor increase in quality for MDC and consistency of quality in the higher layers for ADL.

More importantly, these improvements are attained at a lower transmission byte-cost as shown in Table 5.7, which also shows the relative transmission cost compared to SVC. These savings in byte cost are made possible thanks to the STF component of

Table 5.7: Transmission byte-cost as for the highest quality

Scheme	SVC	MDC	ALD
Transmission (bytes)	9,483,746	19,334,064	12,075,882
Datagrams	7,687	14,248	10,108
Value compared to SVC	100 %	≈ 204 %	≈ 128%



(a) **(b)**

Figure 5.11: Transmission cost of the crew media clip for (a) each layer, and (b) each streaming class, as the STF value and associated number of ALD descriptions increase

ALD. In the shown results, the value of STF is 6 according to the developed optimisation framework shown in Section 4.1.1. Hence, a total of fourteen descriptions are required to stream the video in ALD at the highest quality. The main drawback of section thinning in ALD is the higher transmission cost when low quality video is requested. As we shall show, this problem is eliminated by using SCs.

5.4 SC Performance Evaluation

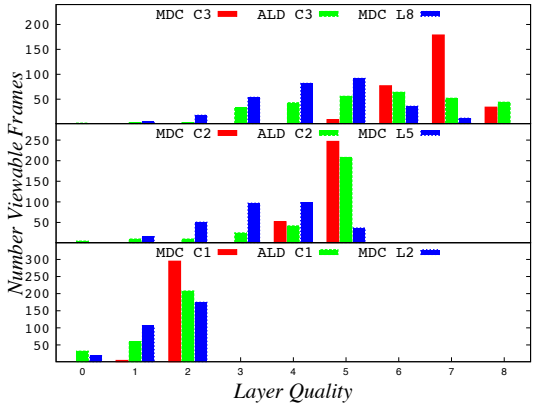
First, we present the transmission cost in bytes for ALD and the corresponding savings per class when ALD-SC (ALD using a Streaming Class model) is used. Figure 5.11a plots the transmission byte-cost for the crew video versus the ALD-STF value for distinct video qualities. The structure of Figure 5.11a is identical to Figure 4.4 from Section 4.4.2.2, only the underlying bitrates are different due to encoding and QP value decisions. Note that the eight ALD description encoding, $STF = 0$, degenerates to MDC, thus Figure 5.11a represents both ALD and MDC. Figure 5.11a illustrates the aforementioned limitation of ALD showing the increase of the transmission cost of streaming low quality as STF increase. On the contrary, Figure 5.11a also shows that as STF increases the transmission cost of streaming high quality video decreases. Figure 5.11b plots the transmission cost of SCs versus the number of ALD descriptions. It can be seen that by using ALD-SC, there is a significant drop in transmission cost and that the transmission cost of ALD-SC always decreases as STF increases.

Table 5.8: Transmission byte cost for Different Encoding Schemes

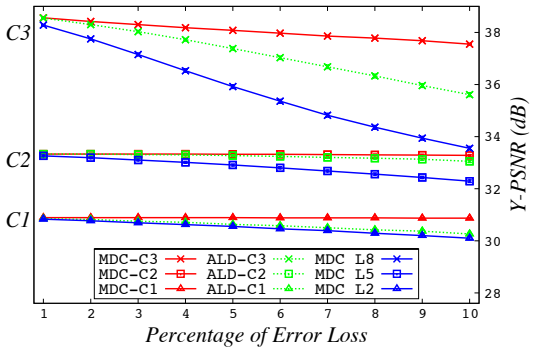
Layer	SVC	MDC	ALD	MDC-SC (ROP)	Dg	ALD-SC (ROP)	Dg	ALD-SC (IRP)	Dg
8	9,483,746 (8)	19,334,064 (8)	12,075,882 (14)	19,347,198 (3)	13,892	12,077,528 (3)	8,825	12,077,528 (3)	13,558
7	7,419,782 (7)	16,917,306 (7)	11,213,319 (13)						
6	5,640,092 (6)	14,500,548 (6)	10,350,756 (12)						
5	3,931,226 (5)	12,083,790 (5)	9,488,193 (11)	9,726,545 (2)	7,072	5,663,237 (2)	4,215	5,663,237 (2)	6,348
4	2,993,946 (4)	9,667,032 (4)	8,625,630 (10)						
3	2,044,662 (3)	7,250,274 (3)	7,763,067 (9)						
2	1,232,574 (2)	4,833,516 (2)	6,900,504 (8)	2,776,152 (1)	2,090	1,651,653 (1)	1,281	1,651,653 (1)	2,700
BL	617,526 (1)	2,416,758 (1)	6,037,941 (7)						

Table 5.8 illustrates the per quality cumulative transmission cost for SVC, MDC, ALD ($STF = 6$) and the respective SC schemes. The number in the brackets denotes the number of transmission units (layers for SVC, descriptions for MDC and ALD, and classes for the SCs) required to decode the target quality. It can be seen that due to the packetisation of MDC-SC, there is a reduction in the transmission cost of layer 2 and layer 5, with respect to MDC. It is important to note that these figures are content and encoding specific. Minor modifications to the quantisation parameters (QP) in JSVM and subsequent decoding quality can reduce the increase costs of streaming classes. For ALD-SC the transmission cost values show a marked reduction in the transmission overhead for the quality layers within the lower classes, with respect to all other models. ALD-SC class allocation for layers six and seven mandate a minor increase in transmission cost of ALD-SC relative to ALD, due to the allocation of lower layer FEC data within C3. However, it is important to note that such minor increases in transmission cost, for both MDC-SC and ALD-SC, is accompanied by an improvement in achievable quality due to layer grouping. Note: as each class is composed of layer data and incremental levels of FEC, only a subset of packets is required to decode the lower layers per class, thus by implementing a simple feedback mechanism for deterring the number of packets required per layer, per class. Lower layer streaming per class could be performed thus further reducing decoding complexity at the device. Overall transmission cost would not be decreased, as all packets per class would still be transmitted, whereby only a subset of packets would be received. Table 5.8 also shows the number of transmitted datagrams for each SC schemes assuming a maximum transmission unit of 1500 bytes. It is important to note that transmitting more datagrams implies an increased transmission overhead for lower layers (not shown).

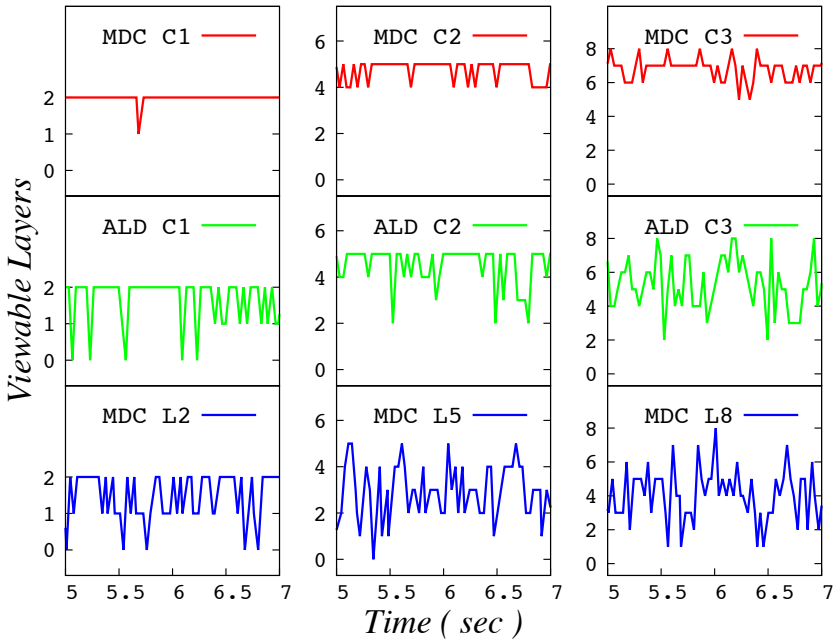
In the following, we compare the streaming quality performance of MDC, MDC-SC, and ALD-SC by showing PSNR and viewable frames different quality levels when ROP and IRP are used. As illustrated in Section 5.1.5.1, with larger STF values ALD-SC requires a greater number of FEC sections in the lower classes to accommodate greater levels of packet loss. Thus in this comparison, an FEC mapping ratio (see Section 5.1.5.1) of 2, 3, 2 is used for MDC while ALD-SC has an FEC mapping of 3, 3, 1. The mapping ratio of ALD-SC is larger in C1 due to the smaller byte-allocation per lower layer section, the increased impact of datagram loss on achievable quality and the $LR_{C_i}^{\max}$ as defined by Equation 5.3 in Section 5.1.3, thus mandating a higher



(a) Video quality for different schemes at 10% loss



(b) Average Y-PSNR value for different schemes with 95% confidence interval results



(c) 2 second sample of viewable quality transitions for MDC (layers L2, L5 and L8) and for each of the classes of ALD-SC and MDC-SC (maximum quality per class equating to layers 2, 5 and 8)

Figure 5.12: Performance evaluation of considered schemes using ROP for SCs.

level of FEC for C1. MDC has larger levels of inherent FEC per section with which to combat packet loss thus a lower level of allocated FEC is required for C1.

Figure 5.12a plots the number of viewable frames per quality level for each of the two SC classes (MDC-SC and ALD-SC) at 10% datagram loss rate using ROP for the streaming classes, as well as the respective highest layer in MDC: Layer 2 (L2) in C1, Layer 5 (L5) in C2, and Layer 8 (L8) in C3). SVC has been removed from all subsequent plots as we are now compare description based models. SVC-SC is presented in Appendix C. The three subfigures may be considered a representation for users with different bandwidth availability. Clearly, the figure shows that MDC-SC has

the best performance for all user types followed by ALD-SC then MDC. Noting that both MDC and MDC-SC stream identical data (same number of sections) when the highest quality is requested (top most figure), the performance gain for MDC-SC over MDC is interpreted by the positive impact of SD over the defined super sections. For the same highest quality, the success of ALD in decoding more frames at the highest quality (layer 8) is due to SD packetisation. For limited bandwidth users (bottom subfigure), the included FEC sections are considered the key reason for MDC-SC and ALD-SC in achieving a higher video quality in comparison to MDC. A similar performance is noticed in the middle figure (intermediate bandwidth availability).

Figure 5.12b highlights the average Y-PSNR values versus the packet loss ratio for the same considered schemes. Note that the SC grouping for ALD-SC and MDC-SC has provided near consistent quality for each iteration of the simulation, thus mandating a near un-viewable range of confidence interval error bars. Clearly, Figure 5.12b is consistent with the results in Figure 5.12a since MDC-SC achieves the highest PSNR followed by ALD-SC then MDC. The figure shows a 2dB difference in the PSNR between MDC-SC and ALD-SC when the highest quality is streamed at 10% loss. For the same quality and loss ratio, a larger performance gap of 4dB exists between MDC-SC and MDC, due to the SD component in SCs. This PSNR gap is much smaller for the intermediate and limited bandwidth cases (middle and bottom figures). It can also be seen that by utilising ALD-SC with ROP, so as to reduce transmission cost for lower classes, the PSNR values of ALD-SC have dropped when compared with ALD from Figure 5.10a.

Figure 5.12c illustrates a two second sample of the frequency of layer switching for each of models. Note the high impact of packet loss on the variation of quality for each model, that occurs with ROP packetisation.

In practice PSNR for a selected resolution is analysed against the same resolution. But in our figures, this would have created increased PSNR values for all lower classes, where the goal of the PSNR figures is to highlight the consistency of the quality, per class, as datagram loss increases. Thus in our evaluation, we calculated the PSNR values for each model per class, irrespective of original resolution, by comparing the modified YUV file against the original YUV file with the highest quality and resolution, *e.g.* Layer 8. This creates reduced PSNR values for the lower classes, but provides PSNR values that are consistent with the achievable quality of each layer.

Table 5.9 outlines a summary of the mean layer value viewed over the duration of the clip, *i.e.* over all 300 frames, and the maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer value viewed, thus illustrating the the variation in quality that occurred during decoding. I will comment on each one of the streaming models separately. For MDC-SC, we note that each additional class received can increase mean layer quality to each respective resolution. For ALD-SC, we note that the non-decodable frames in C1

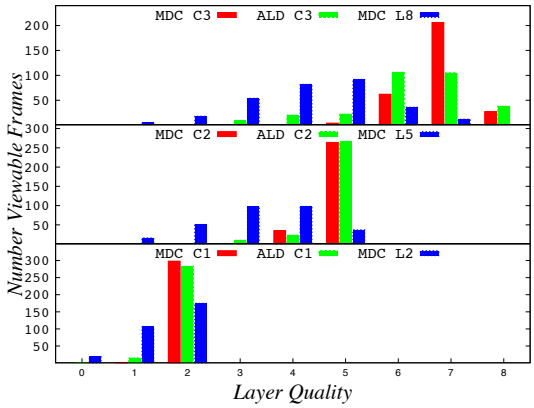
Table 5.9: Example of the mean layer value and the variation that occurs between maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer viewable quality over the duration of the clip

	Mean	$\min_{qual}$	$\max_{qual}$
MDC C1	1	1	2
MDC C2	4	3	5
MDC C3	6	4	8
ALD C1	1	0	2
ALD C2	4	0	5
ALD C3	5	0	8
MDC L2	1	0	2
MDC L5	3	1	5
MDC L8	4	0	8

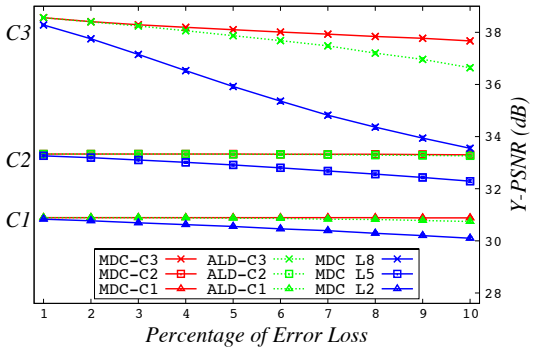
cascade over all higher classes, thus reducing overall viewable quality for these specific frames. While for MDC, the quality is dependent on the number of descriptions received without data loss, thus MDC L5 has a minimum decodable layer quality of layer 1, while MDC L8 is unable to decode some frames. The selective packetisation utilised in MDC-SD illustrates the increase in achievable quality with no increase in transmission cost with respect to MDC.

On using IRP, the performance gap between ALD-SC and MDC-SC shrinks to 1dB for the highest quality at 10% loss. Additionally, the PSNR performance gap becomes insignificant for both low and intermediate quality levels. Figure 5.13a, Figure 5.13b and Figure 5.13c respectively show the same performance metrics as Figure 5.12a, Figure 5.12b and Figure 5.12c but for IRP packetisation. This performance gain is attained due to distributing the error impact over a larger number of smaller packets. However, these packets also introduce an additional transmission cost of extra packet headers belonging to lower layers. For the crew video, this additional overhead can be estimated as a 3% increase in the total transmission cost in IRP in comparison to ROP (assuming a 60-byte header in a 1,500 byte packet). In conclusion, the additional overhead of MDC-SC is considered useful only for users having abundant bandwidth and lossy links. In case of limited or low bandwidth, ALD-SC performs closely to MDC-SC but with a much lower overhead.

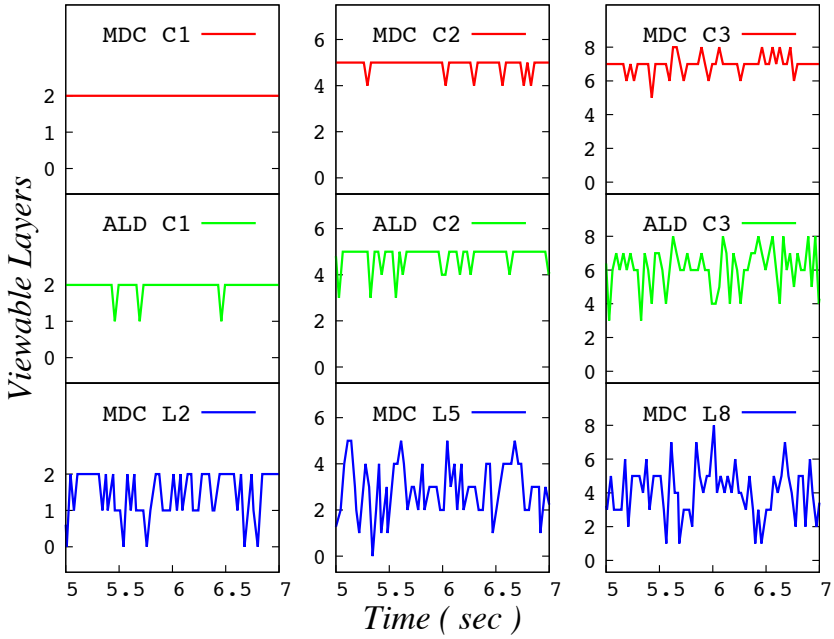
Table 5.10 outlines a summary of the mean layer value viewed over the duration of the clip, *i.e.* over all 300 frames, and the maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer value viewed, thus illustrating the the variation in quality that occurred during decoding. I will comment on each one of the streaming models separately. For MDC-SC, we note similar results to Table 5.9 but an increase in the minimum quality layer from layer 4 to layer 5 for C3. For ALD-SC, we note that due to IRP as well as the underlying ICI class structure, there is an increase in the minimum quality layer from a non-decodable frame to layer 3 for C2 and C3. While the results for MDC remain the same as for Table 5.9, as MDC does not use either ICI or IRP.



(a) Video quality for different schemes at 10% loss



(b) Average Y-PSNR value for different schemes with 95% confidence interval results



(c) 2 second sample of viewable quality transitions

Figure 5.13: Performance evaluation of considered schemes using IRP for SCs.

Table 5.10: Example of the mean layer value and the variation that occurs between maximum, $\max_{qual}$, and minimum, $\min_{qual}$, layer viewable quality over the duration of the clip

	Mean	$\min_{qual}$	$\max_{qual}$
MDC C1	1	1	2
MDC C2	4	3	5
MDC C3	6	5	8
ALD C1	1	0	2
ALD C2	4	3	5
ALD C3	6	3	8
MDC L2	1	0	2
MDC L5	3	1	5
MDC L8	4	0	8

5.5 Discussion

As we have seen from the evaluation results from Section 5.4, Section 5.1.5.1 and Appendix C, the design principles of SC increase the viewable quality of the underlying streaming models. Section 5.4 presents our evaluated results for description-based SC streaming (MDC-SC and ALD-SC) in comparison to defined layer values in MDC. Section 5.1.5.1 illustrates a simple example of the variation in quality of ALD dependent on the packetisation options ROP and IRP, while Appendix C extended the granularity of the packetisation by showing the variation in quality over varying GOP values for an SVC-SC model. Each of our design principles provide a means of adapting elements of the stream flow to suit network and user requirements, as well as illustrating the interaction that can occur between the elements. For layered schemes such as SVC this mandates a minor increase in transmission cost, relative to the levels of error resiliency allocated, while for description-based schemes such as MDC and ALD the transmission cost can be reduced and optimised to suit achievable quality. Our evaluated examples provide a sample of the possible configuration options that can be selected by using our streaming class framework and even these samples illustrate the gains that can be made in viewable quality.

What can be inferred from our results:

- That increases in transmission cost as mandated by MDC do not always provide for stability in achievable quality.
- That strategic choices made during the various elements of the stream flow can increase viewable quality.
- That the inherent complexity in the number of options offered by SC may overly complicate the decision making process.
- That a larger test sample offered by a complete real-world implementation of SC is required to full appreciate the operations and interdependency between the elements of the network flow and our design principles.
- That no one group of SC options is sufficient to accommodate for all streaming models and transmission mediums, and the issues that can occur during streaming over the network.

5.6 Conclusion

In this chapter, Streaming Classes (*SC*) is proposed as a new approach for the transmission of layered and description-based adaptive streaming. We present five design principles by which high levels of viewable quality can be maintained over the duration

of the entire media clip. We propose that data belonging to different quality layers are grouped together to form streaming classes, based on user, network or scalable dimensionality requirements, by which transmission byte-cost and achievable quality can be managed and improved respectively. Evaluation results for streaming class extension of existing video models (SVC-SC, MDC-SC and ALD-SC) shows significant performance improvements such as consistent high levels of quality for users with varying resource availability, as well as a reduction in transmission cost for devices requiring lower layer description-based decoding.

Chapter 6

Subjective Testing

6.1 Design Process

In this chapter, we present our scalable video subjective testing [16]. The goal of our scalable video subjective testing is to confirm the performance of our techniques and models with subjective evaluation. The subjective testing was undertaken prior to the development of our Streaming Class (SC) streaming model, as presented in Chapter 5, thus SC is not included in this evaluation. We begin this chapter by presenting the decision making process based on the choice of options available during video clip selection, encoding, simulation and trace analysis.

6.1.1 Design Options

For each of the Methodology steps, a number of design options are offered. It is important to note that these options are not exhaustive, but only represent a sample of the options available. The options shown provide an overview of the possible options available, so as to permit the reader to fully understand the range of options available. At the end of each subsection, we shall present the specific option selected.

6.1.1.1 Number of Layers to Encode

In this section, we determine the number of SVC layers encoded into each media clip. As previously stated in Section 3.4.2, in our assessment of JSVM a maximum of eight layers can be encoded per SVC media clip using Medium-Grained Scalability (MGS) and a GOP value of one. A larger number of layers can be created when the temporal dimension is considered, *e.g.* when larger GOP values are used. The allocation of these layers is limited to a maximum of three resolutions. Options include:

1. An eight layer SVC clip provides a sufficient number of layers per resolution to provide variation per resolution as loss affects achievable quality.
2. A six layer SVC clip offers an equal number of quality layers per resolution, *e.g.* two layers for each resolution.
3. A three layer SVC clip will offer a relatively high level of inter layer transition as well as a high level of frame non-decoding, as single datagram loss will affect entire layers/descriptions for SVC and MDC. While also creating large transmission cost variation as devices move between the different resolutions.

An eight layer SVC clip is selected as this provides an encoding more applicable to a real-world scenario with varying device requirements, while also providing a sufficient number of layers for variation in viewable quality.

6.1.1.2 Maximum Datagram Loss Rate

Which percentage of datagram loss, μ, per media clip should be used :

1. A high percentage loss rate of 10%: This provides clear variation in the achievable quality for each of the schemes, as highlighted in the PSNR test results from the earlier chapters.
2. A low percentage loss rate of 1% to 2%: Even relatively low datagram loss rates illustrate the change in achievable PSNR values and subsequent variation in viewable quality for each of the transmission schemes.
3. A mixture of percentage loss rates of 3%, 6% and 10%: Thus illustrating the variation in quality as loss increases.

As each additional loss rate will double the number of clips to be viewed and graded, a single loss rate of 10% is selected. A 10% loss rate over the duration of the media clip provides both sufficient single and bursty loss within the clip so as to fully illustrate the variation in quality that can occur during streaming.

6.1.1.3 Group of Pictures (*GOP*) Rate

Number of frames per GOP:

1. *GOP value of one* : Illustrates the effects of packet loss on the viewable quality of a single frame only and does not demonstrate the inter-dependency of multiple frames. Each frame in the stream is encoded as an I-frame.
2. *GOP value of eight* : Highlights how datagram loss can affect both frame quality and the inter-dependency of eight frames per GOP. A frame hierarchy of IBBBPBBB per GOP is proposed.

3. *GOP value of sixteen* : Shows how datagram loss can affect both frame quality and the inter-dependency of sixteen frames per GOP. A frame hierarchy of IBBBBBBPBBBBBB per GOP is proposed.

Our goal is to subjectively evaluate the effects of network loss on the achievable quality of the individual streaming models and as such a GOP value of one is selected. A higher number of frames per GOP would introduce the inter-dependency of multiple frames and for some models would cause a cascading degradation in viewable quality across a subset of frames per GOP.

6.1.1.4 Evaluated Models

While we consider the streaming models which can be evaluated, we will also outline how stream quality is determined based on the transmission unit of each streaming model and the underlying packet loss in network:

1. *SVC* : A streaming model where the loss of one or more datagrams from a layer, per GOP, makes that specific layer undecodable. Thus reducing stream quality to the maximum cumulative, 1 .. N, layer with no datagram loss.
2. *MDC* : A streaming model where the loss of one or more datagrams from a description, per GOP, makes that specific description undecodable. Thus reducing stream quality to the number of descriptions received with no datagram loss.
3. *MDC-SDP* : A streaming model where the loss of one or more datagrams from a description section, per GOP, makes that specific description section undecodable. Thus reducing stream quality to the number of description sections received with no datagram loss.
4. *MDC-SD* : A streaming model where the achievable quality is determined by the cumulative datagram loss rate of all descriptions per GOP.
5. *SDC-SDP* : A streaming model where the loss of one or more datagrams from a description section, per GOP, makes that specific description section undecodable. Thus reducing stream quality to the number of description sections received with no datagram loss.
6. *SDC-SDP-NC* : A streaming model where the loss of one or more datagrams from a description section, per GOP, makes that specific description section undecodable. Thus reducing stream quality to the number of description sections received with no datagram loss. Coupled with the benefits provided by Network Coding (*NC*).
7. *SDC-SDP-SD* : A streaming model where SD is applied to the complete descriptions, D_c, and redundancy descriptions, D_r, only. In these descriptions the

achievable quality is determined by the cumulative datagram loss rate of all descriptions per GOP. SDP is only applied to the scalable description, D_s, as any loss using SD would make the entire D_s invalid. SDP allows us to re-use sections which were not effected by datagram loss and as such can be viewed as where the loss of one or more datagrams from a scalable description section, per GOP, makes that specific description section undecodable.

The cumulative quality of the decoded stream is based on the loss rates of both the SDP and SD elements of the encoding.

8. *ALD* : A streaming model which utilises SD by default, where the achievable quality per GOP is determined by the cumulative datagram loss rate.

9. *ALD-IER*$_{8,1}$: A streaming model where the achievable quality, per GOP, is determined by the cumulative datagram loss rate. Coupled with higher error resilience on the highest layer, *e.g.* layer 8.

10. *ALD-IER*$_{8,1}$ *and ALD-IER*$_{7,1}$: A streaming model where the achievable quality, per GOP, is determined by the cumulative datagram loss rate. Coupled with higher error resilience on the highest two layer, *e.g.* layer 7 and layer 8. From this point forward and for clarity, IER on the highest two layer shall be denoted as IER-2, i.e. one additional section on the highest two layers.

Seven streaming models are chosen: SVC, MDC, MDC-SDP, MDC-SD, SDC-SDP-SD, ALD and ALD-IER-2. These models provided a good sample representative of all the techniques and streaming models researched.

6.1.1.5 Media Clip

This section determines the number of media clips to choose. The media clips proposed for our subjective testing were previously introduced in Section 3.4.1, but are reproduced here for ease of access. Based on the spatial-temporal classification as defined in [159] (temporal - movement and spatial - blockiness, blurriness and brightness), we group the spatial and temporal complexity of each clip into an ordered pair of values, *i.e.* (spatial, temporal) so as to define 4 distinct values - (high,high), (high,low), (low,high), and (low,low). The spatial-temporal classification of each clip is included below.

1. *City* - 10 second 300 frame low resolution low motion media clip of an aerial view of a city skyline with specific focus on one building. Resolutions include: 176x144 (*QCIF*), 352x288 (*CIF*) and 704x576 (*4CIF*). This clip has a spatial-temporal classification of (low,low) - low brightness and low movement.

2. *Crew* - 10 second 300 frame low resolution low motion media clip of a number of astronauts walking down a corridor while waving. Resolutions include: QCIF,

CIF and 4CIF. This clip has a spatial-temporal classification of (low,high) - low brightness and high movement.

3. *Harbour* - 10 second 300 frame low resolution low motion media clip of a number of boats, complete with flying birds. Resolutions include: QCIF, CIF and 4CIF. This clip has a spatial-temporal classification of (high,low) - high brightness and low movement.

4. *Soccer* - 10 second 300 frame low resolution high motion media clip of a number of people playing soccer. Resolutions include: QCIF, CIF and 4CIF. This clip has a spatial-temporal classification of (high,high) - high blurriness (background) and high movement.

5. *Sintel* - 52 second 1248 frame high resolution low/high motion media clip, of a trailer for an animated movie. Resolutions include: 854x480p (*480p*), 1280x720p (*720p*) and 1920x1080p (*1080p*). The resolution of the Sintel clips shall be rescaled to provide consistency with the resolution of the other clip types. This clip has a spatial-temporal classification of (high,high) - high blurriness (background) and high movement. Sintel is the only clip which contains fading between scenes (blacked out frames), white text on a black background (title and credit frames), animation and a clip duration of 52 seconds. This clip provides a more realistic viewing experience and the subjective testing of Sintel illustrates the annoyance (variation in viewable quality, upscaling and blocky pixels) and benefits (consistency of quality and high resolution) that can be found in the evaluated streaming models.

All five clip types are selected. A single frame from each of the media clips evaluated by subjective testing is presented in Figure 6.1.

6.1.1.6 Resolution Allocation

Resolution allocation is based on the number of quality levels for a given resolution. We allocate the quality levels based on the following resolution schema: (QCIF, CIF, 4CIF), where QCIF (low Resolution), CIF (mid Resolution) and 4QCIF (high Resolution) are integer vales denoting the number of different quality layers per resolution. As selected in the layer selection option, each clip shall contain a total of eight layers.

1. (1, 2, 5) : one layer at the low resolution, two layers at the mid resolution and five layers at the high resolution

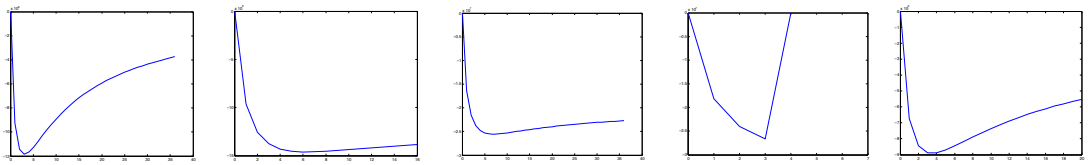
2. (2, 2, 4) : two layer at the low resolution, two layers at the mid resolution and four layers at the high resolution

3. (2, 3, 3) : two layer at the low resolution, three layers at the mid resolution and three layers at the high resolution

Figure 6.1: Single frame from each of the media clips evaluated by subjective testing: crew, city, harbour, soccer and Sintel

4. Realistically, we could continue varying the encodings until all eight layer combinations have been included. In a real world scenario, the final encoding would depend on a number of factors. The number of users per resolution, the different quality requirements, the bandwidth limitation of the transmission network and the original encoding cost, to name but a few. If the encoding can be created in real-time, then the encoding rates will alter dependent on the factors previously outlined. If the encoding cannot be undertaken in real-time, then an approximation of user requirements will be made, and one or more subsets of the encodings will be created and stored.

Based on the findings in Section 3.4.2.1, a (2, 3, 3) encoding is selected as this provides a near balanced allocation of layers to each of the resolution levels.



(a) City stream, STF 3 **(b)** Crew stream, STF 6 **(c)** Harbour stream, STF 7 **(d)** Sintel stream, STF 3 **(e)** Soccer stream, STF 3

Figure 6.2: STF values determined for each of the clip types based on SVC transmission costs and the optimisation framework shown in Section 4.1.1 Note the x-axis is the STF value, while the y-axis is the cumulative transmission cost.

6.1.1.7 Quantisation Parameter Values

In this section we determine the JSVM Quantisation Parameter (QP) values per clip. We have a number of options for defining the QP value per encoding and per layer:

1. Define a static range of QP values for all media clips, thus maintaining QP over all clips. This may provide differing PSNR values for each layer within each clip.
2. Define a range of QP values specific to each clip, so as to maintain defined PSNR values per layer for all media clips. This provides identical quality for each layer in each media clip.
3. Define a range of QP values specific to each clip, so as to maintain equal bit-rate increases per additional layer quality. Thus providing a gradual degradation in quality.
4. Define a range of QP values, so as to maximise PSNR values per layer (but also maximise transmission cost) or limit QP values to an acceptable PSNR value per layer.

A defined static range of QP values is selected thus maintaining the same QP values over all clips. This mandates consistency of QP over all clips and illustrates that the changes in bitrate per clip is dependent on clip content rather than QP values. Table 6.1 shows the selected QP value per layer and the layer allocation per resolution. As stated, the same QP values where used for all clip types.

Table 6.1: QP value per layer for all clip types

Resolution	QCIF		CIF			4CIF		
Layer	BL	2	3	4	5	6	7	8
QP Value	34	28	33	30	28	35	32	30

Table 6.2 defines the maximum achievable PSNR, measured in dB, for the highest layer, *e.g.* layer 8, for each media clip based on the QP encoding parameters selected. Note how the same QP values provide varying PSNR values based on clip type. The

PSNR values for the sintel trailer are very high due primarily to the fading in and out of the various clips, plus the high level of black frames.

Table 6.2: Maximum achievable PSNR value (dB) per clip type for layer 8

Layer	City	Crew	Harbour	Sintel	Soccer
PSNR	36.70	38.75	37.02	49.11	38.03

6.1.1.8 ALD STF values

This section determines the ALD STF value per clip. We have two options:

1. Define a default STF value to use for ALD for all clips.
2. Determine the optimal STF value, STF_o for each ALD clip.

A per clip STF value determine by STF_o is selected, this may mandate different STF values for each clip. Figure 6.2 provides the optimal STF value based on the optimisation framework shown in Section 4.1.1 for each of the media clips. Table 6.3 illustrating the STF for each of the clip types. Note: STF is based on the optimal cumulative transmission cost for each layer of the media stream.

Table 6.3: STF for each of the clip types

Layer	City	Crew	Harbour	Sintel	Soccer
STF	3	6	7	3	3

6.1.2 Final Selection

1. Based on the factors as previously highlighted, let us now define the number of iterations per subjective testing session. The selected choices are:

   A Eight layer SVC encoding.

   B One loss rates: 10%.

   C One GOP frame values: value of one.

   D Seven Transmission Schemes: SVC, MDC, MDC-SDP, MDC-SD, SDC-SDP-SD, ALD and ALD-IER-2.

   E Five media clips: city, crew, harbour, soccer and sintel.

   F One resolution encoding: (2, 3, 3).

   G One range of QP values: same range defined for all.

   H ALD - STF values specific to each media clip.

Table 6.4: Stream Model Allocation per Clip Number, based on models: SVC, MDC, MDC-SDP, MDC-SD, SDC-SDP-SD, ALD and ALD-IER-2

	Duration	Clip Number						
Iteration	secs	1	2	3	4	5	6	7
1 (city)	10	MDC	SDC-SDP-SD	ALD-IER-2	MDC-SD	SVC	ALD	MDC-SDP
2 (crew)	10	MDC-SDP	ALD	SDC-SDP-SD	SVC	MDC	ALD-IER-2	MDC-SD
3 (harbour)	10	SDC-SDP-SD	ALD-IER-2	MDC	MDC-SD	ALD	MDC-SDP	SVC
4 (sintel)	52	ALD	SVC	ALD-IER-2	MDC-SDP	SDC-SDP-SD	MDC	MDC-SD
5 (soccer)	10	MDC	SVC	MDC-SD	SDC-SDP-SD	ALD-IER-2	MDC-SDP	ALD

I Finally, two clip durations are chosen, 10 seconds and 52 seconds, so as to provide the test subjects with varying viewing times on which to base their streaming model grading and ranking.

This selection of options provides one streaming model for SVC and SDC, two stream models for ALD and three stream models for MDC.

2. This defined list gives us a total of thirty five clips, based on five different seven clip iterations. The thirty five clips are composed of 28 ten second clips and 7 fifty-two second clips. Giving an expected total test time of approximately 25 minutes.

   Note: If we were to include one additional choice from any of the encoding options (layer size encoding, loss rate, GOP value, QP range, STF or resolution encoding), this would double our test time, which is undesirable for practical reasons.

3. Table 6.4 defines the allocation of each streaming model per iteration to the appropriate clip number in the subjective test list. The test subjects were only shown the clip number and were unaware of the relative streaming model, thus limiting the opportunity for grading clips (models) based on observed quality in previous iterations.

6.1.3 Questions to be Answered

Prior to undertaking our subjective testing, we considered what questions do we want answered:

1. For each of the clips, what is the ranking of the various streaming models?

   More importantly how do the subjective testing results of our streaming models SDC and ALD rank when compared with existing streaming models, SVC and MDC?

   Will the viewable stream quality vary when our streaming techniques IER, SDP and SD are utilised by ALD and MDC?

2. Do the ranking of the various streaming models change when different types of clips are viewed?

Will the clip models all rank the same irrespective of stream type?

3. Will the STF_o values chosen for each ALD clip, change the respective ranking of ALD in the different clip types?

   Will different STF_o values vary the rank position of ALD? It can be assumed that it will, as low STF_o , and thus higher FEC, should increase the consistency of viewable quality.

4. As we have seen, our streaming models and techniques fair well for PSNR and percentage of achievable quality per layer. But PSNR values over the stream length can contain large fluctuation in achievable quality, while still maintaining a high mean PSNR for the entire clip.

   How do the results for PSNR and percentage of achievable quality per layer compare when a visual comparison is undertaken?

5. Once we determine the ranking, we can assess the transmission cost, thus illustrating how achievable quality is based on more than packet loss rate but on how select manipulation of the packetisation of the underlying stream model can increase achievable quality.

   Will streaming models with higher transmission cost and greater levels of FEC provide consistent high levels of viewable quality?

6.1.4 ffmpeg Transcoding Scripts

ffmpeg [129], an open-source multimedia framework was utilised to transcode all the decoded YUV files to MP4 x264 for web compatibility. A sample of the underlying command utilised is illustrated below.

1. Sample command used to create an MP4 x264 file from a YUV file

   ffmpegv1.2 -s 704x576 -r 24 -vcodec rawvideo -f rawvideo -pix_fmt yuv420p -i sintel_trailer-704x576x24fpsx52sec.yuv -vcodec libx264 sintel_trailer-704x576x24
   fpsx52sec.mp4

   A. *-r*: - frame rate

   B. *-f*: - file format

   C. *-pix_fmt*: - YUV pixel format

   D. *-vcodec*: - output codec to use

Figure 6.3: Image of the subjective testing

6.1.5 Design Process Conclusion

We utilised five, well known, distinct ten second clip types, *crew*, *city*, *harbour* and *soccer* videos, all obtained from the well known Leibniz Universität Hannover video library [127] and the fifty two second *sintel* trailer obtained from the Blender Foundation [128]. From these clips, JSVM created scalable encodings based on both the industry standard, Scalable Video Coding (SVC), a well known alternative, Multiple Description Coding (MDC), as well as our patent-pending techniques and models (Adaptive Layer Distribution - ALD, Scalable Description Coding - SDC, Section based Description Packetisation - SDP, Section Distribution - SD and Improved Error Resiliency - IER). Each encoding was based on an eight layer SVC stream, composed of 3 different resolutions and 2 or 3 fidelity levels per resolution. We utilised a packet loss rate of 10% and limited the frame interdependence of the model to one frame per GOP. Thus providing a means of illustrating the effects of packet loss rather than the effects of frame interdependence.

The test was implemented on a web server hosted locally on a set of Apple iMac machines. Eighteen people undertook the subjective test within the confines of Lab 1.22, WGB, UCC on the sixth of June 2013. The test subjects were mainly computer science research staff and PhD students but they did not have experience in video. The test subjects consisted of three women and fifteen men, aged between 22 and 50. Three of the men wore corrective glasses but no additional medical history was requested from the subjects. Figure 6.3 is a photograph that was taken while the subjective testing was in progress. The test was performed in a well lit laboratory. Our subjective testing methodologies follow the recommendations as proposed by ITU-R Rec. BT.500 - Methodology for the subjective assessment of the quality of television

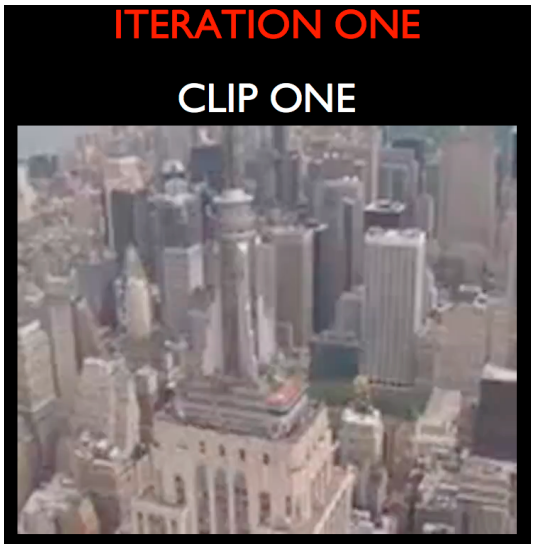
pictures [160], ITU-T Rec. P.910 - Subjective video quality assessment methods for multimedia applications [161], and ITU-T Rec. P.911 - Subjective audiovisual quality assessment methods for multimedia applications [162], but vary in some procedures. While it is typical for a single computer to be used for all subject testing, *i.e.* one computer, statically located within a room, so as to mandate the same lighting levels, the same distance from subject to screen, the same sound levels, the same levels of external distraction and the same conditions for all subjects, in our subjective testing, all subjects undertook the subjective testing at the same time. While variations in lighting levels would have occurred, the same computers (specifications, models, screen sizes) were used, the same distance from screen to subject was enforced and the same instructions were given to all participants. None of our clips contained audio, so that aspect of the subject testing setup could be negated.

Each clip type was shown eight times and for each iteration of clips begins with a viewing of the original clip with no packet loss, thus providing a base case on which the participants could rank/grade the streaming models. Followed by a viewing of each of the seven evaluated streaming models. Finally all eight clips are shown on one screen, so as provide the subjects an opportunity to compare clips. Figure 6.4 provides a snapshot of the *city* subjective testing. Images of the original city clip, streaming model clip 1 and the city comparison screen.

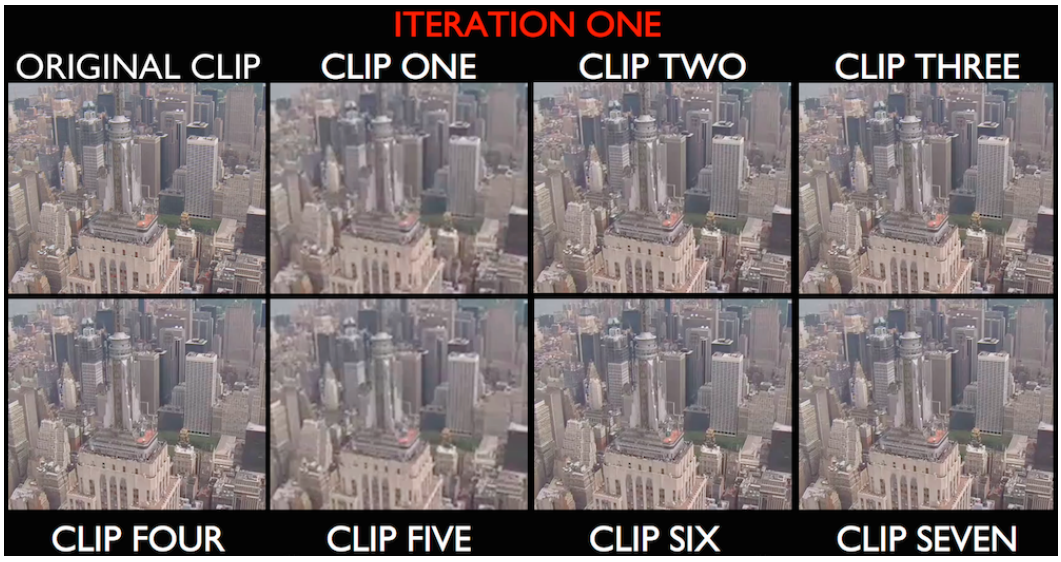
Each streaming model per iteration was graded twice. Once immediately after viewing the streaming model, thus providing the quality value for the individual model per iteration and a second time once all models had been viewed per iteration. As different models may have received the same quality value, the second grading is use to provide a means of ranking the models. For each streaming model, the achievable quality of the stream is based on the maximum layer per frame that contains no visual impairment when compared with the original clip with no packet loss. Details of the display and marking systems are provided in the next section.

In the literature, numerous references were found for scalable subjective testing, but these focused on SVC, examples of which include comparisons between SVC and AVC [163], different SVC codecs [164] and the effects of multi-dimensional scalability [144]. We are unaware of any subjective testing results that compare scalable and description-based coding.

(a) City original clip



(b) City clip 1



(c) City comparison clips

Figure 6.4: A snapshot of the *city* subjective testing. Images of the (a) original city clip, (b) streaming model clip 1 and (c) the city comparison screen are shown

6.2 Trace Analysis

This section details the trace analysis from the underlying design process outlined in the previous section and the results from our subjective testing. As stated, our subjective testing is composed of a number of video clips representing each of our evaluated models. But the clip creation is only the final step of the design process. The trace data from our subjective testing can also be utilised to determine the transmission cost, percentage of viewable frames and achievable PSNR of each clip, with respect to the individual streaming models.

- *Transmission cost per streaming model*: Viewing the overall transmission cost of each of the models relative to a single clip and between clips provides us with a sense of the overall variation that can occur between the streaming models. We begin with Table 6.5, which provides the per clip transmission cost of each streaming model. Costs are broken into data and header, so as to illustrate the increase in header cost for some of the streaming techniques, *e.g.* SDP and IER.

  Items to note include:

  1. Based on the content or the duration of the clip type, the underlying transmission costs of SVC varies. It can be seen that sintel has a very low SVC transmission cost even with a clip duration of 52 seconds, primarily due to the animated contents and the various fading in/out between scenes.

  2. The static QP value we selected during the design process varies the transmission cost of the individual SVC layers, which leads to large differences in the encoded MDC transmission costs.

  3. The differing STF values per clip type provide a means of illustrating the changing transmission cost of ALD. This can be seen where ALD for a specific clip had a low STF value, which mandates a modest decrease in transmission cost relative to MDC, while larger STF values further decrease the overall transmission cost relative to MDC.

  4. Including the packet header transmission costs illustrates the trade off between an increase in the number of viewable frames for the higher layers and the packetisation mechanisms of SD and SDP.

- *Percentage of Viewable Frames per steaming model*: Figure 6.5 plots the percentage of viewable frames for each of the streaming model based on each of the clip types.

  Items to note include:

  1. The viewable quality is increased for streaming models with larger percentages of viewable frames in the higher layers.

Table 6.5: Transmission cost in bytes for each of the clip types, based on the streaming models: SVC, MDC, MDC-SDP, MDC-SD, SDC-SDP-SD, ALD and ALD-IER-2. Total cost of transmission is broken into *Data* cost and *Header* cost for each streaming model.

Clip Type	Cost	SVC	MDC	MDC-SDP	MDC-SD	SDC-SDP	ALD	ALD-IER-2
City	Data	18,492k	32,798k	32,798k	32,798k	27,904k	24,902k	25,751k
	Header	845k	1,437k	1,878k	1,437k	1,292k	1,150k	1,172k
	Total	19,337k	34,235k	34,676k	34,235k	29,196k	26,052k	26,923k
Crew	Data	9,484k	19,334k	19,334k	19,334k	15,808k	12,076k	12,395k
	Header	471k	872k	1,301k	872k	758k	626k	640k
	Total	9,955k	20,206k	20,635k	20,206k	16,566k	12,702k	13,035k
Harbour	Data	18,597k	37,516k	37,516k	37,516k	29,429k	23,374k	23,900k
	Header	853k	1,619k	2,234k	1,619k	1,357k	1,080k	1,080k
	Total	19,450k	39,135k	39,750k	39,135k	30,786k	24,454k	24,980k
Sintel	Data	14,361k	31,451k	31,451k	31,451k	23,465k	21,091k	21,612k
	Header	946k	1,648k	5,046k	1,648k	1,847k	1,375k	1,393k
	Total	15,307k	33,099k	36,497k	33,099k	25,312k	22,466k	23,005k
Soccer	Data	11,742k	21,225k	21,225k	21,225k	17,988k	15,893k	16,475k
	Header	562k	958k	1,414k	958k	853k	771k	792k
	Total	12,304k	22,183k	22,639k	22,183k	18,841k	16,664k	17,267k

Table 6.6: Streaming model PSNR dB values for each clip type, based on the specified 10% packet loss rate.

Clip Type	SVC	MDC	MDC-SDP	MDC-SD	SDC-SDP	ALD	ALD-IER-2
City	26.13	26.71	33.72	35.13	34.07	34.56	36.00
Crew	31.40	33.47	37.82	37.57	37.41	36.70	37.56
Harbour	24.67	25.82	34.32	35.55	33.85	34.10	35.37
Sintel	43.34	45.99	48.05	48.20	46.95	47.85	48.46
Soccer	29.32	32.21	36.61	36.57	36.31	36.19	37.34

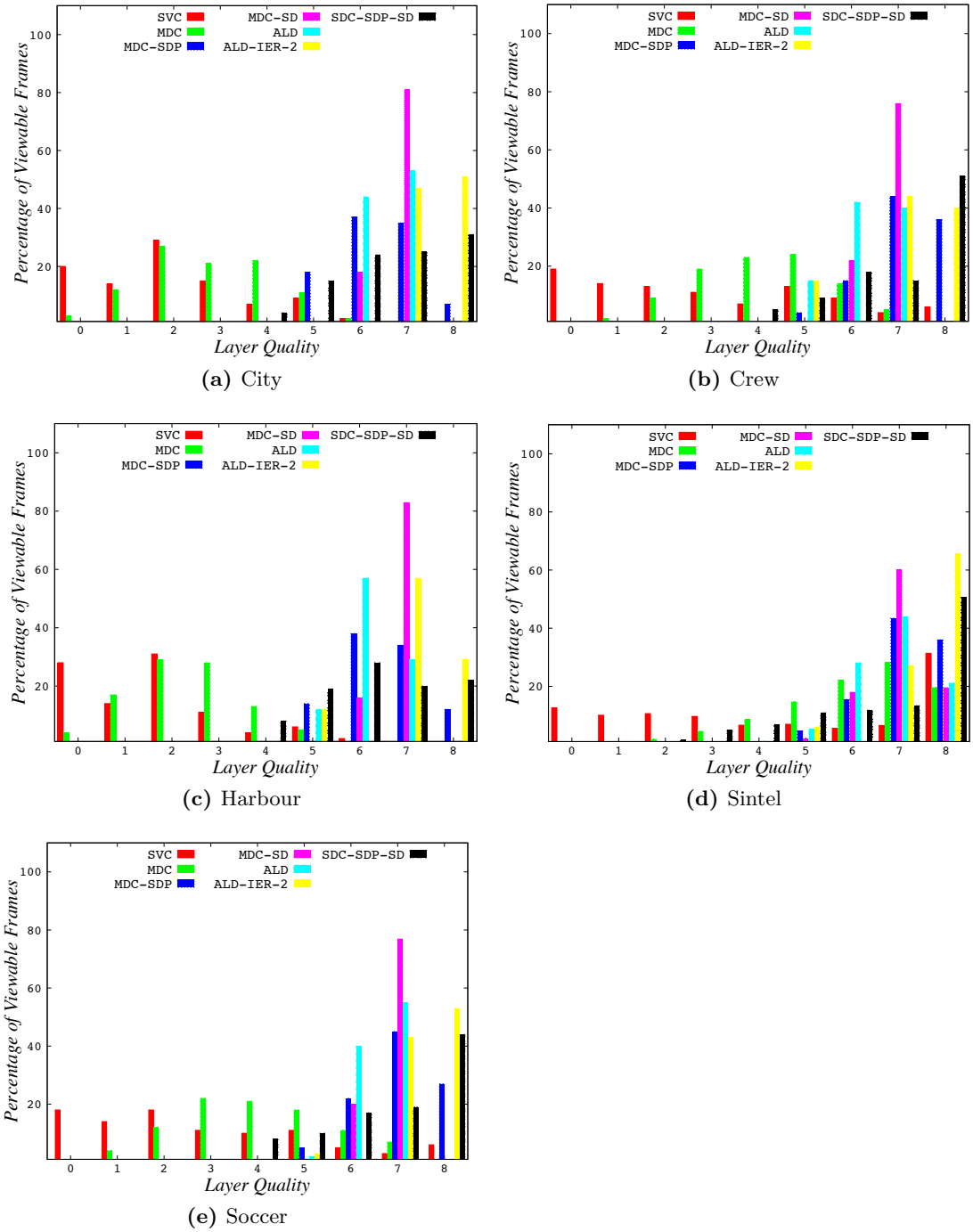
2. The streaming models with the highest achievable quality over all clip types are: MDC-SD, ALD-IER-2, ALD and MDC-SDP, in varying orders, this illustrates how our models and techniques mandate high levels of viewable quality.

3. The difference in quality between MDC and MDC-SD is the packetisation mechanism of SD. The transmission cost is identical for both schemes. MDC-SDP has a slight increase in the header cost relative to MDC but provides for greater overall viewing quality.

4. As seen in Table 6.5, ALD-IER-2 mandates a marginal increase in transmission cost for all clip types, but provides distinct increases in the number of viewable frames for the higher layers.

- *Trace Data PSNR values*

  Table 6.6 provides the streaming model PSNR values for each clip type, based on the specified packet loss rate of 10%. It can be seen that these PSNR values



(a) City **(b)** Crew

(c) Harbour **(d)** Sintel

(e) Soccer

Figure 6.5: Percentage of viewable frames for each of the streaming models per clip types

Table 6.7: Ranking of streaming models per clip type based on PSNR values from Table 6.6.

Clip Type	1	2	3	4	5	6	7
City	ALD-IER-2	MDC-SD	ALD	SDC-SDP	MDC-SDP	MDC	SVC
Crew	MDC-SDP	MDC-SD	ALD-IER-2	SDC-SDP	ALD	MDC	SVC
Harbour	MDC-SD	ALD-IER-2	MDC-SDP	ALD	SDC-SDP	MDC	SVC
Sintel	ALD-IER-2	MDC-SD	MDC-SDP	ALD	SDC-SDP	MDC	SVC
Soccer	ALD-IER-2	MDC-SDP	MDC-SD	SDC-SDP	ALD	MDC	SVC

are somewhat reflective of the percentage of viewable quality seen in Figure 6.5. Table 6.7 re-orders the streaming models and creates a streaming model ranking based on PSNR. Thus providing a means of comparing the streaming model ranking to the subjective testing ranking later in the chapter.

Items to note include:

1. For all clip types SVC and MDC have the lowest PSNR values. This illustrates that SVC and MDC either stream at the lower quality levels, or vary the quality levels, during decoding. Both outcomes reduce overall viewable quality.
2. The PSNR values for MDC-SD and MDC-SDP are near consistent for each media clip, with the exception of City and Harbour, demonstrating that the achievable quality may be dependent on the clip type and underlying content.
3. The PSNR values of SDC-SDP-SD fair very well, and is even better than ALD in some instances. Thus indicating that the perceived quality of SDC-SDP-SD may be better than the results seen in our evaluated quality of SDC-SDP-SD.
4. ALD-IER-2 outperforms all other streaming models in most clip types due to the error protection mechanism of IER, as well as the underlying packetisation and description-based distribution method inherent in ALD.
5. Overall, the PSNR values for SDC, ALD and ALD-IER-2 are consistent with MDC-SD and MDC-SDP, even though the transmission cost of SDC, ALD and ALD-IER-2 are noticeably lower than the MDC variants.

6.3 Subjective Testing Results

In this section we present the results of our subjective testing. We outline the clip iteration scheme and the associated marking/grading utilised. We present comments on the viewable quality of the various models from our test subjects and finally we present the results of our subjective testing. We also answer the questions posed in the "Design Process" section. We begin this section with an overview of our grading/ranking scheme.

6.3.1 Grading/Ranking scheme

Each of the video clip types is denoted as an individual iteration, while for each iteration, a number of evaluated models are displayed, *e.g.* once for the original unaltered clip, seven times for the evaluated streaming models and once for a screen which contains all eight clips, thus providing a means of close comparison. The order in which we display the models per iteration and how we mark the various clips per iteration, needs to be considered:

1. For each clip iteration we will alter the streaming model display allocation, so that each iteration randomises the position of the different models, *e.g.* so that SVC is not always shown first, MDC second, etc.
2. Per iteration, we mark each clip on a scale of one to five, and at the each iteration, we ask the viewer to rate all seven clips in a scale of one to seven. With one being the best clip, or least annoying, and seven being the worst clip. We may notice that the same streaming model in different clips is rated differently.
3. Overall, we ask the test subjects to offer comments on the underlying quality of the various clips. What aspect of the clip or the underlying content of the clip annoyed them the most, what increased perceived quality and what would improve their viewing pleasure. The comments provided by our test subjects are detailed in Section B.
4. At the end of each iteration, we should not inform the viewers which clip belongs to which encoding, as that may influence them to try and choose specific models in future tests cases.
5. For the current subjective testing we are not interested in streaming quality versus transmission cost and as such are not concerned with optimising the encoding cost of SVC.

6.3.1.1 Subjective Testing Clip Comments

In this Section, we provide a list of comments provided by our test subjects during our subjective testing. The comments are grouped per clip type:

A. Test subjects comments on City

1. Hard to rank some clips, very similar
2. Some noise apparent in all clips, but more obvious in clips 1, 2 and 5
3. Clip 5 (SVC) pops in and out of sharpness, very distracting.
4. Bright and smooth, I can easily distinguish the quality
5. Clip 1 (MDC) - met life (building label) can't be read. Water and land not so differentiated. Clip 5 (SVC) terrible - lots of pressure on the eyes. Clips 4, 6, 7 - only colour issues (only lighter than the original)

B. Test subjects comments on Crew

1. Having multiple people (in the clip) drew more attention to those further back, and distortions were more visible on their faces (In particular the first blue uniformed man's face was always noticeably more blurred than the original)
2. Lost detail on fence on left of early frame in all clips apart from original
3. This iteration is quite hard. Even the worst looks okay as it flickers

C. Test subjects comments on Harbour

1. All of them nearly similar, except 3 and 7
2. clip 3 hurts the eyes

D. Test subjects comments on Sintel

1. Videos out of sync when viewed together
2. The snowy first scene always looked bad. Distortion on flickering of the text in many clips was also annoying
3. Very little difference for ranking. Found it very difficult to find differences.
4. Clip 4 and 7 similar but colours worse in 4
5. The anime was the hardest to determine differences. I had to watch it several times.

E. Test subjects comments on Soccer

1. The fence in the background was the source of the most noticeable quality drops.
2. Clip 2 jitter frame-rate
3. All looks good except 1 and 2

F. Test subjects comments on All Clips

1. Note for all clips, the original contains some error/blurring/artefacts, etc. which made determining the errors in the streaming model clips more difficult, as the original had to be re-referenced on numerous occasions.
2. Note for all clip, noticing that colour tends to be more washed out in clips 4, 5, 6, 7 for all iterations

6.3.1.2 Grading Scheme

We implement two grading schemes, based on test methods from the ITU-T recommendation document: P.910 : Subjective video quality assessment methods for multimedia applications [165]. Both schemes are based on a rating of 1 to 5 inclusive, but one is based on Impairment, while the other is based on Quality, both illustrated in Table 6.8. *Impairment* is based on how much variation or bad quality a test subject can see, with *Quality* based on the level of viewable quality a test subject can see.

Table 6.8: Both grading schemes and their respective ratings

Impairment	Quality	Grade
Imperceivable	Excellent	5
Perceptible, but not annoying	Good	4
Slightly annoying	Fair	3
Annoying	Poor	2
Very annoying	Bad	1

Table 6.9: Mean grading results of each clip number as per the stream model allocation as outline in Table 6.4

Clip Type	1	2	3	4	5	6	7
City	1.47	3.75	4.55	3.83	1.16	3.66	3.41
Crew	4.09	3.75	3.86	1.66	2.25	3.83	3.61
Harbour	3.66	4.16	1.27	4.4	3.75	3.08	1.16
Sintel	3.48	1.44	3.94	3.75	3.22	2.48	4
Soccer	2.36	1.33	4.58	3.88	4.44	4.11	4.08

Table 6.10: Stream model ranking based on mean grading results for each clip number

Clip Type	1	2	3	4	5	6	7
City	ALD-IER-2	MDC-SD	SDC-SDP	ALD	MDC-SDP	MDC	SVC
Crew	MDC-SDP	SDC-SDP	ALD-IER-2	ALD	MDC-SD	MDC	SVC
Harbour	MDC-SD	ALD-IER-2	ALD	SDC-SDP	MDC-SDP	MDC	SVC
Sintel	MDC-SD	ALD-IER-2	MDC-SDP	ALD	SDC-SDP	MDC	SVC
Soccer	MDC-SD	ALD-IER-2	MDC-SDP	ALD	SDC-SDP	MDC	SVC

Table 6.11: Mean ranking value results.

Clip Type	SVC	MDC	MDC-SDP	MDC-SD	SDC-SDP	ALD	ALD-IER-2
City	6.72	5.94	3.83	2.38	3.11	2.94	1.38
Crew	6.88	5.88	1.66	2.94	2.44	2.50	2.55
Harbour	6.61	6.05	4.16	1.50	3.00	3.16	1.88
Sintel	6.44	5.72	2.88	2.05	4.11	2.83	2.11
Soccer	6.88	6.05	2.88	1.77	3.27	3.44	1.66

6.3.1.3 Grading/Ranking Results

Once the subjective testing was complete, we tabulate the grading and ranking values. In Table 6.9 we present the mean grading results of each clip number as per the stream model allocation as outline in Table 6.4 We utilised Mean Opinion Score (MOS) to determine these values. The grading values per test subject were widely variable, thus further highlighting the individual, per test subject, nature of subjective testing, with respect to perceived variation and distortion in viewable quality. In Table 6.10 we rank the streaming models based on the mean grading value per clip type, as presented in Table 6.9. The grading ranking are very consistent to Table 6.7, once you take into consideration the very similar PSNR values for the models in Table 6.6.

6.3.1.4 Test Ranking Results

In the test results, some subjects provided ranking based on 1 to 7, while others gave similar clips the same ranking values. To maintain consistency of values over the entire subject base, in ranking scheme where similar values were given to multiple clips, higher values were changed to reflect actual ranking values, *e.g.* a ranking of 1, 2, 2, 2, 3, 3,

Table 6.12: Ranking based on mean ranking results.

Clip Type	1	2	3	4	5	6	7
City	ALD-IER-2	MDC-SD	ALD	SDC-SDP	MDC-SDP	MDC	SVC
Crew	MDC-SDP	SDC-SDP	ALD	ALD-IER-2	MDC-SD	MDC	SVC
Harbour	MDC-SD	ALD-IER-2	SDC-SDP	ALD	MDC-SDP	MDC	SVC
Sintel	MDC-SD	ALD-IER-2	ALD	MDC-SDP	SDC-SDP	MDC	SVC
Soccer	ALD-IER-2	MDC-SD	MDC-SDP	SDC-SDP	ALD	MDC	SVC

4 was changed to 1, 2, 2, 2, 5, 5, 7.

Table 6.11 displays the mean ranking ranking values per iteration (clip type) for each of the streaming models. While Table 6.12 re-orders the streaming models and creates a streaming model ranking based on subjective testing ranking. Again the ranking is very consistent to Table 6.7, again taking into consideration the very similar PSNR values for the models in Table 6.6.

6.3.2 Questions Answered

In this section we provide answers to the questions proposed earlier in this chapter:

1. For each of the clips, what is the ranking of the various streaming models?

   More importantly how do the subjective testing results of our streaming models SDC and ALD rank when compared with existing streaming models, SVC and MDC?

   Will the viewable stream quality vary when our streaming techniques IER, SDP and SD are utilised by ALD and MDC?

A. The ranking of the various streaming models is shown in Table 6.12.

   ALD and SDC compare very favourable to SVC and the various variations of MDC. It is important to remember that our models are comparing very favourably with MDC, even though we have reduced the overall transmission cost.

   Our streaming techniques IER, SDP and SD increase achievable quality for both MDC and ALD, while not overly increasing relative transmission cost.

2. Do the ranking of the various streaming models change when different types of clips are viewed?

   Will the clip models all rank the same irrespective of stream type?

A. The ranking of the various streaming models remains relatively consistent to the PSNR values but ranking and grading values did vary quite considerably dependent on test subject.

3. Will the STF_o values chosen for each ALD clip, change the respective ranking of ALD in the different clip types?

   Will different STF_o values vary the rank position of ALD? It can be assumed that it will, as low STF_o , and thus higher FEC, should increase the consistency of viewable quality.

A. ALD ranking for PSNR, grading and overall subjective ranking remains relatively consistent within the mid-range across all media clip types. The ranking of ALD may be a reflection of the viewable quality of the other streaming models that are ranked higher rather than a failure of the STF_o chosen for ALD itself.

   It is important to note from Figure 6.5 that the layer transitions in ALD remain contained within the higher layers, so while STF provided sufficient FEC protection for the lower layers, additional FEC resilience is required for the higher layer. This additional resilience is reflected in the ALD-IER-2 plots.

4. As we have seen, our streaming models and techniques fair well for PSNR and percentage of achievable quality per layer. But PSNR values over the stream length can contain large fluctuation in achievable quality, while still maintaining a high mean PSNR for the entire clip.

   How do the results for PSNR and percentage of achievable quality per layer compare when a visual comparison is undertaken?

A. There are some variation to the overall model rankings but it is minor. Seeing as the difference in PSNR values for some model is so small, the swopping of model ranking is to be expected.

5. Once we determine the ranking, we can assess the transmission cost, thus illustrating how achievable quality is based on more than packet loss rate but on how select manipulation of the packetisation of the underlying stream model can increase achievable quality.

   Will streaming models with higher transmission cost and greater levels of FEC provide consistent high levels of viewable quality?

A. The answer is no. Table 6.5 illustrates the transmission cost for each streaming model per clip type. Based on the ranking in Table 6.12, we can see that in all clip types ALD-IER-2 compares very favourably (top three for all types) with the other highly ranking streaming models. ALD-IER-2 provides a transmission cost, relative to MDC, of 78% (city), 64% (crew), 63% (harbour), 69% (sintel) and 77% (soccer). So selective packetisation and adaptive error protection rather than higher levels of FEC can provide consistent high levels of viewable quality.

6.4 Discussion

As we have seen, the results of our Scalable Video Subjective Testing supports our simulated experimentation results. While SVC and MDC faired worst, our techniques and models were able to provide levels of consistent high quality viewing, with lower transmission cost, relative to MDC, irrespective of clip type. Our Subjective Testing results highlight the benefits of not only intuitive encoding and transmission but also of selective packetisation. This can be seen in the increase in PSNR and ranking values attained by MDC, with no increase in transmission cost, when SDP and SD are utilised. One item to note is that it was a surprise to see how well SDC-SDP faired in the both grading and ranking values (particularly considering the mean grading values), especially for "City", "Crew" and "Harbour" and especially when we consider the variation in transitions that occur during periods of loss, *i.e.* large drops in the decodable layer value due to loss of the scalable description.

One item to note is that in some instances the grading results for the same clips per iteration were widely variable. So the quality of clip does not only depend on the layer quality achievable, but also on the person viewing the clip.

6.5 Conclusion

Each of the existing schemes contain known design issues which impede their respective deployment. While the adaptable quality is a benefit of SVC, the prioritised hierarchy and its dependency on the base layer is its greatest weakness. As we have highlighted, network transmission issues can affect all packets, and lower layer loss in SVC is detrimental to stream quality. While MDC offers consistent quality, the increased byte-cost of transmission is an inherent weakness. Our new streaming models and techniques provide the framework to achieve the high levels of adaptable stream quality promised by SVC, at a reduced transmission cost relative to MDC.

Chapter 7

Conclusion

The achievable quality of streamed video over the Internet can be affected by constraints in the network and limitations in the decoding device. Adaptation in viewable quality provides a dynamic mechanism by which media streaming can adjust to changes in both network transmission (congestion and subsequent packet loss) and device requirements (WIFI signal strength, device mobility and device capabilities). Multi-bitrate streaming, such as HTTP streaming, enables stream quality adaptation by creating a number of different bitrates of the same video content, *e.g.* such as based on different resolutions, different frame rates, and different quality levels. The video content of each of these bitrates is separated into video segments, thus permitting the decoder to switch between the different bitrate segments to adjust to variation in viewable quality. A limiting factor of HTTP streaming is the transmission of multi-bitrate streaming over TCP, such that each device demands a single reliable connection to the server, thus the ability to use multicast is removed and overall transmission cost over the network increases.

Layered Coding, or Scalable Video, has been proposed as a means of adapting stream quality by adjusting the stream bitrate. Within each scalable video is a number of layers where the selected combination of the layers provides for a means of adaptive viewable quality, thus providing a means of varying the bitrate to adjust viewable quality. As scalable video can be streamed using UDP, multicasting the same clip to heterogeneous devices where each requires varying levels of achievable quality is a means of reducing overall transmission cost over the network and providing the level of granularity demanded by heterogeneous devices. While for a single user scalable video provides the promise of graceful degradation in viewable quality as network loss increases. Scalable coding has been shown to be an especially effective and efficient solution, but it fares badly in the presence of network losses. As we have seen models implementing scalable video, such as SVC and MDC, are susceptible to network loss, such that even small percentages of network loss can mandate noticeable variations in

achievable quality.

Hence the focus of this work is to negate the issues of network loss for scalable video models, to reduce the effects of said loss on viewable quality and to provide a balance between the consistency of achievable quality over the duration of a clip and the increased transmission cost mandated by error resiliency techniques.

7.1 Contributions

Our first contribution is the creation of new segmentation and encapsulation techniques which increase the viewable quality of existing scalable models by fragmenting and re-allocating the video sub-streams based on user requirements, available bandwidth and variations in loss rates. Improved Error Resiliency (IER) from Section 4.2 is an example of this work. Streaming Classes (SC) as presented in Chapter 5 provides a novel framework by which achievable quality can be adapted to suit the needs of the underlying streaming models, with respect to network constraints, user requirements, and transmission cost.

In our second contribution we offer new packetisation techniques which reduce the effects of packet loss on viewable quality by leveraging the increase in the number of frames per group of pictures (*GOP*) and by providing equality of data in every packet transmitted per GOP. These provide novel mechanisms for packetising and error resiliency, as well as providing new applications for existing techniques such as Interleaving and Priority Encoded Transmission. The techniques of Section 3.3, such as provided for packetisation, *e.g.* Section-Based Description Packetisation (SDP) and Section Distribution (SD) are an example of this work. Note: SDP and IER are our new techniques while SD takes inspiration from the well-known techniques, Interleaving and Priority Encoded Transmission.

It is important to note that contribution one and two provide optimisation video delivery techniques that can be utilised by existing scalable coding models to increase viewable quality. From these techniques, we learned that selective packetisation and error resiliency can increase achievable quality and that all aspects of the *stream flow*, *e.g.* the process from encoding to subsequent viewing, should be considered to maximise viewable quality.

Our third contribution is the introduction of three new scalable coding models, which offer a balance between transmission cost and the consistency of viewable quality, while also considering the interaction and interdependence between design options of the stream flow. The goal of this work was to look at scalable video and determine the limitations that exist. From these limitations design new scalable models which offer to reshape the data being transmitted so as to increase achievable quality. We

began with Scalable Description Coding (*SDC*) in Chapter 3 and designed a new model which created a scalable description of higher layer sections by which transmission cost could be reduced while increasing viewable quality. We determined that mandating prioritisation in the scalable description did not provide a sufficient increase in overall quality. We took this limitation in SDC and designed a new model called Adaptive Layer Distribution (*ALD*), as outlined in Chapter 4, as a means of mandating description equality and equality over the packetisation process. ALD distributed the original SVC data over an increased number of descriptions prior to allocating a reduced level of error resiliency per layer, thus lowering transmission cost and reducing the effects of network loss. With ALD we learned that low levels of error resiliency could be allocated per layer and still maintain high levels of viewable quality, but with an increased transmission cost for lower layer quality.

To conclude we reviewed each of the design aspects of our work and focused on all aspects of the stream flow, from which we proposed our novel streaming framework: Streaming Classes (*SC*) in Chapter 5. SC provides a means of dynamic adaptation of all scalable streaming models, be they existing models (SVC and MDC) or our new models (SDC and ALD). SC removes the limitation in ALD by reducing transmission cost required for lower layer quality, but mandates increased complexity in determining the optimal settings for maintaining viewable quality. We learned that maintaining the consistency of achievable quality comes at a cost (prioritisation, increased transmission cost and complexity), but the benefits of consistent achievable quality over the duration of the media stream, for heterogeneous devices irrespective of layer requirement, far out ways these underlying costs. We also learned that loss naturally occurs in the transmission network and simply pushing video data into the network without consideration of said loss, the needs of users, or the optimisation of the stream flow is neither beneficial to the transmission costs of network providers or their clients, or mandates consistent of achievable quality for users.

Finally, as a useful research tool, we built upon the work of myEvalSVC, which is an open source tool for evaluating SVC-based streaming in ns-2. We created new modules for evaluating the existing streaming model MDC, and our new models SDC, ALD and SC, as well for our techniques SDP, SD and IER, as seen in Section 3.6.1. This allowed us to fully understand the impact of datagram loss on achievable quality for all evaluated models. Once our research is completely published, these new modules as well as "modPSNR" will be made available to the academic community.

7.2 Future Work

Due to the inter-dependence of layers/descriptions in adaptive streaming, it is of benefit to the receiver to ascertain without delay which specific descriptions have been

received during transmission. Hence, the device can infer what has been lost and what is ultimately required to maximise stream quality - all remaining descriptions or just a subset. SDC is ideally positioned to benefit from this type of inference, as its architecture consists of a combination of prioritised (D_s and D_{r-nc}), equal importance (D_c) and redundancy (D_r) descriptions, and it offers the potential of maximum stream quality with only a subset of descriptions. With this benefit in mind, SDC can be further improved by optimising the transmission of its descriptions.

Incremental Datagram Delivery (IDD) is proposed as a dynamic transmission protocol, which is based on an incremental delay in the initial transmission of each description in a GOP, with higher priority descriptions being offered transmission precedence. Thus increasing the time available for retransmission of the important descriptions and for the receiving device to infer its ongoing requirements based on the descriptions received so far. In this manner, IDD offers a prioritised transmission scheme which can offer benefits by reducing the occurrence of transmission obstacles such as delivery latency, packet re-ordering, and by increasing QoS.

By employing IDD-based SDC (I-SDC), and by implementing the transmission schedule as outlined in Section 3.1 (first transmitting D_c, then D_s and finally D_r), the device is able to disregard the final description, D_r, should all previous description be received. Table 7.1 shows that I-SDC would attain significant processing savings over SDC, a 35% reduction, by eliminating the need for decoding the redundant SDC description, D_r. As can be seen, the current implementation of I-SDC while decreasing resource usage on the device, is not reducing network consumption, as the final description is only being dropped after being received at the device.

Table 7.1: GOP byte size processed at the device, using an example of 300 bytes per SVC layer

	Number of bytes processed at the device			
	4 Layer		6 Layer	
	Bytes * (Layers or Desc)	Total	B * (L or D)	Total
SVC	300 * 4	1,200	300 * 6	1,800
SDC	625 * 2 + 575 * 1	1,825	735 * 3 + 630 * 1	2,835
I-SDC	625 * 1 + 575 * 1	1,200	735 * 2 + 630 * 1	2,100
MDC	625 * 4	2,500	735 * 6	4,410

It is important to note that while SDC-NC and I-SDC are separate optimisation techniques, they combine quite naturally, to further increase SDC performance.

As previously stated, the High Efficiency Video Coding (*HEVC*) standard, often referenced as *H.265*, is expected to yield an increase in the number of channels of HD video content while halving the bit-rate for the selected quality stream with respect to H.264. H.265 provides this decrease in transmission cost by increasing the complexity of the encoding and decoding. Even though H.265 will decrease the transmission cost

relative to H.264, Ultra High Definition TV (*UHDTV*), commonly known as *4K*, will lead to an increase in overall transmission cost. 4K defines a resolution of approximately 4096 x 2160, or the more commonly used resolution of 3840 x 2160, which equates to a four fold increase in the number of pixels when compared to a full HD screen resolution of 1080p (1920 x 1080). H.265 enabled UHDTV devices, such as the Samsung UE48HU7500 48" TV, are now commercially available. Some streaming services, such as Netflix, provide a selection of their programming in 4K [166]. Amazon will also begin to stream content in 4K in the near future [167]. For the same streaming content, *e.g.* TV episode or movie, albeit at different resolutions, a 4K stream encoded using H.265 will lead to a two fold increase in transmission cost relative to the current streaming cost of H.264. This will lead to both monetary and bandwidth costs for network providers, as well as an increase in the number of multi-bitrate streams encoded and stored by Netflix and similar operators.

Understandably UHDTV and HEVC are new technologies and research challenges still exist to mandate consistent delivery of high quality, such as in the areas of bandwidth management, models to offset packet loss, encoding and decoding complexity, and storage decisions. Due to the number of pixels per square inch in a UHDTV stream, the effects of packet loss on quality will be immediately noticeable. Upscaling from HD content to UHD can lead to noticeable pixelation (square shaped single-colour display components on the screen), while upscaling from standard definition (854x480) content with no packet loss leads to degradation in overall viewable pleasure. Careful selection of resolutions and quality levels proposed during encoding will need careful consideration and in-depth analysis during multi-bitrate stream encoding. To offset the increase in the number of multi-bitrate streams required, proposals for a scalable extension to HEVC are currently under consideration. The techniques and models proposed by our research are beneficial in this regard. As long as the scalable extension to HEVC, which we shall call High Efficiency Scalable Video Coding (*HESVC*), defines a layered structure, which would be consistent with previous iterations of the H.26X standard, then our work can provide the optimal packetisation, error correction and adaptive framework required to provide the balance between transmission cost and achievable quality.

While Software Defined Networks (*SDN*) and Openflow-assisted QoE provide a new delivery technology with which to increase viewable quality, our research does not leverage in-network optimisation techniques or feedback mechanisms. We determined optimality based on defined levels of network loss and then dynamically adapted the error resiliency to suit changes in the network loss rate, such as with SC, or defined static levels of error resiliency with which to combat variation in the network loss rate, such as with ALD. In this regard one aspect of our future work will investigate the benefits of reactive network feedback and stream flow provisioning provided by SDN. As our work is primarily both description-based and packet-based, it is uncomplicated for an

in-network transmission mechanism to selectively drop packets or entire descriptions used by our techniques to suit network and link requirements, with limited effects on viewable quality. SDN provides a means of pre-determining the routes and available bandwidth over which the underlying video is streamed, thus reducing the levels of packet loss that can occur and mandating consistency of quality over the duration of the clip. As SDN controls all flows over finite bandwidth availability, the provisioning of bandwidth per stream must be dynamic in nature otherwise new stream flows cannot be initiated due to diminished levels of available bandwidth. Couple this with the demands of UHD and the decisions underlying the provisioning of bandwidth, stream quality, underlying bitrate, fairness (who should be allocated the best stream quality and for how long), and the dynamic adaption required for scaling the system during moments of increased loading on the network will provide unique research challenges and opportunities.

While not sufficiently developed for inclusion in this thesis, we outlined an in-network architecture for the delivery of streaming media for both multi-bitrate and scalable video, which we called a Centralised Media Streaming Element (*CMSE*) [168]. The aim of a CMSE is to increase the availability and reliability of high quality video delivery for Mobile Devices, while reducing mobile media streaming complexity. Mobile in this regard can be redefined as a device on any constrained network. The concept of a centralised element, available locally within each domain, being able to monitor and adapt stream availability within a constrained network is the basis of this work. CMSE is similar to SDN techniques in that a central node controls streaming data from Server to Client, stream flows are created and adapted to suit network conditions, and multicast groups can be created to reduce bandwidth consumption, while different in that CMSE does not define the network paths on the routers and switches on the network. CMSE acts as a management node rather than a control node, such that the decisions made by CMSE affect the network as a whole, rather than any individual device.

The benefits of CMSE are greatly enhanced by the inclusion of adaptive streaming mechanisms, such as the techniques and models illustrated in this thesis. For the work outlined in this thesis, CMSE would provide a means of determining the current loss rate, optimising the selection of stream quality levels required, initialising multicast groups, and improving the consistency of the achievable stream quality. CMSE would also provide a means of refining the decision making process of SC by increasing the interaction between the different elements of the stream flow.

7.3 Conclusion

Scalable video provides a means for efficient bandwidth utilisation in transmission networks. Its greatest application is to reduce transmission cost when a large number of heterogeneous devices request live streaming such as for concerts, sporting events and TV. Scalable video also offers a mechanism for the stream quality of a single user to adapt in the presence of transmission loss, such that the content of the media is consistently viewable, even though the quality is reduced. But scalable video is unable to cope with variations in loss rates, such that degradation in quality occurs. Clearly SVC is unsuitable for lossy networks, while MDC and hybrid schemes, such as our work, offers increased performance. In this thesis, we investigated new scalable models and techniques by which quality can be maintained in the presence of network loss. We found that the results of our subjective testing support our simulated results and our underlying concepts. Our results allow researchers and practitioners to appreciate the quantitative differences in the key techniques, guiding them in selecting the best approach for their work.

References

[1] Cisco Visual Networking Index: Forecast and Methodology, 2012–2017. Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.
html.

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast, 2013-2018. Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[3] G.J. Conklin, G.S. Greenbaum, K.O. Lillevold, A.F. Lippman, and Y.A. Reznik. Video Coding for Streaming Media Delivery on the Internet. *IEEE Trans. Circuits and Systems for Video Technology*, 11(3):269–281, March 2001.

[4] Netflix encoding requirements available at: http://gizmodo.com/5969677/netflix-encodes-every-movie-120-different-
ways.

[5] T. Stockhammer. Dynamic Adaptive Streaming over HTTP – Standards and Design Principles. In *MMSys '11 Proceedings of the second annual ACM conference on Multimedia systems*, pages 133–144, New York, New York, USA, 2011. ACM Press.

[6] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Trans. Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007.

[7] T. Wiegand, L. Noblet, and F. Rovati. Scalable Video Coding for IPTV Services. *Broadcasting, IEEE Transactions on*, 55(2):527–538, June 2009.

[8] Overview of the Internet Multicast Routing Architecture. Available at: https://tools.ietf.org/html/rfc5110.

[9] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven Layered Multicast. In *Conference Proceedings on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '96, pages 117–130, New York, NY, USA, 1996. ACM.

[10] R. Gopalakrishnan, J. Griffioen, G. Hjalmtysson, C.J. Sreenan, and Su Wen. A Simple Loss Differentiation Approach to Layered Multicast. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 461–469 vol.2, 2000.

[11] Codec Choices for Videoconferencing. Available at: http://www.cisco.com/web /telepresence/collateral/Great_Codec_Debate_White_Paper.pdf.

[12] J. J. Quinlan. TV on the Move: How the Growth in Internet Streaming Influences the Video Quality on your Mobile Device. *The Boolean*, 2014(00):152–156, June 2014.

[13] The Boolean. Available at: http://publish.ucc.ie/boolean/home.

[14] J.J. Quinlan, A.H. Zahran, and C.J. Sreenan. SDC: Scalable Description Coding for Adaptive Streaming Media. In *Proc. of 19th IEEE International Packet Video Workshop (PV2012)*, pages 59–64, May 2012.

[15] J.J. Quinlan, A.H. Zahran, and C.J. Sreenan. ALD: Adaptive Layer Distribution for Scalable Video. *Proc. of ACM Multimedia Systems (MMSys'13)*, pages 202–213, February 2013.

[16] J.J. Quinlan, A.H. Zahran, and C.J. Sreenan. ALD: Adaptive Layer Distribution for Scalable Video. *Multimedia Systems*, pages 1–20, 2014.

[17] J.J. Quinlan, A.H. Zahran, and C.J. Sreenan. Method and System for Scalable Description Coding for Adaptive Streaming of Data. *Patent pending - Application number: PCT/EP2013/059686*, November 2013.

[18] D. Wu, Y. Hou, W. Zhu, Y. Zhang, and J. Peha. Streaming Video Over the Internet: Approaches and Directions. *IEEE Trans. Circuits and Systems for Video Technology*, 11(3):282–300, Mar. 2001.

[19] T. Wiegand and G.J. Sullivan. The H.264/AVC Video Coding Standard [Standards in a Nutshell]. *Signal Processing Magazine, IEEE*, 24(2):148–153, March 2007.

[20] H.264 : Advanced Video Coding for Generic Audiovisual Services. Available at: http://www.itu.int/rec/T-REC-H.264.

[21] D. Marpe, T. Wiegand, and G.J. Sullivan. The H.264/MPEG4 Advanced Video Coding Standard and its Applications. *Communications Magazine, IEEE*, 44(8):134–143, Aug 2006.

[22] P. Frojdh, U. Horn, M. Kampmann, A. Nohlgren, and M. Westerlund. Adaptive Streaming Within the 3GPP Packet-switched Streaming Service. *IEEE Network*, 20(2):34–40, 2006.

[23] Wei Pu, Zixuan Zou, and Chang Wen Chen. Dynamic Adaptive Streaming over HTTP from Multiple Content Distribution Servers. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5, Dec 2011.

[24] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen. Scalable Internet Video using MPEG-4. *Signal Processing: Image Communication*, 15:95–126, 1999.

[25] T. Schierl, T. Stockhammer, and T. Wiegand. Mobile Video Transmission Using Scalable Video Coding. *IEEE Trans. Circuits and Systems for Video Technology*, 17(9):1204 – 1217, 2007.

[26] I. Unanue, I. Urteaga, R. Husemann, J. Del Ser, V. Roesler, A. Rodriguez, and P. Sanchez. A Tutorial on H.264/SVC Scalable Video Coding and its Tradeoff between Quality, Coding Efficiency and Performance. In *Recent Advances on Video Coding*, pages 1–24. InTech, June 2011.

[27] R. Singh, A. Ortega, L. Perret, and W. Jiang. Comparison of Multiple Description Coding and Layered Coding based on Network Simulations. In *10] N. Maxemchuk, Dispersity Routing in Store and Forward Networks*, pages 929–939, 2000.

[28] Y.C. Lee, J. Kim, Y. Altunbasak, and R.M. Mersereau. Layered Coded vs. Multiple Description Coded Video over Error-prone Networks. In *Signal Processing: Image Communication*, pages 337–356, 2003.

[29] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560–576, July 2003.

[30] High Efficiency Video Coding (HEVC). Available at: http://hevc.info/.

[31] G.J. Sullivan, J. Ohm, Woo-Jin Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(12):1649–1668, Dec 2012.

[32] J. Ohm, G.J. Sullivan, H. Schwarz, Thiow Keng Tan, and T. Wiegand. Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC). *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(12):1669–1684, Dec 2012.

[33] G. Ghinea and J.P. Thomas. Quality of Perception: User Quality of Service in Multimedia Presentations. *IEEE Trans. on Multimedia*, 7(4):786–789, 2005.

[34] ITU-T SVC Standard. Available at: http://www.itu.int/ITU-T/recommendations/index. aspx?ser=H.

[35] Ye-Kui Wang, M.M. Hannuksela, S. Pateux, A. Eleftheriadis, and Stephan Wenger. System and Transport Interface of SVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1149–1163, Sept 2007.

[36] Draft RTP Payload Format for SVC. Available at: http://tools.ietf.org/html/draft-ietf-avt-rtp-svc-18#page-6.

[37] S. Wenger, Ye-kui Wang, and M.M. Hannuksela. RTP Payload Format for H.264/SVC Scalable Video Coding. *Journal of Zhejiang University SCIENCE A*, 7(5):657–667, 2006.

[38] T. Schierl, C. Hellge, S. Mirta, K. Gruneberg, and T. Wiegand. Using H.264/AVC-based Scalable Video Coding (SVC) for Real Time Streaming in Wireless IP Networks. In *2007 IEEE International Symposium on Circuits and Systems*, pages 3455–3458. IEEE, 2007.

[39] P. Helle, H. Lakshman, M. Siekmann, J. Stegemann, T. Hinz, H. Schwarz, D. Marpe, and T. Wiegand. A Scalable Video Coding Extension of HEVC. In *Data Compression Conference (DCC), 2013*, pages 201–210, 2013.

[40] J. Nightingale, Qi Wang, and C. Grecos. Scalable HEVC (SHVC)-Based Video Stream Adaptation in Wireless Networks. *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 3573–3577, 2013.

[41] S Xiao, C Wu, Y. Li, and J. Du. Priority Ordering Algorithm for Scalable Video Coding Transmission over Heterogeneous Network. In *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, pages 896–903, March 2008.

[42] R. Puri, K. Ramchandran, and K. Lee. Forward Error Correction (FEC) Codes Based Multiple Description Coding for Internet Video Streaming and Multicast. *Signal Processing: Image Communication*, 16(8):745–762, May 2001.

[43] B. Wang, J. Kurose, P. Shenoy, and D. Towsley. Multimedia Streaming via TCP: an Analytic Performance Study. *ACM Transactions on Multimedia Computing, Communications and Applications*, 4(2), May 2008.

[44] R. Kuschnig, I. Kofler, and H. Hellwagner. Evaluation of HTTP-based Request-response Streams for Internet Video Streaming. *Proc. of ACM Multimedia Systems Conference (MMSys '11)*, pages 245–256, February 2011.

[45] M. Przybylski, B. Belter, and A. Binczewski. Shall We Worry about Packet Reordering? *Computational Methods in Science and Technology*, pages 141–146, December 2005.

[46] D. Loguinov and H. Radha. End-to-End Internet Video Traffic Dynamics: Statistical Study and Analysis. In *IEEE Information Communications Conference (INFOCOM 2002)*, pages 723–732.

[47] W. Chen, U. Mitra, and M.J. Neely. Packet Dropping Algorithms for Energy Savings. *Information Theory, 2006 IEEE International Symposium on*, pages 227–231, 2006.

[48] S. Xiao, H. Wang, and C.-C. Kuo. Priority Ordering and Packetization for Scalable Video Multicast with Network Coding. In Horace H S Ip, Oscar C Au, Howard Leung, Ming-Ting Sun, Wei-Ying Ma, and Shi-Min Hu, editors, *Advances in Multimedia Information Processing – PCM 2007*, volume 4810 of *Lecture Notes in Computer Science*, pages 520–529. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[49] S. Wenger, Y. Wang, and T. Schierl. Transport and Signaling of SVC in IP Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1164–1173, 2007.

[50] R. Trestian, G. Muntean, and O. Ormond. Signal Strength-based Adaptive Multimedia Delivery Mechanism. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pages 297–300, Oct 2009.

[51] S. Sen, S. Gilani, S. Srinath, S. Schmitt, and S. Banerjee. Design and Implementation of an "Approximate" Communication System for Wireless Media Applications. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 15–26, June 2010.

[52] K. Jamieson and H. Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, volume 37, pages 409–420. ACM, October 2007.

[53] A. Nafaa, T. Taleb, and L. Murphy. Forward Error Correction Strategies for Media Streaming over Wireless Networks. *IEEE Communications Magazine*, 46(1):72–79, January 2008.

[54] V.K. Goyal. Multiple Description Coding: Compression Meets the Network. *Signal Processing Magazine*, 18(5):74–93, Sept 2002.

[55] Y. Wang, A.R. Reibman, and S. Lin. Multiple Description Coding for Video Delivery. *Proceedings of the IEEE*, 93(1):57–70, Jan 2005.

[56] M. Kazemi, S. Shirmohammadi, and K.H. Sadeghi. A Review of Multiple Description Coding Techniques for Error-resilient Video Delivery. *Multimedia Systems*, 20(3):283–309, 2014.

[57] J. Heide, J.H. Sorensen, R. Krigslund, P. Popovski, T. Larsen, and J. Chakareski. Cooperative Media Streaming using Adaptive Network Compression. *2008 Int. Symp. World of Wireless, Mobile and Multimedia Networks, WoWMo. '08*, pages 1 – 7, June 2008.

[58] H. Bai, A. Wang, Y. Zhao, Jeng-Shyang Pan, and A. Abraham. *Distributed Multiple Description Coding.* Principles, Algorithms and Systems. Springer-Verlag New York Inc, November 2011.

[59] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro. Multiple Description Video Coding for Scalable and Robust Transmission over IP. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(3):321–334, March 2005.

[60] C.S. Kim and S. Lee. Multiple Description Coding of Motion Fields for Robust Video Transmission. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(9):999–1010, Sep 2001.

[61] H. Bai, Y. Zhao, and C. Zhu. Multiple Description Video Coding using Adaptive Temporal Sub-Sampling. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1331–1334, 2007.

[62] P. Correia, P. Assuncao, and V. Silva. Enhanced H.264/AVC Video Streaming using Network-adaptive Multiple Description Coding. In *IEEE EUROCON 2011 - International Conference on Computer as a Tool*, pages 1–4. IEEE, 2011.

[63] Y. Wang, M.T. Orchard, V. Vaishampayan, and A.R. Reibman. Multiple Description Coding Using Pairwise Correlating Transforms. *Image Processing, IEEE Transactions on*, 10(3):351–366, 2001.

[64] Zhi Li, A. Khisti, and B. Girod. Forward Error Protection for Low-delay Packet Video. *Packet Video Workshop (PV), 2010 18th International*, pages 1–8, 2010.

[65] I.S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal Society for Industrial and Applied Mathematics*, 8(2):300–304, Jun., 1960.

[66] K. Kirihara, H. Masuyama, S. Kasahara, and Y. Takahashi. FEC Recovery Performance for Video Streaming Services Based on H. 264/SVC. *Recent Advances on Video Coding, InTech*, 2011.

[67] J.H. Sorensen, J. Ostergaard, P. Popovski, and J. Chakareski. Multiple Description Coding with Feedback Based Network Compression. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6, 2010.

[68] R. Puri. Multiple Description Source Coding using Forward Error Correction Codes. *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers.*, 1:342–346, August 1999.

[69] A. Shokrollahi. Raptor Codes. *Information Theory, IEEE Trans. on*, 52(6):2551–2567, 2006.

[70] S. Winkler and P. Mohandas. The evolution of video quality measurement: From psnr to hybrid metrics. *Broadcasting, IEEE Transactions on*, 54(3):660–668, Sept 2008.

[71] D Salomon. *Guide to Data Compression Methods*. Springer, 2002.

[72] B. Girod. Digital images and human vision. chapter What's Wrong with Mean-squared Error?, pages 207–220. MIT Press, Cambridge, MA, USA, 1993.

[73] G. Van Wallendael, S. Van Leuven, J. De Cock, P. Lambert, R. Van de Walle, N. Staelens, and P. Demeester. Evaluation of full-reference objective video quality metrics on high efficiency video coding. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 1294–1299, May 2013.

[74] A. Lindgren, A. Almquist, and O. Schelen. Evaluation of quality of service schemes for ieee 802.11 wireless lans. In *Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on*, pages 348–351, 2001.

[75] An Chan, Kai Zeng, P. Mohapatra, Sung-Ju Lee, and S. Banerjee. Metrics for evaluating video streaming quality in lossy ieee 802.11 wireless networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.

[76] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang. A quest for an internet video quality-of-experience metric. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pages 97–102, New York, NY, USA, 2012. ACM.

[77] Zhong Ren, Chen-Khong Tham, Chun-Choong Foo, and Chi-Chung Ko. Integration of mobile ip and multi-protocol label switching. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 7, pages 2123–2127 vol.7, 2001.

[78] I. Mahadevan and K.M. Sivalingam. Quality of service architectures for wireless networks: Intserv and diffserv models. In *Parallel Architectures, Algorithms, and Networks, 1999. (I-SPAN '99) Proceedings. Fourth InternationalSymposium on*, pages 420–425, 1999.

[79] V.K. Adhikari, Yang Guo, Fang Hao, M. Varvello, V. Hilt, M. Steiner, and Zhi-Li Zhang. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *INFOCOM, 2012 Proceedings IEEE*, pages 1620–1628, March 2012.

[80] Lihong Xu and Shuhua Ai. A new feedback control strategy of video transmission based on rtp. In *Industrial Electronics and Applications, 2006 1ST IEEE Conference on*, pages 1–4, May 2006.

[81] H.M. Radha, M. van der Schaar, and Yingwei Chen. The mpeg-4 fine-grained scalable video coding method for multimedia streaming over ip. *Multimedia, IEEE Transactions on*, 3(1):53–68, Mar 2001.

[82] U.H. Reimers. Dvb-the family of international standards for digital video broadcasting. *Proceedings of the IEEE*, 94(1):173–182, Jan 2006.

[83] The Broadcasters Guide To SMPTE 2022. Available At: http://www.artel.com/docs/default-source/whitepapers-and-articles/broadcasters-guide-to-smpte-2022.pdf.

[84] M. Etoh and T. Yoshimura. Advances in wireless video delivery. *Proceedings of the IEEE*, 93(1):111–122, Jan 2005.

[85] K.J. Ma, R. Bartos, S. Bhatia, and R. Nair. Mobile video delivery with http. *Communications Magazine, IEEE*, 49(4):166–175, April 2011.

[86] E. Ryu and S.W. Han. Priority-based Selective H.264 SVC Streaming over Erroneous Converged Networks. *Multimedia Tools and Applications*, 68(2):337–353, May 2012.

[87] C. Muller, D. Renzi, S. Lederer, S. Battista, and C. Timmerer. Using Scalable Video Coding for Dynamic Adaptive Streaming over HTTP in Mobile Environments. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 2208–2212, Aug 2012.

[88] I. Sodagar. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *MultiMedia, IEEE*, 18(4):62–67, April 2011.

[89] H. Kalva, V. Adzic, and B. Furht. Comparing MPEG AVC and SVC for Adaptive HTTP Streaming. In *2012 IEEE International Conference on Consumer Electronics (ICCE)*, pages 158–159. IEEE, 2012.

[90] Y. Sanchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Le Louédec. Efficient HTTP-based Streaming using Scalable Video Coding. *Signal Processing: Image Communication*, 27(4):329–342, April 2012.

[91] Y. Sánchez de la Fuente, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Le Louédec. iDASH: Improved Dynamic Adaptive Streaming over HTTP using Scalable Video Coding. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, pages 257–264, February 2011.

[92] S. Ibrahim, A.H. Zahran, and M.H. Ismail. SVC-DASH-M: Scalable Video Coding Dynamic Adaptive Streaming Over HTTP Using Multiple Connections. In *Telecommunications (ICT), 2014 21st International Conference on*, pages 400–404, Lisbon, Portugal, May 2014.

[93] S. Na and C.M. Kyung. Activity-Based Motion Estimation Scheme for H.264 Scalable Video Coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(11):1475–1485, 2010.

[94] Chia-Wen Lin, Chia-Ming Tsai, Po-Chun Chen, and Li-Wei Kang. Low-overhead Content-adaptive Spatial Scalability for Scalable Video Coding. In *Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on*, pages 235–239, 2013.

[95] Jie Zhao, K. Misra, and A. Segall. Content-adaptive Upsampling for Scalable Video Coding. *Picture Coding Symposium (PCS), 2013*, pages 378–381, 2013.

[96] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. Towards Network-wide QOE Fairness using Openflow-assisted Adaptive Video Streaming. In *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking*, FhMN '13, pages 15–20. ACM, August 2013.

[97] H.E. Egilmez, B. Gorkemli, A.M. Tekalp, and S. Civanlar. Scalable Video Streaming over OpenFlow Networks: An Optimization Framework for QOS Routing. In *2011 18th IEEE International Conference on Image Processing (ICIP 2011)*, pages 2241–2244. IEEE, 2011.

[98] J. Bao, J. Guo, and J. Xu. A Robust Watermarking Scheme for Region of Interest in H.264 Scalable Video Coding. *Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on*, pages 536–538, 2013.

[99] Jung-Hwan Lee and C. Yoo. Scalable ROI Algorithm for H.264/SVC-based Video Streaming. *Consumer Electronics, IEEE Transactions on*, 57(2):882–887, 2011.

[100] K. Park and W. Wang. AFEC: an Adaptive Forward Error Correction Protocol for End-to-end Transport of Real-time Traffic. In *Proceedings of the 7th International Conference on Computer Communications and Networks*, pages 196–205, 1998.

[101] Li Li, Qi Han, and Xiamu Niu. Enhanced Adaptive FEC Based Multiple Description Coding for Internet Video Streaming over Wireless Network. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, pages 478–481, 2010.

[102] C. Hellge, D. Gomez-Barquero, T. Schierl, and T. Wiegand. Layer-Aware Forward Error Correction for Mobile Broadcast of Layered Media. *IEEE Trans. on Multimedia*, 13(3):551–562, 2011.

[103] P.A. Chou, H.J. Wang, and V. N. Padmanabhan. Layered Multiple Description Coding. *IEEE International Packet Video Workshop (PV2003)*, 2003.

[104] L.P. Kondi. A Rate-Distortion Optimal Hybrid Scalable/Multiple-description Video Codec. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '04)*, pages 269–272, 2004.

[105] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan. Priority Encoding Transmission. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 604–612, 1994.

[106] Z. Zhao, J. Ostermann, and Hexin C. Multiple Description Scalable Coding for Video Streaming. *2009 10th Workshop Image Analysis for Multimedia Interactive Services. WIAMIS '09.*, pages 21–24, 2009.

[107] T. Berkin Abanoz and A. Murat Tekalp. SVC-based Scalable Multiple Description Video Coding and Optimization of Encoding Configuration. *Signal Processing: Image Communication*, 24(9):691–701, 2009.

[108] I. Radulovic, Wang Ye-Kui, S. Wenger, A. Hallapuro, M.M. Hannuksela, and P. Frossard. Multiple Description H.264 Video Coding with Redundant Pictures. In *MV '07: Proceedings of the international workshop on Workshop on mobile video*, pages 37 –42. ACM Request Permissions, September 2007.

[109] Z. Zhao and J. Ostermann. Video Streaming Using Standard-compatible Scalable Multiple Description Coding based on SVC. In *17th IEEE International Conference on Image Processing (ICIP)*, pages 1293–1296, 2010.

[110] G. Muntean and L. Murphy. A Novel Feedback Controlled Multimedia Transmission Scheme. In *IEEE Int. Conf. on Telecom., Bucharest, Romania*, pages 123–128, June 2001.

[111] G. Muntean. Efficient Delivery of Multimedia Streams over Broadband Networks using QOAS. *Broadcasting, IEEE Transactions on*, 52(2):230–235, June 2006.

[112] J. Tsai and T. Moors. A Review of Multipath Routing Protocols: From Wireless Ad Hoc to Mesh Networks, 2006.

[113] K. Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A. Hansen, and P. Engelstad. Improving the Performance of Quality-adaptive Video Streaming over Multiple Heterogeneous Access Networks. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, pages 57–68, New York, NY, USA, 2011. ACM.

[114] C. Liu, I. Bouazizi, and M. Gabbouj. Rate Adaptation for Adaptive HTTP Streaming. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, pages 169–174, New York, NY, USA, 2011. ACM.

[115] R. Kibria and J. Kim. H. 264/AVC-based Multiple Description Coding for Wireless Video Transmission. In *Proceedings of the 12th WSEAS International Conference on Communications*, ICCOM'08, pages 429–432, 2008.

[116] C. Grecos and Q. Wang. Video Networking: Trends and Challenges. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '10, pages 17–24, New York, NY, USA, 2010. ACM.

[117] M. Ghareeb, A. Ksentini, and C. Viho. Scalable Video Coding (SVC) for Multipath Video Streaming over Video Distribution Networks (VDN). In *2011 International Conference on Information Networking (ICOIN)*, pages 206–211. IEEE, 2011.

[118] M. Ghareeb, A. Ksentini, and C. Viho. An Adaptive QOE-based Multipath Video Streaming Algorithm for Scalable Video Coding (SVC). In *2011 IEEE Symposium on Computers and Communications (ISCC)*, pages 824–829. IEEE, 2011.

[119] Wei Xiang, Ce Zhu, Chee-Kheong Siew, Yuanyuan Xu, and Minglei Liu. Forward Error Correction-Based 2-D Layered Multiple Description Coding for Error-Resilient H.264 SVC Video Transmission. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(12):1730–1738, Dec 2009.

[120] O.S. Badarneh, Yi Qian, Bo Rong, A.K. Elhakeem, and M. Kadoch. Multiple Description Coding Based Video Multicast over Heterogeneous Wireless Ad Hoc Networks. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1–6, June 2009.

[121] E. Cizeron and S. Hamma. Multipath Routing in MANETs Using Multiple Description Coding. In *Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009. IEEE International Conference on*, pages 282–287, Oct 2009.

[122] M. Halloush and H. Radha. Practical Network Coding for Scalable Video in Error Prone Networks. *Picture Coding Symposium, 2009. PCS 2009*, pages 1–4, 2009.

[123] R. Ahlswede, Cai Ning, S.-Y.R. Li, and R.W. Yeung. Network Information Flow. *IEEE Trans. on Information Theory*, 46(4):1204–1216, July 2000.

[124] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the Air: Practical Wireless Network Coding. *IEEE/ACM Trans. Networking*, 16(3):497–510, June 2008.

[125] C. Fragouli, J. Le Boudec, and J. Widmer. Network Coding. *ACM SIGCOMM Computer Communication Review*, 36(1):63–68, January 2006.

[126] T. Gan, L. Gan, and K. Ma. Reducing Video-quality Fluctuations for Streaming Scalable Video using Unequal Error Protection, Retransmission, and Interleaving. *IEEE Transactions on Image Processing*, 15(4):819–832, 2006.

[127] Video Traces for Network Performance Evaluation. Available at: ftp://ftp.tnt .uni-hannover.de/pub/svc/testsequences/.

[128] Sintel, the Durian Open Movie Project. Available at: http://www.sintel.org/download.

[129] ffmpeg Available at: http://www.ffmpeg.org.

[130] ffmpeg code snippers available at: http://processors.wiki.ti.com/index.php/Open_Source_Vide Tools_-_MPlayer,_FFMpeg,_AviSynth,_MKVToolnix,_MP4Box.

[131] J. Reichel, H. Schwarz, and M. Wien. Joint Scalable Video Model 9 (JSVM 9_19_15). *Joint Video Team, doc.*, Jul. 2007.

[132] The Moving Picture Experts Group. Available at: http://mpeg.chiariglione.org.

[133] The Video Coding Experts Group. Available at: http://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/default.aspx.

[134] E. Francois and J. Vieron. Extended Spatial Scalability : A Generalization of Spatial Scalability for Non Dyadic Configurations. In *Image Processing, 2006 IEEE International Conference on*, pages 169–172, Oct 2006.

[135] K. Turkowski. Graphics Gems. chapter Filters for Common Resampling Tasks, pages 147–165. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[136] S. Jakubczak and D. Katabi. A Cross-Layer Design for Scalable Mobile Video. In *MobiCom '11: Proc. 17th Mobile Computing and Networking Conf.*, pages 289–300, New York, New York, USA, 2011. ACM Press.

[137] A. Detti, G. Bianchi, C. Pisa, F.S. Proto, P. Loreti, W. Kellerer, S. Thakolsri, and J. Widmer. SVEF: an Open-Source Experimental Evaluation Framework for H.264 Scalable Video Streaming. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pages 36–41, 2009.

[138] The Network Simulator - ns-2. Available at: http://nsnam.isi.edu/nsnam/index.php.

[139] C.H. Ke. myEvalSVC: an Integrated Simulation Framework for Evaluation of H.264/SVC Transmission. *KSII Transactions on Internet and Information Systems*, 6(1):379–394, January 2012.

[140] Gnuplot. Available at: http://gnuplot.sourceforge.net.

[141] A. Dutta, J. Chennikara, Wai Chen, O. Altintas, and H. Schulzrinne. Multicasting Streaming Media to Mobile Users. *Communications Magazine, IEEE*, 41(10):81–89, 2003.

[142] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar. Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders. In *Picture Coding Symposium (PCS), 2013*, pages 394–397, Dec 2013.

[143] F. Bossen. Common HM Test Conditions and Software Reference Configurations. In *document JCTVC-L1100 of JCT-VC, Geneva, CH, Jan. 2013.*

[144] A. Eichhorn and Pengpeng Ni. Pick Your Layers Wisely - A Quality Assessment of H.264 Scalable Video Coding for Mobile Devices. In *IEEE International Conference on Communications, 2009. ICC '09.*, pages 1–6, June.

[145] E. Steinbach, N. Färber, and B. Girod. Adaptive Playout for Low Latency Video Streaming. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 962–965, 2001.

[146] C. Sreenan, Jyh-Cheng Chen, P. Agrawal, and B. Narendran. Delay Reduction Techniques for Playout Buffering. *IEEE Transactions on Multimedia*, 2(2):88–100, 2000.

[147] M. Dai, D. Loguinov, and H.M. Radha. Rate-Distortion Analysis and Quality Control in Scalable Internet Streaming. *IEEE Transactions on Multimedia*, 8(6):1135–1146, 2006.

[148] Yuxia Wang, Ting-Lan Lin, and P.C. Cosman. Packet Dropping for H.264 Videos Considering both Coding and Packet-loss Artifacts. *Packet Video Workshop (PV), 2010 18th International*, pages 165–172, 2010.

[149] Helix Mobile Server. Available at: http://www.realnetworks.com/uploadedFiles/helix/resource-library/Whitepaper-Enhanced-Rate-Control.pdf.

[150] D.P. Olshefski, J. Nieh, and D. Agrawal. Inferring Client Response Time at the Web Server. In *Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '02, pages 160–171, New York, NY, USA, 2002. ACM.

[151] IBM Unix network Performance analysis. Available at: http://www.ibm.com/developerworks/aix/library/au-networkperfanalysis/au-networkperfanalysis-pdf.pdf.

[152] G. Muntean. Effect of Delivery Latency, Feedback Frequency and Network Load on Adaptive Multimedia Streaming. In *Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on*, pages 421–427, Oct 2007.

[153] S. Basso, M. Meo, and J.C. De Martin. Strengthening Measurements from the Edges: Application-level Packet Loss Rate Estimation. *SIGCOMM Comput. Commun. Rev.*, 43(3):45–51, July 2013.

[154] Ying Zhang, Zhuoqing Morley Mao, and Ming Zhang. Detecting Traffic Differentiation in Backbone ISPs with NetPolice. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pages 103–115, New York, NY, USA, 2009. ACM.

[155] H.V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An Information Plane for Distributed Services. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 367–380, Berkeley, CA, USA, 2006. USENIX Association.

[156] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, SOSP '01, pages 131–145, New York, NY, USA, 2001. ACM.

[157] Y.A. Wang, C. Huang, J. Li, and K.W. Ross. Queen: Estimating Packet Loss Rate between Arbitrary Internet Hosts. In SueB. Moon, Renata Teixeira, and Steve Uhlig, editors, *Passive and Active Network Measurement*, volume 5448 of *Lecture Notes in Computer Science*, pages 57–66. Springer Berlin Heidelberg, 2009.

[158] R. Rejaie, M. Handley, and D. Estrin. Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the internet. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1337–1345 vol.3, Mar 1999.

[159] A. Khan, L. Sun, and E. Ifeachor. Video quality assessment as impacted by video content over wireless networks. *International Journal on Advances in Networks and Services*, 2:144–154, 2009.

[160] Recommendation ITU-R Rec. BT.500 - a highly controlled environment for rating the video quality of broadcast television signals. Available at: http://www.itu.int/rec/R-REC-BT.500/en.

[161] Recommendation ITU-T Rec. P.910 - a controlled environment for rating the video quality of multimedia services such as videotelephony, videoconferencing and video-on-demand. Available at: http://www.itu.int/rec/T-REC-P.910/en.

[162] Recommendation ITU-T Rec. P.911 - a controlled environment for rating the audiovideo quality of multimedia services such as videotelephony, videoconferencing and video-on-demand. Available at: http://www.itu.int/rec/T-REC-P.911/en.

[163] T. Oelbaum, H. Schwarz, M. Wien, and T. Wiegand. Subjective Performance Evaluation of the SVC Extension of H.264/AVC. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2772–2775, Oct 2008.

[164] J. Lee, F. De Simone, N. Ramzan, Z. Zhao, E. Kurutepe, T. Sikora, J. Ostermann, E. Izquierdo, and T. Ebrahimi. Subjective Evaluation of Scalable Video Coding for Content Distribution. In *Proceedings of the International Conference on Multimedia*, MM '10, pages 65–72, New York, NY, USA, 2010. ACM.

[165] Subjective video quality assessment methods for multimedia applications. Available at: http://www.itu.int/rec/T-REC-P.910/en.

[166] Netflix streaming 4K content. Available at: http://http://gizmodo.com/breaking-bad-is-now-streaming-in-4k-on-netflix-1591610250.

[167] Amazon Prime Streaming 4K in the Near Future: http://www.forbes.com/sites/johnarcher/2014/08/28/amazon-uhd-4k-streams-hitting-samsung-tvs-in-october/.

[168] J.J. Quinlan, A.H. Zahran, and C.J. Sreenan. CMSE: A Network Element for Assistive Media Streaming. In *Proc. of 11th Information Technology and Telecommunication Conference (IT&T)*, pages 122–123, March 2012.

Appendix A

JSVM Results for Variation in GOP Value

In this Appendix, we provide an overview of JSVM GOP design, output and inter layer dependency, with specific reference to the variation that can occur between a GOP value of one and a GOP value of eight. Please note page references are based on the JSVM software manual version 9.19.14. We use the *city* video clip encoded using an eight layer, 300 frame, SVC stream with a (2,3,3) resolution allocation for all examples.

We provide a single example for a GOP value of one, and illustrate an example of two different options for encoding a GOP value of eight.

A.1 JSVM Variables

We first need to understand how JSVM defines variables to control the structure of a GOP for each layer per frame. There are two locations that the GOP structure is defined, these are

1. Main configuration file: The variables in this file are:
   i. *FrameRate* - Maximum frame rate (Hz) of input sequence, page 26.
   ii. *GOPsize* - GOP size (at maximum frame rate), page 27. Dependent on the FrameOut value in the layer configuration file, not all layers may contain the GOPsize number of frames, see pages 63 and 64 for full details.
   iii. *IntraPeriod* - intra period of the encoded sequence, page 27. Default value is -1, which defines that only the first frame is intra coded, *e.g.* is an I-frame, and is not dependent on any other frame.
2. Layer configuration file: The variables in this file are:

i. *FrameRateIn* - input frame rate (Hz) of sequence, page 34.

ii. *FrameRateOut* - output frame rate (Hz) of sequence, page 34

Note: The actual GOP size that is used for encoding a layer is determined by the parameters FrameRate, GOPSize, and FrameRateOut. The actual number of frames that are coded for a layer is determined by the parameters FrameRate and FrameRateOut.

A.2 Example One - GOP value of One

This example is based on the following parameters. Note we use the same FrameRateIn and FrameRateOut values for all eight layer configuration files. In this example we shall illustrate the first 9 frames only (8 frames for the maximum GOP illustrated, plus the next frame to show comparison between GOP values):

1. FrameRate - 30
2. GOPsize - 1
3. IntraPeriod - 1
4. FrameRateIn - 30
5. FrameRateOut - 30

We use the output of three files to determine the various evaluation results. These are 1) sdout.txt, created using "H264AVCEncoderLibTestStatic" and 2) layerTrace.txt and 3) layerRate.txt both created by "BitStreamExtractorStatic"

1. *sdout.txt*

   It can be seen that for each layer per frame, the frame type, the DTQ (dependency_id, temporal_id, quality_id) values, QP value, PSNR values for Y, U and V and total bit rate is provided. Note in this example that the same T value, *e.g.* Temporal value is used because only one frame rate is achievable, *e.g.* as only one frame is grouped per GOP. Also that every layer per frame is an I-frame. Note that we believe the *IK* means an *I*-frame and a *K*ey frame.

```
    frame_type   DTQ    QP_value   Y_PSNR     U_PSNR     V_PSNR     total_bitrate
AU     0: IK   T0 L0 Q0   QP 32   Y 33.1791  U 41.2554  V 42.2076     18288 bit
       0: IK   T0 L0 Q1   QP 26   Y 37.1582  U 42.7726  V 43.9366     17352 bit
       0: IK   T0 L1 Q0   QP 31   Y 33.8355  U 41.6546  V 43.2465     61080 bit
       0: IK   T0 L1 Q1   QP 28   Y 35.8387  U 42.1569  V 43.9621     44688 bit
       0: IK   T0 L1 Q2   QP 26   Y 37.0574  U 42.8566  V 44.4697     37384 bit
       0: IK   T0 L2 Q0   QP 33   Y 33.5901  U 40.9129  V 43.5914    136048 bit
       0: IK   T0 L2 Q1   QP 30   Y 35.2541  U 42.0130  V 44.3455    110384 bit
       0: IK   T0 L2 Q2   QP 28   Y 36.5230  U 42.4217  V 44.6180    109720 bit
AU     1: IK   T0 L0 Q0   QP 32   Y 33.1172  U 41.4350  V 42.2022     18368 bit
       1: IK   T0 L0 Q1   QP 26   Y 37.1930  U 42.7858  V 44.0122     17624 bit
       1: IK   T0 L1 Q0   QP 31   Y 33.7744  U 41.4131  V 43.2845     61344 bit
       1: IK   T0 L1 Q1   QP 28   Y 35.7856  U 42.0472  V 43.8917     45544 bit
```

```
        1: IK   T0 L1 Q2   QP 26   Y 37.0482  U 42.8221  V 44.4286     38672 bit
        1: IK   T0 L2 Q0   QP 33   Y 33.6276  U 41.0901  V 43.8393    131984 bit
        1: IK   T0 L2 Q1   QP 30   Y 35.2613  U 42.0843  V 44.4563    106704 bit
        1: IK   T0 L2 Q2   QP 28   Y 36.5415  U 42.5224  V 44.7207    108936 bit
AU      2: IK   T0 L0 Q0   QP 32   Y 33.2640  U 41.3872  V 42.1310     18488 bit
        2: IK   T0 L0 Q1   QP 26   Y 37.1962  U 42.8391  V 44.0223     17536 bit
        2: IK   T0 L1 Q0   QP 31   Y 33.8270  U 41.4551  V 43.2263     60152 bit
        2: IK   T0 L1 Q1   QP 28   Y 35.8613  U 41.9806  V 43.7946     45144 bit
        2: IK   T0 L1 Q2   QP 26   Y 37.1074  U 42.7086  V 44.3791     38192 bit
        2: IK   T0 L2 Q0   QP 33   Y 33.7681  U 41.1183  V 43.6306    125760 bit
        2: IK   T0 L2 Q1   QP 30   Y 35.3860  U 42.1417  V 44.3833    104584 bit
        2: IK   T0 L2 Q2   QP 28   Y 36.6435  U 42.5900  V 44.6910    105192 bit
AU      3: IK   T0 L0 Q0   QP 32   Y 33.3343  U 41.1636  V 42.3612     18448 bit
        3: IK   T0 L0 Q1   QP 26   Y 37.2878  U 42.8333  V 44.2199     17400 bit
        3: IK   T0 L1 Q0   QP 31   Y 33.8329  U 41.5328  V 43.2296     60248 bit
        3: IK   T0 L1 Q1   QP 28   Y 35.8323  U 42.0940  V 43.8652     44576 bit
        3: IK   T0 L1 Q2   QP 26   Y 37.0632  U 42.7614  V 44.4315     37560 bit
        3: IK   T0 L2 Q0   QP 33   Y 33.7500  U 41.1875  V 43.6657    128752 bit
        3: IK   T0 L2 Q1   QP 30   Y 35.3832  U 42.1931  V 44.4276    104528 bit
        3: IK   T0 L2 Q2   QP 28   Y 36.6123  U 42.6341  V 44.7175    103928 bit
AU      4: IK   T0 L0 Q0   QP 32   Y 33.2819  U 41.2031  V 42.4348     18392 bit
        4: IK   T0 L0 Q1   QP 26   Y 37.3520  U 42.9367  V 44.0899     17560 bit
        4: IK   T0 L1 Q0   QP 31   Y 33.8409  U 41.6654  V 43.3202     60168 bit
        4: IK   T0 L1 Q1   QP 28   Y 35.8595  U 42.2579  V 43.9674     44472 bit
        4: IK   T0 L1 Q2   QP 26   Y 37.0657  U 42.9236  V 44.5969     37416 bit
        4: IK   T0 L2 Q0   QP 33   Y 33.7163  U 41.2260  V 43.8320    127224 bit
        4: IK   T0 L2 Q1   QP 30   Y 35.3400  U 42.2657  V 44.5522    105080 bit
        4: IK   T0 L2 Q2   QP 28   Y 36.6091  U 42.7094  V 44.9455    106936 bit
AU      5: IK   T0 L0 Q0   QP 32   Y 33.2068  U 41.2879  V 42.3525     18208 bit
        5: IK   T0 L0 Q1   QP 26   Y 37.2255  U 42.7293  V 44.1243     17624 bit
        5: IK   T0 L1 Q0   QP 31   Y 33.7656  U 41.5667  V 43.5494     61952 bit
        5: IK   T0 L1 Q1   QP 28   Y 35.7750  U 42.3040  V 44.0904     45344 bit
        5: IK   T0 L1 Q2   QP 26   Y 37.0073  U 42.9044  V 44.6613     38528 bit
        5: IK   T0 L2 Q0   QP 33   Y 33.4617  U 41.1449  V 43.7796    146696 bit
        5: IK   T0 L2 Q1   QP 30   Y 35.1325  U 42.1394  V 44.4597    112584 bit
        5: IK   T0 L2 Q2   QP 28   Y 36.3885  U 42.5643  V 44.6916    111152 bit
AU      6: IK   T0 L0 Q0   QP 32   Y 33.2726  U 41.2702  V 42.4873     18192 bit
        6: IK   T0 L0 Q1   QP 26   Y 37.1595  U 42.9039  V 44.2119     17320 bit
        6: IK   T0 L1 Q0   QP 31   Y 33.7398  U 41.7948  V 43.4961     61952 bit
        6: IK   T0 L1 Q1   QP 28   Y 35.7370  U 42.3682  V 44.0158     45232 bit
        6: IK   T0 L1 Q2   QP 26   Y 37.0062  U 43.0307  V 44.6250     39336 bit
        6: IK   T0 L2 Q0   QP 33   Y 33.4582  U 41.1366  V 43.8271    144176 bit
        6: IK   T0 L2 Q1   QP 30   Y 35.1024  U 42.0979  V 44.3772    111488 bit
        6: IK   T0 L2 Q2   QP 28   Y 36.3874  U 42.6388  V 44.7313    113824 bit
AU      7: IK   T0 L0 Q0   QP 32   Y 33.2119  U 41.4240  V 42.5481     18480 bit
        7: IK   T0 L0 Q1   QP 26   Y 37.2007  U 42.7350  V 44.2515     17560 bit
        7: IK   T0 L1 Q0   QP 31   Y 33.7480  U 41.6320  V 43.3923     63136 bit
        7: IK   T0 L1 Q1   QP 28   Y 35.7761  U 42.2762  V 43.8984     45440 bit
        7: IK   T0 L1 Q2   QP 26   Y 37.0021  U 42.8255  V 44.4843     38336 bit
        7: IK   T0 L2 Q0   QP 33   Y 33.3137  U 40.9071  V 43.5424    152400 bit
        7: IK   T0 L2 Q1   QP 30   Y 35.0068  U 42.0642  V 44.2343    116824 bit
        7: IK   T0 L2 Q2   QP 28   Y 36.2772  U 42.5048  V 44.5102    115632 bit
AU      8: IK   T0 L0 Q0   QP 32   Y 33.1687  U 41.2749  V 42.3331     17872 bit
        8: IK   T0 L0 Q1   QP 26   Y 37.1930  U 42.7684  V 43.9254     17464 bit
        8: IK   T0 L1 Q0   QP 31   Y 33.7116  U 41.5996  V 43.2650     62976 bit
        8: IK   T0 L1 Q1   QP 28   Y 35.6974  U 42.1809  V 43.7919     45528 bit
        8: IK   T0 L1 Q2   QP 26   Y 36.9837  U 42.8044  V 44.3757     39832 bit
        8: IK   T0 L2 Q0   QP 33   Y 33.2699  U 40.8847  V 43.3576    158704 bit
        8: IK   T0 L2 Q1   QP 30   Y 34.9482  U 42.0252  V 44.0975    117704 bit
        8: IK   T0 L2 Q2   QP 28   Y 36.2307  U 42.4482  V 44.3729    117752 bit
```

2. *layerTrace.txt*

   From this file we can see the byte cost and DTQ values per layer. Thus permitting calculations of total transmission cost and the layer dependency per frame.

```
Start-Pos.  Length  LId  TId  QId   Packet-Type  Discardable  Truncatable
==========  ======  ===  ===  ===  ============  ===========  ===========
```

0x0000018d	18	0	0	0	SliceData	No	No
0x0000019f	2277	0	0	0	SliceData	No	No
0x00000a84	2169	0	0	1	SliceData	Yes	No
0x000012fd	7635	1	0	0	SliceData	No	No
0x000030d0	5586	1	0	1	SliceData	Yes	No
0x000046a2	4673	1	0	2	SliceData	Yes	No
0x000058e3	17006	2	0	0	SliceData	No	No
0x00009b51	13798	2	0	1	SliceData	Yes	No
0x0000d137	13715	2	0	2	SliceData	Yes	No
0x000106ca	18	0	0	0	SliceData	No	No
0x000106dc	2287	0	0	0	SliceData	No	No
0x00010fcb	2203	0	0	1	SliceData	Yes	No
0x00011866	7668	1	0	0	SliceData	No	No
0x0001365a	5693	1	0	1	SliceData	Yes	No
0x00014c97	4834	1	0	2	SliceData	Yes	No
0x00015f79	16498	2	0	0	SliceData	No	No
0x00019feb	13338	2	0	1	SliceData	Yes	No
0x0001d405	13617	2	0	2	SliceData	Yes	No
0x00020936	18	0	0	0	SliceData	No	No
0x00020948	2302	0	0	0	SliceData	No	No
0x00021246	2192	0	0	1	SliceData	Yes	No
0x00021ad6	7519	1	0	0	SliceData	No	No
0x00023835	5643	1	0	1	SliceData	Yes	No
0x00024e40	4774	1	0	2	SliceData	Yes	No
0x000260e6	15720	2	0	0	SliceData	No	No
0x00029e4e	13073	2	0	1	SliceData	Yes	No
0x0002d15f	13149	2	0	2	SliceData	Yes	No
0x000304bc	18	0	0	0	SliceData	No	No
0x000304ce	2297	0	0	0	SliceData	No	No
0x00030dc7	2175	0	0	1	SliceData	Yes	No
0x00031646	7531	1	0	0	SliceData	No	No
0x000333b1	5572	1	0	1	SliceData	Yes	No
0x00034975	4695	1	0	2	SliceData	Yes	No
0x00035bcc	16094	2	0	0	SliceData	No	No
0x00039aaa	13066	2	0	1	SliceData	Yes	No
0x0003cdb4	12991	2	0	2	SliceData	Yes	No
0x00040073	18	0	0	0	SliceData	No	No
0x00040085	2290	0	0	0	SliceData	No	No
0x00040977	2195	0	0	1	SliceData	Yes	No
0x0004120a	7521	1	0	0	SliceData	No	No
0x00042f6b	5559	1	0	1	SliceData	Yes	No
0x00044522	4677	1	0	2	SliceData	Yes	No
0x00045767	15903	2	0	0	SliceData	No	No
0x00049586	13135	2	0	1	SliceData	Yes	No
0x0004c8d5	13367	2	0	2	SliceData	Yes	No
0x0004fd0c	18	0	0	0	SliceData	No	No
0x0004fd1e	2267	0	0	0	SliceData	No	No
0x000505f9	2203	0	0	1	SliceData	Yes	No
0x00050e94	7744	1	0	0	SliceData	No	No
0x00052cd4	5668	1	0	1	SliceData	Yes	No
0x000542f8	4816	1	0	2	SliceData	Yes	No
0x000555c8	18337	2	0	0	SliceData	No	No
0x00059d69	14073	2	0	1	SliceData	Yes	No
0x0005d462	13894	2	0	2	SliceData	Yes	No
0x00060aa8	18	0	0	0	SliceData	No	No
0x00060aba	2265	0	0	0	SliceData	No	No
0x00061393	2165	0	0	1	SliceData	Yes	No
0x00061c08	7744	1	0	0	SliceData	No	No
0x00063a48	5654	1	0	1	SliceData	Yes	No
0x0006505e	4917	1	0	2	SliceData	Yes	No
0x00066393	18022	2	0	0	SliceData	No	No
0x0006a9f9	13936	2	0	1	SliceData	Yes	No
0x0006e069	14228	2	0	2	SliceData	Yes	No
0x000717fd	18	0	0	0	SliceData	No	No
0x0007180f	2301	0	0	0	SliceData	No	No
0x0007210c	2195	0	0	1	SliceData	Yes	No
0x0007299f	7892	1	0	0	SliceData	No	No
0x00074873	5680	1	0	1	SliceData	Yes	No
0x00075ea3	4792	1	0	2	SliceData	Yes	No

```
0x0007715b   19050    2    0    0     SliceData          No          No
0x0007bbc5   14603    2    0    1     SliceData         Yes          No
0x0007f4d0   14454    2    0    2     SliceData         Yes          No
```

3. *layerRate.txt*

   This file provides the layer number, the resolution, the frame rate, the cumulative bitrate per second, and DTQ per layer. The minBitrate defines the rate to view the lowest layer per resolution and is calculated by adding the bitrate cost of each lowest layer per selected resolution.

Layer	Resolution	Framerate	Bitrate	MinBitrate	DTQ
0	176x144	30.0000	521.10	521.10	(0,0,0)
1	176x144	30.0000	1024.30		(0,0,1)
2	352x288	30.0000	2648.00	2144.80	(1,0,0)
3	352x288	30.0000	3914.00		(1,0,1)
4	352x288	30.0000	5006.00		(1,0,2)
5	704x576	30.0000	8664.00	5802.80	(2,0,0)
6	704x576	30.0000	11679.00		(2,0,1)
7	704x576	30.0000	14792.00		(2,0,2)

   Together these 3 files allow us to view frame types per layer, transmission cost per layer per frame and achievable frame rates per layer.

4. Transmission cost calculation per GOP

   For a GOP of one, we shall use the first frame:

```
Start-Pos.  Length  LId  TId  QId   Packet-Type  Discardable  Truncatable
==========  ======  ===  ===  ===  ============  ===========  ===========
0x0000018d      18    0    0    0     SliceData          No          No
0x0000019f    2277    0    0    0     SliceData          No          No
0x00000a84    2169    0    0    1     SliceData         Yes          No
0x000012fd    7635    1    0    0     SliceData          No          No
0x000030d0    5586    1    0    1     SliceData         Yes          No
0x000046a2    4673    1    0    2     SliceData         Yes          No
0x000058e3   17006    2    0    0     SliceData          No          No
0x00009b51   13798    2    0    1     SliceData         Yes          No
0x0000d137   13715    2    0    2     SliceData         Yes          No
```

   The transmission cost for this GOP using SVC is the summation over all layers and is 66,877 bytes. While the transmission cost for the first 8 frames is 535,481 bytes.

   For MDC, we use Equation 2.5 from Section 2.4.1, where GOP stands for the number of frames per GOP, $frame$ is an incremental counter over the number of frames per GOP and $L_{l,frame}$ defines the transmission cost for Layer l of frame $frame$ within a GOP.

$$MDC_{D_c} = \sum_{frame=1}^{GOP} \left(\sum_{l=1}^{N} \frac{L_{l,frame}}{l} \right) \tag{A.1}$$

   From this we can determine the transmission cost of one description per each frame in the GOP, *e.g.* Layer 1 is divided by 1, layer 2 by 2, etc. Then to

determine the transmission cost of layer q, we simply multiply by q. In our example, we want to determine the transmission cost of the maximum layer, layer 8, so we multiply by 8. Thus the cost per description is 14,773 bytes and the total cost for this GOP, *e.g.* this single frame, is 118,184 bytes (nearly double), while the total transmission cost for the first 8 frames, *e.g.* 8 GOPs is 947,968 bytes.

A.3 Example Two - GOP value of Eight (Option 1)

The following example is based on the following parameters. Note we use the same FrameRateIn and FrameRateOut values for all eight layer configuration files. In this example we shall illustrate the first 8 frames only:

1. FrameRate - 30
2. GOPsize - 8
3. IntraPeriod - 8
4. FrameRateIn - 30
5. FrameRateOut - 30

Again we use the output of three files to determine the various evaluation results. These are 1) sdout.txt, created using "H264AVCEncoderLibTestStatic" and 2) layerTrace.txt and 3) layerRate.txt both created by "BitStreamExtractorStatic".

1. *sdout.txt*

   In this example, we show the first 9 frames, to illustrate the start of the next GOP, *e.g.* I-frame. Also notice how the frames are now out of order. This is due to the interdependence mandated by the increase in GOP number. It can be seen that the second frame transmitted is frame 8, as this frame is required to decode a subset of all internal frames, *e.g.* frames 4 to 7. The frame hierarchy for the highest enhancement layers created by this encoding is *IBBBBBBBI*, while the hierarchy for layers 1 and 2 is *IPPPPPPPI*. Note how the hierarchy contains no P-frames. If the GOPsize and IntraPeriod are the same no P-frames are encoded. Also note the transmission cost for the first frame in this encoding is the same as the GOP of 1 example. The ninth frame contains a slight increase over GOP of 1, this may be due to the non existence of P-frames. It is only the non I-frames that show the marked reduction in transmission cost.

```
    frame_type   DTQ    QP_value   Y_PSNR     U_PSNR     V_PSNR     total_bitrate
AU     0: IK   T0 L0 Q0   QP 32   Y 33.1791  U 41.2554  V 42.2076     18288 bit
       0: IK   T0 L0 Q1   QP 26   Y 37.1582  U 42.7726  V 43.9366     17352 bit
       0: IK   T0 L1 Q0   QP 31   Y 33.8355  U 41.6546  V 43.2465     61088 bit
       0: IK   T0 L1 Q1   QP 28   Y 35.8387  U 42.1569  V 43.9621     44688 bit
       0: IK   T0 L1 Q2   QP 26   Y 37.0574  U 42.8566  V 44.4697     37384 bit
       0: IK   T0 L2 Q0   QP 33   Y 33.5901  U 40.9129  V 43.5914    136056 bit
       0: IK   T0 L2 Q1   QP 30   Y 35.2541  U 42.0130  V 44.3455    110384 bit
```

```
        0: IK   T0 L2 Q2   QP 28   Y 36.5230  U 42.4217  V 44.6180    109720 bit
AU      8: IK   T0 L0 Q0   QP 32   Y 33.1687  U 41.2749  V 42.3331     17888 bit
        8: IK   T0 L0 Q1   QP 26   Y 37.1930  U 42.7684  V 43.9254     17472 bit
        8: IK   T0 L1 Q0   QP 31   Y 33.7116  U 41.5996  V 43.2650     62992 bit
        8: IK   T0 L1 Q1   QP 28   Y 35.6974  U 42.1809  V 43.7919     45528 bit
        8: IK   T0 L1 Q2   QP 26   Y 36.9837  U 42.8044  V 44.3757     39832 bit
        8: IK   T0 L2 Q0   QP 33   Y 33.2699  U 40.8847  V 43.3576    158720 bit
        8: IK   T0 L2 Q1   QP 30   Y 34.9482  U 42.0252  V 44.0975    117704 bit
        8: IK   T0 L2 Q2   QP 28   Y 36.2307  U 42.4482  V 44.3729    117752 bit
AU      4: P    T1 L0 Q0   QP 35   Y 32.6382  U 41.2181  V 42.3212      1624 bit
        4: P    T1 L0 Q1   QP 29   Y 35.5379  U 42.7720  V 43.9140      1256 bit
        4: B    T1 L1 Q0   QP 34   Y 33.7537  U 41.9022  V 43.6420      4992 bit
        4: B    T1 L1 Q1   QP 31   Y 35.7621  U 42.4334  V 44.2981      3360 bit
        4: B    T1 L1 Q2   QP 29   Y 36.8088  U 43.1049  V 44.8696      2048 bit
        4: B    T1 L2 Q0   QP 36   Y 33.7773  U 41.4382  V 44.0290     21832 bit
        4: B    T1 L2 Q1   QP 33   Y 34.4919  U 42.3566  V 44.6825      3288 bit
        4: B    T1 L2 Q2   QP 31   Y 34.9476  U 42.6983  V 44.9524      6304 bit
AU      2: P    T2 L0 Q0   QP 36   Y 32.5768  U 41.2120  V 42.3344      1008 bit
        2: P    T2 L0 Q1   QP 30   Y 35.4823  U 42.6690  V 43.9070       968 bit
        2: B    T2 L1 Q0   QP 35   Y 33.3564  U 41.6036  V 43.3437      3048 bit
        2: B    T2 L1 Q1   QP 32   Y 35.2230  U 42.1277  V 43.9812      2120 bit
        2: B    T2 L1 Q2   QP 30   Y 36.2503  U 42.7621  V 44.4930      1744 bit
        2: B    T2 L2 Q0   QP 37   Y 33.6567  U 41.1502  V 43.7497     15280 bit
        2: B    T2 L2 Q1   QP 34   Y 34.3709  U 41.8948  V 44.3350      2312 bit
        2: B    T2 L2 Q2   QP 32   Y 34.8103  U 42.2274  V 44.5999      3512 bit
AU      1: P    T3 L0 Q0   QP 37   Y 32.5220  U 41.2478  V 42.1666       704 bit
        1: P    T3 L0 Q1   QP 31   Y 35.3002  U 42.6884  V 43.9169       616 bit
        1: B    T3 L1 Q0   QP 36   Y 33.0374  U 41.5626  V 43.2202      1984 bit
        1: B    T3 L1 Q1   QP 33   Y 34.8656  U 42.0191  V 43.9200      1632 bit
        1: B    T3 L1 Q2   QP 31   Y 35.8773  U 42.6779  V 44.3576      1296 bit
        1: B    T3 L2 Q0   QP 38   Y 33.2144  U 41.0969  V 43.7332     12000 bit
        1: B    T3 L2 Q1   QP 35   Y 33.9904  U 41.7757  V 44.3370      2016 bit
        1: B    T3 L2 Q2   QP 33   Y 34.5157  U 42.0686  V 44.6193      3160 bit
AU      3: P    T3 L0 Q0   QP 37   Y 32.3958  U 41.2487  V 42.4926       896 bit
        3: P    T3 L0 Q1   QP 31   Y 35.0485  U 42.6803  V 43.9950       648 bit
        3: B    T3 L1 Q0   QP 36   Y 33.1099  U 41.6132  V 43.5017      2352 bit
        3: B    T3 L1 Q1   QP 33   Y 34.9189  U 42.0743  V 44.1315      1456 bit
        3: B    T3 L1 Q2   QP 31   Y 35.9154  U 42.6771  V 44.6056      1480 bit
        3: B    T3 L2 Q0   QP 38   Y 33.5758  U 41.1658  V 43.7919      9400 bit
        3: B    T3 L2 Q1   QP 35   Y 34.2982  U 41.8286  V 44.3920      2000 bit
        3: B    T3 L2 Q2   QP 33   Y 34.7605  U 42.0951  V 44.6454      2720 bit
AU      6: P    T2 L0 Q0   QP 36   Y 32.3765  U 41.2504  V 42.5824      1080 bit
        6: P    T2 L0 Q1   QP 30   Y 35.3243  U 42.9798  V 44.0530       904 bit
        6: B    T2 L1 Q0   QP 35   Y 33.2648  U 41.7042  V 43.5093      3120 bit
        6: B    T2 L1 Q1   QP 32   Y 35.1636  U 42.2584  V 44.0127      2112 bit
        6: B    T2 L1 Q2   QP 30   Y 36.2209  U 42.8687  V 44.6180      1632 bit
        6: B    T2 L2 Q0   QP 37   Y 33.0463  U 41.1347  V 43.6993     15096 bit
        6: B    T2 L2 Q1   QP 34   Y 33.9126  U 41.9911  V 44.3474      2440 bit
        6: B    T2 L2 Q2   QP 32   Y 34.4332  U 42.3303  V 44.5897      3904 bit
AU      5: P    T3 L0 Q0   QP 37   Y 32.2358  U 41.3176  V 42.4386       792 bit
        5: P    T3 L0 Q1   QP 31   Y 35.1272  U 42.8454  V 43.9944       928 bit
        5: B    T3 L1 Q0   QP 36   Y 32.9611  U 41.7555  V 43.5493      1872 bit
        5: B    T3 L1 Q1   QP 33   Y 34.8065  U 42.2600  V 44.1414      1688 bit
        5: B    T3 L1 Q2   QP 31   Y 35.8010  U 42.7663  V 44.6663      1496 bit
        5: B    T3 L2 Q0   QP 38   Y 32.9864  U 41.0411  V 43.7787     10376 bit
        5: B    T3 L2 Q1   QP 35   Y 33.8834  U 41.8409  V 44.3889      2048 bit
        5: B    T3 L2 Q2   QP 33   Y 34.4385  U 42.1565  V 44.6013      2928 bit
AU      7: P    T3 L0 Q0   QP 37   Y 32.4802  U 41.3195  V 42.4870       664 bit
        7: P    T3 L0 Q1   QP 31   Y 35.1988  U 42.8082  V 44.0122       616 bit
        7: B    T3 L1 Q0   QP 36   Y 33.0090  U 41.5588  V 43.3987      2328 bit
        7: B    T3 L1 Q1   QP 33   Y 34.7736  U 42.0522  V 43.8777      1552 bit
        7: B    T3 L1 Q2   QP 31   Y 35.7718  U 42.6404  V 44.3871      1120 bit
        7: B    T3 L2 Q0   QP 38   Y 32.5854  U 40.8977  V 43.4558     10920 bit
        7: B    T3 L2 Q1   QP 35   Y 33.4921  U 41.7392  V 44.1066      1888 bit
        7: B    T3 L2 Q2   QP 33   Y 34.0381  U 42.0535  V 44.3258      2488 bit
```

2. *layerTrace.txt*

The frame order is the same as the sdout.txt frame order. Note the reduction in transmission cost with respect to a GOP value of 1.

Start-Pos.	Length	LId	TId	QId	Packet-Type	Discardable	Truncatable
0x0000038b	18	0	0	0	SliceData	No	No
0x0000039d	2277	0	0	0	SliceData	No	No
0x00000c82	2169	0	0	1	SliceData	Yes	No
0x000014fb	7636	1	0	0	SliceData	No	No
0x000032cf	5586	1	0	1	SliceData	Yes	No
0x000048a1	4673	1	0	2	SliceData	Yes	No
0x00005ae2	17007	2	0	0	SliceData	No	No
0x00009d51	13798	2	0	1	SliceData	Yes	No
0x0000d337	13715	2	0	2	SliceData	Yes	No
0x000108ca	18	0	0	0	SliceData	No	No
0x000108dc	2227	0	0	0	SliceData	No	No
0x0001118f	2184	0	0	1	SliceData	Yes	No
0x00011a17	7874	1	0	0	SliceData	No	No
0x000138d9	5691	1	0	1	SliceData	Yes	No
0x00014f14	4979	1	0	2	SliceData	Yes	No
0x00016287	19840	2	0	0	SliceData	No	No
0x0001b007	14713	2	0	1	SliceData	Yes	No
0x0001e980	14719	2	0	2	SliceData	Yes	No
0x000222ff	18	0	1	0	SliceData	Yes	No
0x00022311	194	0	1	0	SliceData	Yes	No
0x000223d3	157	0	1	1	SliceData	Yes	No
0x00022470	624	1	1	0	SliceData	Yes	No
0x000226e0	420	1	1	1	SliceData	Yes	No
0x00022884	256	1	1	2	SliceData	Yes	No
0x00022984	2729	2	1	0	SliceData	Yes	No
0x0002342d	411	2	1	1	SliceData	Yes	No
0x000235c8	788	2	1	2	SliceData	Yes	No
0x000238dc	18	0	2	0	SliceData	Yes	No
0x000238ee	117	0	2	0	SliceData	Yes	No
0x00023963	121	0	2	1	SliceData	Yes	No
0x000239dc	381	1	2	0	SliceData	Yes	No
0x00023b59	265	1	2	1	SliceData	Yes	No
0x00023c62	218	1	2	2	SliceData	Yes	No
0x00023d3c	1910	2	2	0	SliceData	Yes	No
0x000244b2	289	2	2	1	SliceData	Yes	No
0x000245d3	439	2	2	2	SliceData	Yes	No
0x0002478a	18	0	3	0	SliceData	Yes	No
0x0002479c	78	0	3	0	SliceData	Yes	No
0x000247ea	77	0	3	1	SliceData	Yes	No
0x00024837	248	1	3	0	SliceData	Yes	No
0x0002492f	204	1	3	1	SliceData	Yes	No
0x000249fb	162	1	3	2	SliceData	Yes	No
0x00024a9d	1500	2	3	0	SliceData	Yes	No
0x00025079	252	2	3	1	SliceData	Yes	No
0x00025175	395	2	3	2	SliceData	Yes	No
0x00025300	18	0	3	0	SliceData	Yes	No
0x00025312	102	0	3	0	SliceData	Yes	No
0x00025378	81	0	3	1	SliceData	Yes	No
0x000253c9	294	1	3	0	SliceData	Yes	No
0x000254ef	182	1	3	1	SliceData	Yes	No
0x000255a5	185	1	3	2	SliceData	Yes	No
0x0002565e	1175	2	3	0	SliceData	Yes	No
0x00025af5	250	2	3	1	SliceData	Yes	No
0x00025bef	340	2	3	2	SliceData	Yes	No
0x00025d43	18	0	2	0	SliceData	Yes	No
0x00025d55	126	0	2	0	SliceData	Yes	No
0x00025dd3	113	0	2	1	SliceData	Yes	No
0x00025e44	390	1	2	0	SliceData	Yes	No
0x00025fca	264	1	2	1	SliceData	Yes	No
0x000260d2	204	1	2	2	SliceData	Yes	No
0x0002619e	1887	2	2	0	SliceData	Yes	No
0x000268fd	305	2	2	1	SliceData	Yes	No
0x00026a2e	488	2	2	2	SliceData	Yes	No

0x00026c16	18	0	3	0	SliceData	Yes	No
0x00026c28	89	0	3	0	SliceData	Yes	No
0x00026c81	116	0	3	1	SliceData	Yes	No
0x00026cf5	234	1	3	0	SliceData	Yes	No
0x00026ddf	211	1	3	1	SliceData	Yes	No
0x00026eb2	187	1	3	2	SliceData	Yes	No
0x00026f6d	1297	2	3	0	SliceData	Yes	No
0x0002747e	256	2	3	1	SliceData	Yes	No
0x0002757e	366	2	3	2	SliceData	Yes	No
0x000276ec	18	0	3	0	SliceData	Yes	No
0x000276fe	73	0	3	0	SliceData	Yes	No
0x00027747	77	0	3	1	SliceData	Yes	No
0x00027794	291	1	3	0	SliceData	Yes	No
0x000278b7	194	1	3	1	SliceData	Yes	No
0x00027979	140	1	3	2	SliceData	Yes	No
0x00027a05	1365	2	3	0	SliceData	Yes	No
0x00027f5a	236	2	3	1	SliceData	Yes	No
0x00028046	311	2	3	2	SliceData	Yes	No

3. *layerRate.txt*

   Note that now due to the increase in GOP value, the frame rate per layer has increased. Each layer has 4 frame rates, note that the frame rate per layer is based on the FrameRateOut value in the layer configuration file. But if this value is the same as the FrameRate in the main configuration file, the number of frame rates per layer is equal to $n+1$, where n is determined by $2^n = \text{GOPsize}$. In this example we have a GOPsize of 8, which equates to 2^3, which defines the number of frame rates per layer to be $3+1=4$. Please ignore the bitrate costs, as only the first 32 frames were encoded for this example and the bitrate costs reflect this encoding with a lower number of frames.

Layer	Resolution	Framerate	Bitrate	MinBitrate	DTQ
0	176x144	3.7500	68.60	68.60	(0,0,0)
1	176x144	7.5000	75.00	75.00	(0,1,0)
2	176x144	15.0000	83.00	83.00	(0,2,0)
3	176x144	30.0000	94.30	94.30	(0,3,0)
4	176x144	3.7500	134.00		(0,0,1)
5	176x144	7.5000	146.70		(0,1,1)
6	176x144	15.0000	163.10		(0,2,1)
7	176x144	30.0000	185.70		(0,3,1)
8	352x288	3.7500	365.00	299.60	(1,0,0)
9	352x288	7.5000	396.70	325.00	(1,1,0)
10	352x288	15.0000	436.90	356.80	(1,2,0)
11	352x288	30.0000	492.10	400.70	(1,3,0)
12	352x288	3.7500	532.70		(1,0,1)
13	352x288	7.5000	577.10		(1,1,1)
14	352x288	15.0000	634.30		(1,2,1)
15	352x288	30.0000	713.40		(1,3,1)
16	352x288	3.7500	676.50		(1,0,2)
17	352x288	7.5000	731.00		(1,1,2)
18	352x288	15.0000	802.80		(1,2,2)
19	352x288	30.0000	903.70		(1,3,2)
20	704x576	3.7500	1209.30	832.40	(2,0,0)
21	704x576	7.5000	1352.80	946.80	(2,1,0)
22	704x576	15.0000	1541.90	1095.90	(2,2,0)
23	704x576	30.0000	1800.00	1297.00	(2,3,0)
24	704x576	3.7500	1625.60		(2,0,1)
25	704x576	7.5000	1786.00		(2,1,1)
26	704x576	15.0000	1995.00		(2,2,1)
27	704x576	30.0000	2285.00		(2,3,1)
28	704x576	3.7500	2044.00		(2,0,2)
29	704x576	7.5000	2240.00		(2,1,2)
30	704x576	15.0000	2482.00		(2,2,2)

```
31      704x576      30.0000    2817.00              (2,3,2)
```

SVC transmission cost per GOP is 91,069 bytes (sum over all layer for all frames per GOP). While using Equation 4.2 from Chapter 4 we can determine the MDC transmission cost of one description per frame, *e.g.* 20,120 bytes. We multiple this by the layer required, layer 8, thus providing a total MDC transmission cost of 161,680 bytes. Note: we can sum over all frames based on their frame number, sequentially, or we can sum over all frames based on their transmission allocation, non-sequentially, for both options the same transmission cost would be found.

A.4 Example Three - GOP value of Eight (Option 2)

The following example is based on the following parameters. Note we use the same FrameRateIn and FrameRateOut values for all eight layer configuration files. In this example we shall illustrate the first 8 frames only. Note how we have halved the GOPsize value:

1. FrameRate - 30
2. GOPsize - 4
3. IntraPeriod - 8
4. FrameRateIn - 30
5. FrameRateOut - 30

Again we use the output of three files to determine the various evaluation results. These are 1) sdout.txt, created using "H264AVCEncoderLibTestStatic" and 2) layerTrace.txt and 3) layerRate.txt both created by "BitStreamExtractorStatic"

1. *sdout.txt*

   In this example, we show the first 9 frames, to illustrate the start of the next GOP, *e.g.* I-frame. Also notice how the frames are now out of order. This is due to the interdependence mandated by the increase in GOP number. It can be seen that the second frame transmitted is frame 4, as this frame is required to decode frames 1, 2 and 3. Similarly frame 2 is transmitted before frames 1 and 3 for the same reason. By reducing the GOPsize value, we can reduce the waiting time, before frames can be decoded. The frame hierarchy for the highest enhancement layers created by this encoding is *IBBBPBBBI*, while the hierarchy for layers 1 and 2 is *IPPPPPPPI*. Note the existence of the P-frame in the enhancement layers. Also note the transmission cost for the first and ninth frames in this encoding is the same as the GOP of 1 example. It is only the non I-frames that show the marked reduction in transmission cost.

	frame_type	DTQ	QP_value	Y_PSNR	U_PSNR	V_PSNR	total_bitrate
AU	0: IK	T0 L0 Q0	QP 32	Y 33.1791	U 41.2554	V 42.2076	18288 bit
	0: IK	T0 L0 Q1	QP 26	Y 37.1582	U 42.7726	V 43.9366	17352 bit
	0: IK	T0 L1 Q0	QP 31	Y 33.8355	U 41.6546	V 43.2465	61080 bit
	0: IK	T0 L1 Q1	QP 28	Y 35.8387	U 42.1569	V 43.9621	44688 bit
	0: IK	T0 L1 Q2	QP 26	Y 37.0574	U 42.8566	V 44.4697	37384 bit
	0: IK	T0 L2 Q0	QP 33	Y 33.5901	U 40.9129	V 43.5914	136048 bit
	0: IK	T0 L2 Q1	QP 30	Y 35.2541	U 42.0130	V 44.3455	110384 bit
	0: IK	T0 L2 Q2	QP 28	Y 36.5230	U 42.4217	V 44.6180	109720 bit
AU	4: PK	T0 L0 Q0	QP 32	Y 33.0741	U 41.4126	V 42.3288	3696 bit
	4: PK	T0 L0 Q1	QP 26	Y 37.3021	U 42.6972	V 44.0074	17104 bit
	4: PK	T0 L1 Q0	QP 31	Y 33.6715	U 41.8628	V 43.5747	17160 bit
	4: PK	T0 L1 Q1	QP 28	Y 35.5626	U 42.3688	V 44.1222	34016 bit
	4: PK	T0 L1 Q2	QP 26	Y 36.8504	U 42.9543	V 44.6312	32216 bit
	4: PK	T0 L2 Q0	QP 33	Y 33.9804	U 41.5197	V 43.9867	100712 bit
	4: PK	T0 L2 Q1	QP 30	Y 34.8098	U 42.4008	V 44.6326	55912 bit
	4: PK	T0 L2 Q2	QP 28	Y 36.1474	U 42.8176	V 44.9110	94976 bit
AU	2: P	T1 L0 Q0	QP 35	Y 32.6892	U 41.4081	V 42.3910	1120 bit
	2: P	T1 L0 Q1	QP 29	Y 35.9567	U 42.7202	V 44.0688	1312 bit
	2: B	T1 L1 Q0	QP 34	Y 33.3360	U 41.6372	V 43.3606	3584 bit
	2: B	T1 L1 Q1	QP 31	Y 35.4880	U 42.2300	V 44.0442	2488 bit
	2: B	T1 L1 Q2	QP 29	Y 36.6595	U 43.0054	V 44.6095	1584 bit
	2: B	T1 L2 Q0	QP 36	Y 33.5405	U 41.2278	V 43.6894	16032 bit
	2: B	T1 L2 Q1	QP 33	Y 34.3797	U 42.0592	V 44.3992	2312 bit
	2: B	T1 L2 Q2	QP 31	Y 35.0945	U 42.4709	V 44.6759	4240 bit
AU	1: P	T2 L0 Q0	QP 36	Y 32.5340	U 41.3627	V 42.3284	712 bit
	1: P	T2 L0 Q1	QP 30	Y 35.4730	U 42.6550	V 43.9291	752 bit
	1: B	T2 L1 Q0	QP 35	Y 33.1174	U 41.5773	V 43.2632	2184 bit
	1: B	T2 L1 Q1	QP 32	Y 35.0868	U 42.0779	V 43.9476	1568 bit
	1: B	T2 L1 Q2	QP 30	Y 36.2183	U 42.8421	V 44.4220	1424 bit
	1: B	T2 L2 Q0	QP 37	Y 33.1865	U 41.2190	V 43.6951	13264 bit
	1: B	T2 L2 Q1	QP 34	Y 34.0538	U 41.9707	V 44.4058	2192 bit
	1: B	T2 L2 Q2	QP 32	Y 34.7445	U 42.3141	V 44.6854	3312 bit
AU	3: P	T2 L0 Q0	QP 36	Y 32.6137	U 41.4106	V 42.4509	736 bit
	3: P	T2 L0 Q1	QP 30	Y 35.4018	U 42.7481	V 44.1530	744 bit
	3: B	T2 L1 Q0	QP 35	Y 33.0925	U 41.6600	V 43.5079	2096 bit
	3: B	T2 L1 Q1	QP 32	Y 34.9586	U 42.1663	V 44.0868	1496 bit
	3: B	T2 L1 Q2	QP 30	Y 36.0753	U 42.8356	V 44.5813	1640 bit
	3: B	T2 L2 Q0	QP 37	Y 33.5421	U 41.2850	V 43.7491	10432 bit
	3: B	T2 L2 Q1	QP 34	Y 34.2565	U 41.9548	V 44.3916	1968 bit
	3: B	T2 L2 Q2	QP 32	Y 34.9790	U 42.3036	V 44.6300	2544 bit
AU	8: IK	T0 L0 Q0	QP 32	Y 33.1687	U 41.2749	V 42.3331	17872 bit
	8: IK	T0 L0 Q1	QP 26	Y 37.1930	U 42.7684	V 43.9254	17464 bit
	8: IK	T0 L1 Q0	QP 31	Y 33.7116	U 41.5996	V 43.2650	62976 bit
	8: IK	T0 L1 Q1	QP 28	Y 35.6974	U 42.1809	V 43.7919	45528 bit
	8: IK	T0 L1 Q2	QP 26	Y 36.9837	U 42.8044	V 44.3757	39832 bit
	8: IK	T0 L2 Q0	QP 33	Y 33.2699	U 40.8847	V 43.3576	158704 bit
	8: IK	T0 L2 Q1	QP 30	Y 34.9482	U 42.0252	V 44.0975	117704 bit
	8: IK	T0 L2 Q2	QP 28	Y 36.2307	U 42.4482	V 44.3729	117752 bit
AU	6: P	T1 L0 Q0	QP 35	Y 32.5874	U 41.3357	V 42.3250	1192 bit
	6: P	T1 L0 Q1	QP 29	Y 35.8418	U 42.7216	V 44.1693	1120 bit
	6: B	T1 L1 Q0	QP 34	Y 33.7328	U 42.0057	V 43.7880	3224 bit
	6: B	T1 L1 Q1	QP 31	Y 35.7485	U 42.5931	V 44.3565	2240 bit
	6: B	T1 L1 Q2	QP 29	Y 36.8818	U 43.2059	V 44.9576	1520 bit
	6: B	T1 L2 Q0	QP 36	Y 33.5546	U 41.5148	V 44.0752	15064 bit
	6: B	T1 L2 Q1	QP 33	Y 34.2516	U 42.3538	V 44.6459	2544 bit
	6: B	T1 L2 Q2	QP 31	Y 34.8663	U 42.6856	V 44.8843	4568 bit
AU	5: P	T2 L0 Q0	QP 36	Y 32.4250	U 41.4947	V 42.3766	776 bit
	5: P	T2 L0 Q1	QP 30	Y 35.3763	U 42.7826	V 44.0675	920 bit
	5: B	T2 L1 Q0	QP 35	Y 33.2465	U 41.8401	V 43.6599	2072 bit
	5: B	T2 L1 Q1	QP 32	Y 35.0731	U 42.4424	V 44.2653	1736 bit
	5: B	T2 L1 Q2	QP 30	Y 36.1158	U 42.9995	V 44.8024	1416 bit
	5: B	T2 L2 Q0	QP 37	Y 33.4395	U 41.3664	V 44.0332	12376 bit
	5: B	T2 L2 Q1	QP 34	Y 34.0127	U 42.1591	V 44.6121	2272 bit
	5: B	T2 L2 Q2	QP 32	Y 34.5685	U 42.4565	V 44.8221	3552 bit
AU	7: P	T2 L0 Q0	QP 36	Y 32.5675	U 41.3523	V 42.4416	768 bit
	7: P	T2 L0 Q1	QP 30	Y 35.4404	U 42.6993	V 44.1539	768 bit
	7: B	T2 L1 Q0	QP 35	Y 33.2437	U 41.7251	V 43.5913	2648 bit
	7: B	T2 L1 Q1	QP 32	Y 35.1144	U 42.2419	V 44.0277	1528 bit

```
7: B    T2 L1 Q2   QP 30   Y 36.1693  U 42.8345  V 44.5852      1304 bit
7: B    T2 L2 Q0   QP 37   Y 33.0013  U 41.1399  V 43.7164     13200 bit
7: B    T2 L2 Q1   QP 34   Y 33.8069  U 41.9455  V 44.3042      1984 bit
7: B    T2 L2 Q2   QP 32   Y 34.3766  U 42.2269  V 44.5312      2976 bit
```

2. *layerTrace.txt*

   The frame order is the same as the sdout.txt frame order. Details as per GOP of 1. Note reduction in transmission cost in comparison to a GOP value of 1, but note the increase in the cost for frame 4 with respect to a GOP value of eight, option one, as this is a P-frame in this encoding.

Start-Pos.	Length	LId	TId	QId	Packet-Type	Discardable	Truncatable
0x000002d4	18	0	0	0	SliceData	No	No
0x000002e6	2277	0	0	0	SliceData	No	No
0x00000bcb	2169	0	0	1	SliceData	Yes	No
0x00001444	7635	1	0	0	SliceData	No	No
0x00003217	5586	1	0	1	SliceData	Yes	No
0x000047e9	4673	1	0	2	SliceData	Yes	No
0x00005a2a	17006	2	0	0	SliceData	No	No
0x00009c98	13798	2	0	1	SliceData	Yes	No
0x0000d27e	13715	2	0	2	SliceData	Yes	No
0x00010811	18	0	0	0	SliceData	No	No
0x00010823	453	0	0	0	SliceData	No	No
0x000109e8	2138	0	0	1	SliceData	Yes	No
0x00011242	2145	1	0	0	SliceData	No	No
0x00011aa3	4252	1	0	1	SliceData	Yes	No
0x00012b3f	4027	1	0	2	SliceData	Yes	No
0x00013afa	12589	2	0	0	SliceData	No	No
0x00016c27	6989	2	0	1	SliceData	Yes	No
0x00018774	11872	2	0	2	SliceData	Yes	No
0x0001b5d4	18	0	1	0	SliceData	Yes	No
0x0001b5e6	131	0	1	0	SliceData	Yes	No
0x0001b669	164	0	1	1	SliceData	Yes	No
0x0001b70d	448	1	1	0	SliceData	Yes	No
0x0001b8cd	311	1	1	1	SliceData	Yes	No
0x0001ba04	198	1	1	2	SliceData	Yes	No
0x0001baca	2004	2	1	0	SliceData	Yes	No
0x0001c29e	289	2	1	1	SliceData	Yes	No
0x0001c3bf	530	2	1	2	SliceData	Yes	No
0x0001c5d1	17	0	2	0	SliceData	Yes	No
0x0001c5e2	80	0	2	0	SliceData	Yes	No
0x0001c632	94	0	2	1	SliceData	Yes	No
0x0001c690	273	1	2	0	SliceData	Yes	No
0x0001c7a1	196	1	2	1	SliceData	Yes	No
0x0001c865	178	1	2	2	SliceData	Yes	No
0x0001c917	1658	2	2	0	SliceData	Yes	No
0x0001cf91	274	2	2	1	SliceData	Yes	No
0x0001d0a3	414	2	2	2	SliceData	Yes	No
0x0001d241	17	0	2	0	SliceData	Yes	No
0x0001d252	83	0	2	0	SliceData	Yes	No
0x0001d2a5	93	0	2	1	SliceData	Yes	No
0x0001d302	262	1	2	0	SliceData	Yes	No
0x0001d408	187	1	2	1	SliceData	Yes	No
0x0001d4c3	205	1	2	2	SliceData	Yes	No
0x0001d590	1304	2	2	0	SliceData	Yes	No
0x0001daa8	246	2	2	1	SliceData	Yes	No
0x0001db9e	318	2	2	2	SliceData	Yes	No
0x0001dcdc	18	0	0	0	SliceData	No	No
0x0001dcee	2225	0	0	0	SliceData	No	No
0x0001e59f	2183	0	0	1	SliceData	Yes	No
0x0001ee26	7872	1	0	0	SliceData	No	No
0x00020ce6	5691	1	0	1	SliceData	Yes	No
0x00022321	4979	1	0	2	SliceData	Yes	No

0x00023694	19838	2	0	0	SliceData	No	No
0x00028412	14713	2	0	1	SliceData	Yes	No
0x0002bd8b	14719	2	0	2	SliceData	Yes	No
0x0002f70a	18	0	1	0	SliceData	Yes	No
0x0002f71c	140	0	1	0	SliceData	Yes	No
0x0002f7a8	140	0	1	1	SliceData	Yes	No
0x0002f834	403	1	1	0	SliceData	Yes	No
0x0002f9c7	280	1	1	1	SliceData	Yes	No
0x0002fadf	190	1	1	2	SliceData	Yes	No
0x0002fb9d	1883	2	1	0	SliceData	Yes	No
0x000302f8	318	2	1	1	SliceData	Yes	No
0x00030436	571	2	1	2	SliceData	Yes	No
0x00030671	17	0	2	0	SliceData	Yes	No
0x00030682	88	0	2	0	SliceData	Yes	No
0x000306da	115	0	2	1	SliceData	Yes	No
0x0003074d	259	1	2	0	SliceData	Yes	No
0x00030850	217	1	2	1	SliceData	Yes	No
0x00030929	177	1	2	2	SliceData	Yes	No
0x000309da	1547	2	2	0	SliceData	Yes	No
0x00030fe5	284	2	2	1	SliceData	Yes	No
0x00031101	444	2	2	2	SliceData	Yes	No

3. *layerRate.txt*

Note that now due to the increase in GOP value, the frame rate per layer has increased. Each layer has 3 frame rates, note that the frame rate per layer is based on the FrameRateOut value in the layer configuration file. But if this value is the same as the FrameRate in the main configuration file, the number of frame rates per layer is equal to $n+1$, where n is determined by $2^n = \text{GOPsize}$. In this example we have a GOPsize of 4, which equates to 2^2, which defines the number of frame rates per layer to be $2+1=3$.

Layer	Resolution	Framerate	Bitrate	MinBitrate	DTQ
0	176x144	7.5000	79.40	79.40	(0,0,0)
1	176x144	15.0000	87.90	87.90	(0,1,0)
2	176x144	30.0000	98.70	98.70	(0,2,0)
3	176x144	7.5000	203.50		(0,0,1)
4	176x144	15.0000	220.70		(0,1,1)
5	176x144	30.0000	242.80		(0,2,1)
6	352x288	7.5000	461.90	337.80	(1,0,0)
7	352x288	15.0000	501.50	368.70	(1,1,0)
8	352x288	30.0000	552.20	408.10	(1,2,0)
9	352x288	7.5000	744.50		(1,0,1)
10	352x288	15.0000	799.70		(1,1,1)
11	352x288	30.0000	872.00		(1,2,1)
12	352x288	7.5000	990.30		(1,0,2)
13	352x288	15.0000	1057.40		(1,1,2)
14	352x288	30.0000	1147.90		(1,2,2)
15	704x576	7.5000	1798.00	1145.50	(2,0,0)
16	704x576	15.0000	1971.00	1282.30	(2,1,0)
17	704x576	30.0000	2225.00	1485.20	(2,2,0)
18	704x576	7.5000	2368.00		(2,0,1)
19	704x576	15.0000	2559.00		(2,1,1)
20	704x576	30.0000	2844.00		(2,2,1)
21	704x576	7.5000	3097.00		(2,0,2)
22	704x576	15.0000	3321.00		(2,1,2)
23	704x576	30.0000	3650.00		(2,2,2)

SVC transmission cost per GOP is 131,598 bytes (sum over all layer for all frames per GOP). While using Equation 4.2 from Chapter 4 we can determine the MDC transmission cost of one description per frame, *e.g.* 28,020 bytes. We multiple this by the layer required, layer 8, thus providing a total MDC transmission cost

Table A.1: Transmission byte-cost for SVC and MDC for each of the GOP options. Values provided per GOP and for the first 8 frames, thus illustrating a comparison between the cost of a GOP of 8 and the first eight frames with a GOP of 1.

	GOP1	GOP8 opt1	GOP8 opt2
SVC per GOP	66,877	91,069	131,598
SVC 8 frames	535,481	91,069	131,598
MDC per GOP	118,184	161,680	224,160
MDC 8 frames	947,968	161,680	224,160

of 224,160 bytes.

A.5 Conclusion

Items to note include:

i. Both transmission cost per GOP and overall transmission cost per clip reduces as GOP increases.

ii. A GOP size greater than one mandates non-sequential encoding and transmission.

iii. GOP options 1 and 2 created two different frame hierarchies for the higher layers, *IBBBBBBBI* and *IBBBPBBBI* respectively. GOP option 1 contains no P-frames. Thus it is our opinion that GOP option 2 is the more conventional encoding to use for a GOP of 8 and this is the encoding we have used to simulate our evaluation results. Thus the GOP value used is twice the size of the GOPsize and equal to the IntraPeriod used in the main JSVM configuration file.

iv. If the frameRateOut per layer is equal to the FrameRate then all layers provide the same frame rates. Thus each frame contains the same number of layers. If a lower layer contains a lower frameRateOut value, then some frames will not contain data for that specific layer.

v. Table A.1 illustrates transmission cost relative to the GOP options illustrated.

vi. GOP of 1 has the highest transmission cost over 8 frames and GOP of 8 option 1 has the lowest. The increase in GOP of 8 option 2 relative to the cost over GOP of 8 option 1 is the P-frame as well as minor increases in cost relative to the B-frames. Remember P-frames are only one direction and only compensate from previous I-frames, while the 4th frame B-frame, in GOP of 1 option 1, can use both the first and 9th I-frame with which to reduce encoding cost.

vii. We can not make the assumption that the MDC achievable quality for the P-frame would be the same for the B-frame. Regardless of the reduction in packet number, the B-frame is now dependent not only on the achievable quality, viewable layer, of the first I-frame, but it also dependent on the achievable quality of the ninth

I-frame. Thus increasing the possibility of cascading loss over a number of frames, or mandating low quality over the GOP.

viii. Please note that to fully understand this final comment, Section 3.3 should be read first: Finally we can make the assumption that the achievable quality using SD from Section 3.3.2, would be the same for both GOP of 8 options, irrespective of inter-dependency, assuming the same loss rate per frame. Remember SD packetises to mandate packet equality, and it is the number of lost packets that mandate viewable quality and not frame inter-dependence. The same layer value will be defined for all frames per GOP.

Appendix B

Subjective Testing Instructions

In this Appendix, we provide information on the questionnaire used and briefing notes presented at the subjective testing. Prior to the commencement of the subjective testing, Figure B.1 was shown, via a large screen projector, to all the test subjects and this information was used to define the requirements of the subjects, with respect to grading and ranking of the test clips. The grading sheet mentioned in these instructions is presented in Figure B.3. During the duration of the subjective testing, for each clip iteration, Figure B.2 was shown on screen, so as to permit the subjects ease of access to the grading metrics.

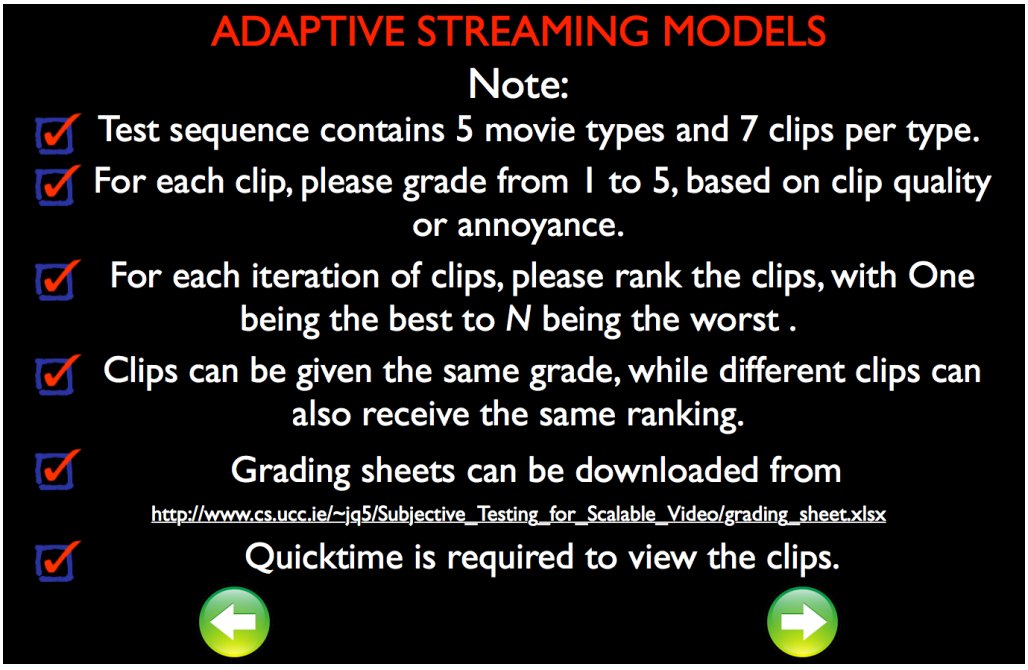


Figure B.1: Briefing details presented prior to commencement of subjective testing

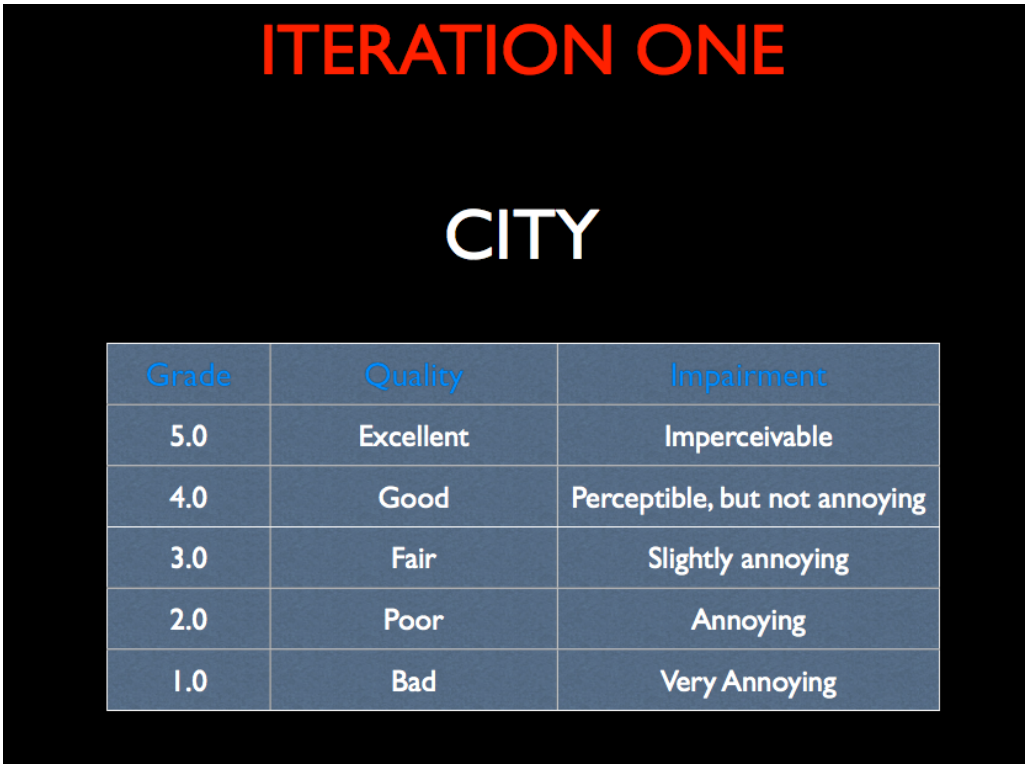


Figure B.2: Overview of grading schemes utilised

Subjective Testing of Adaptive Streaming Models
Iteration 1

NAME: _____

Clip Rating:
Each Clip can be Rated from 1 to 5, with 1 being the lowest rating and 5 being the highest rating.
Please consider, achievable quality, stream consistency and least annoyance when rating.

Clip Ranking:
Once all Clips have been viewed, please Rank the clips in order of preference.
With 1 being the best clip, or least annoying, and 7 the worst clip, or most annoying.

Clip Number	Clip Rating	Clip Ranking
Clip One		
Clip Two		
Clip Three		
Clip Four		
Clip Five		
Clip Six		
Clip Seven		

Notes:

==========

Please do not write below this line

SIGNATURE: _____

SIGNATURE OF PROJECT LEADER: _____

Figure B.3: Questionnaire sheet provided for each iteration of the subjective testing

Appendix C

Streaming Classes - Class Packetisation extended Evaluation Results

In this Appendix, we extend the example in Appendix 5.1.5.1 and provide examples of defining the packet byte-size allocation based on fixed values of 1,440 bytes, 1,000 bytes and 500 bytes for each class, rather than the defined section size of the underlying description model. Defining the packet byte-size allocation based on fixed byte sizes is consistent with the packet byte-size allocation required for SVC, thus we shall view these evaluation results as SVC-SC (SVC using a Streaming Class model). These values define that all packets per GOP, over all classes, are the same byte size, but not of equal importance (lower class packets are still of higher priority). We also defined a model where a 1,440 byte threshold is allocated to the highest class, a 1,000 byte threshold to the middle class and a 500 byte threshold to the lowest class, thus illustrating bytes sizes relative to the underlying byte cost of the respective classes. While a final option determines the number of packets for the highest layer using a 1,440 byte allocation, and mandated that all lower classes utilise the same number of packets, as a result forcing an optimal packet threshold for the lower classes based on the number of packets for the highest class. Finally we illustrate results for these packet byte-size allocation models over one loss rate of 10% and over four GOP values, *e.g.* GOP-1, GOP-8, GOP-16 and GOP-32.

The focus of this appendix is to investigate the error resilience of SC under different SC structures and packetisation options. Of the five Streaming class design principles, as presented in Section 5.1, the allocation rates of the principles of *Error Resiliency* and *Class Packetisation and Granularity of Packet Data Byte-size* are the subject of this Appendix. For the remainder of the design principles, in our evaluated results we mandate a *Class Packet Loss Rate* of 10%, a *Layer Allocation and Hierarchy*

Table C.1: $LR_{\max}$ and $LR_{C_i}^{\max}$ SVC-SC packet byte-size allocation per layer, and per class, for 10% packet loss rates

Layer	$LR_{\max}$	$LR_{C_i}^{\max}$	Class
L8	14%	17%	Class 3
L7	18%	22%	
L6	23%	27%	
L5	14%	17%	Class 2
L4	18%	22%	
L3	23%	27%	
L2	14%	17%	Class 1
BL	18%	22%	

Table C.2: Based on initial loss rate, maximum $LR_{\max}$ and $LR_{C_i}^{\max}$ SVC-SC allocation per layer, and per class, for a 10% packet loss rate

	SVC-SC 10%		
Layer	$LR_{\max}$	$LR_{C_i}^{\max}$	Class
L8	14%	17%	Class 3
L7	18%	22%	
L6	23%	27%	
L5	28%	33%	Class 2
L4	34%	39%	
L3	40%	46%	
L2	47%	53%	Class 1
BL	54%	61%	

of three classes based on an eight layer SVC encoding, with class composition based on resolution and a *Streaming Class Structure* of Independent Class Composition (ICC) is used.

We begin by defining the values for the FEC allocation options $LR_{\max}$ ($\lceil \mu + \sqrt{\mu} \rceil$) and $LR_{C_i}^{\max}$ (see equation 5.2 in Section 5.1.3) based on the 10% loss rate, as illustrated in Table C.1. Table C.1 also illustrates the *Layer Allocation and Hierarchy* of the three classes. In this example we extend the $LR_{\max}$ and $LR_{C_i}^{\max}$ values over all classes, thus increasing the overall transmission cost of SVC-SC. We use the same initial $LR_{\max}$ and $LR_{C_i}^{\max}$ allocation for all classes, *e.g.* 14% and 17% respectively. Note how the level of FEC is based on the index of the layer in the underlying class, with higher layers receiving lower levels of FEC and lower layers receive allocation rates based on the preceding higher layer plus the standard deviation of the previous higher layer, i.e. the FEC allocation rates of layer 7 are based on the FEC rates of layer 8 plus the square root of the FEC rates of layer 8. Using this allocation rate of FEC, it can be seen that while C1 is the highest priority class it contains the lowest levels of FEC, due to only two layers being allocated to this class.

In the next option, illustrated in Table C.2, we apply the initial $LR_{\max}$ and $LR_{C_i}^{\max}$ allocation for all classes, *e.g.* 14% and 17% respectively, to the highest layer only, layer

Table C.3: Transmission costs per layer for SVC, MDC, ALD, and per class for both FEC allocation options for SVC-SC, for a packet loss rate of 10%. A GOP value of one and the FEC allocation rates as per Table C.1 are implemented for these results

	Existing				SVC-SC 10%	
Layer	SVC	MDC	ALD	Class	$LR_{\max}$	$LR_{C_i}^{\max}$
L8	9,483,746	19,334,064	12,075,882	Class 3	11,176,159	11,519,753
L7	7,419,782	16,917,306	11,213,319			
L6	5,640,092	14,500,548	10,350,756			
L5	3,931,226	12,083,790	9,488,193	Class 2	4,619,070	4,760,894
L4	2,993,946	9,667,032	8,625,630			
L3	2,044,662	7,250,274	7,763,067			
L2	1,232,574	4,833,516	6,900,504	Class 1	1,430,363	1,473,636
BL	617,526	2,416,758	6,037,941			

8. We then view the stream based on layer dependency rather than class structure, and extend increasing levels of $LR_{\max}$ and $LR_{C_i}^{\max}$ over all layers. This provides a direct correlation between the level of FEC and the priority of the underlying layer. In this option, the FEC rates of C3 are unchanged, while the rates of C1 and C2 have increased. While the options shown in Table C.1 and Table C.2 can be viewed as the minimum (best-case) and maximum (worst-case) FEC allocation rates for our schemes, table C.2 can also be viewed as a outline of the possible $LR_{\max}$ and $LR_{C_i}^{\max}$ rates that can be used. An example of this would be to increase the allocation rate for C1 from 17% and 22% to 22% and 27% for $LR_{\max}$ and from 14% and 18% to 18% and 23% for $LR_{C_i}^{\max}$. Thus providing a means of adapting the FEC allocation even further so as to provide increased adaptation to loss or to degradation in viewable quality.

C.1 Varying Packet Size for a GOP of One

Table C.3 illustrates the GOP1 transmission cost for SVC, MDC, ALD (STF=6) and SVC-SC for the FEC allocation rates $LR_{\max}$ and $LR_{C_i}^{\max}$ as per Table C.1. Figure C.1 plots the transmission cost for the SVC-SC classes C1, C2 and C3 for $LR_{\max}$ and $LR_{C_i}^{\max}$ with packet loss rates from 0% to 10% based on the FEC rates of Table C.1. Note the slight increase in transmission cost for $LR_{C_i}^{\max}$ over $LR_{\max}$. Note that the transmission cost with a 0% packet loss rates, denotes the standard SVC transmission cost of each of the highest layers per class.

From this point forward, we are going to use FEC to denote $LR_{\max}$, as this is our basic level of FEC and we are going to use $FEC_{\max}$ to denote $LR_{C_i}^{\max}$, as this is our maximum level of FEC. This is purely to make the legends and text in the plots clearer and more legible. This would change the notation in Figure C.1 from $LR_{\max}$ and $LR_{C_i}^{\max}$ to the notation FEC and $FEC_{\max}$ as used in Figure C.2. Note the plots lines in Figure C.1 and Figure C.2 are identical as only the notation has changed.

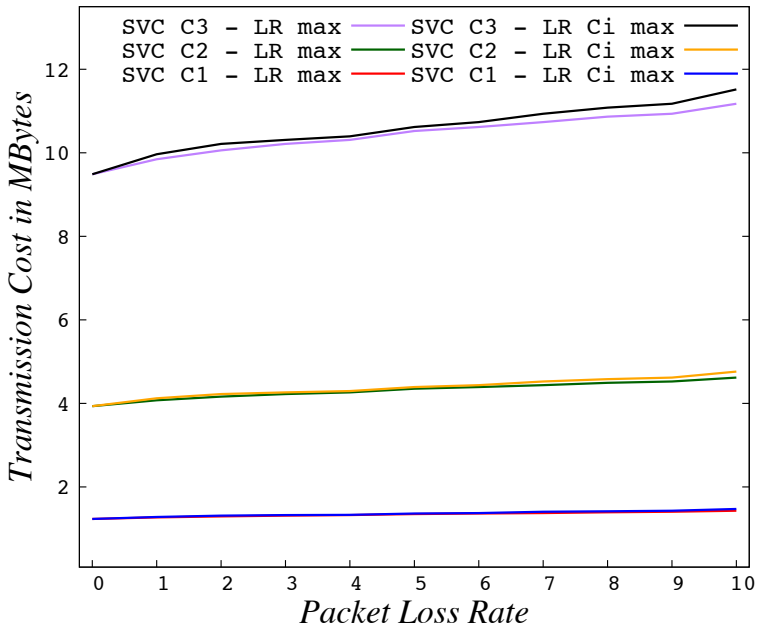


Figure C.1: Transmission cost for the SVC-SC classes C1, C2 and C3 for both $LR_{\max}$ and $LR_{C_i}^{\max}$ with packet loss rates from 0% to 10%

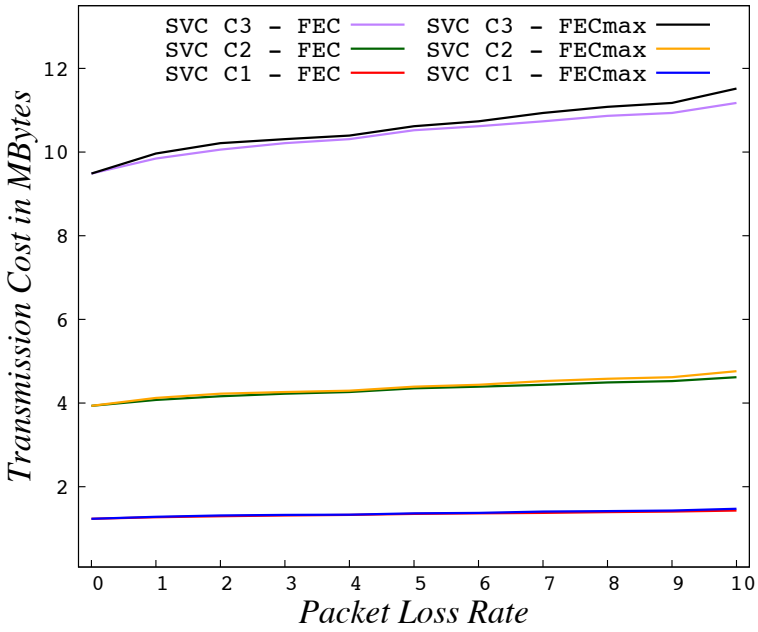


Figure C.2: Transmission cost for the SVC-SC classes C1, C2 and C3 for both FEC and $FEC_{\max}$ with packet loss rates from 0% to 10%

Evaluation results are provided for both FEC and $FEC_{\max}$ and for five different packet byte-size allocation schemes. Where each of the classes is defined based on the same packet byte-size allocation, *e.g.* three different Packet threshold, $Packet_{thres}$, are simulated: 1,440 bytes, 1,000 bytes and 500 bytes. We also evaluate where each of the classes is defined based on a different $Packet_{thres}$, *e.g.* 500 bytes for C1, 1,000 bytes for C2 and 1,440 bytes for C3, and we call this *multi*, *e.g.* multiple different class byte-allocations per stream. Additionally, we consider another packetisation scheme in which we evaluate based on the number of packets transmitted. Rather than define a different $Packet_{thres}$ for each class, we shall define the same number of packets for each class, based on the number of packets defined for the highest class. Thus the byte-allocation per packet is dependent on the bitrate per class. We call this scheme '$Packet_{equal}$'. '$Packet_{equal}$' is similar to *multi* in that different $Packet_{thres}$ are defined for each of the classes, but the $Packet_{thres}$ of '$Packet_{equal}$' is dynamically allocated dependent on the number of packets created for the highest class, as apposed to the static $Packet_{thres}$ allocation of *multi*. For extremely low quality streaming, situations may occur where '$Packet_{equal}$' may mandate that the byte allocation of the packets of the lowest classes contain only a small number of bytes. This would occur when the quality of the encoded resolutions and their underlying bitrates are vastly different, especially with reference to the highest and lowest quality levels, *e.g.* Full HD (1920 x 1080) at layer 8 and QCIF (176 x 144) at the base layer, but this allocation of resolution during encoding would rarely, if ever, be selected.

Table C.4 shows for each of the packet byte-size allocation schemes: number of packets sent per class, and per stream, for a defined loss rate, 10%, the header cost of these packets, the packets lost per class, the loss rate determined per class and the maximum per frame loss rate. Note that the percentage values are rounded up.

It can be seen that there is little variation in the overall loss rates per class, relative to the loss rate experienced, thus illustrating the the loss rates are equally spread overall classes. 500 byte packet threshold has the largest number of packets transmitted followed by $Packet_{equal}$ and both of these options have the lowest maximum loss rate per frame for all five schemes. Giving us our first indication that lower byte cost, or a higher number of packets, can spread the effects of network loss and improve viewable quality.

For Figure C.3 it can be seen that the $Packet_{thres}$ of '500', 'multi' and $Packet_{equal}$, called 'equalP' in the plot, provide the highest number of viewable frames for layer 8, with the lowest number of non decodable frames (un-viewable frames). But the viewable quality across all layers and models is overall very bad. Figure C.4 does prove that the minor increase in FEC provided by $FEC_{\max}$ is very beneficial to the overall quality, with marked reductions in all the lower layers and noticeable increases in the highest layer, but the noticeable levels of non decodable frames (un-viewable frames),

Table C.4: FEC and $FEC_{\max}$ models for all four $Packet_{thres}$, as well as $Packet_{equal}$. Values are illustrated for packets transmitted, additional packet header cost (60 bytes per packet for header information) and individual packets sent, packets received, loss rate per class and maximum loss rate per frame for SVC-SC, for an overall packet loss rate of 10% and a GOP of one.

	SVC-SC 10% Different packet loss rates per class, with a GOP of 1									
	FEC					$FEC_{\max}$				
Byte size	1,440	1,000	500	Multi	$Packet_{equal}$	1,440	1,000	500	Multi	$Packet_{equal}$
# Packets	8,215	11,594	22,776	11,034	14,082	8,469	11,969	23,487	11,379	14,532
Header Cost	492,900	695,640	1,366,560	662,040	844,920	508,140	718,140	1,409,220	682,740	871920
Packets Sent										
C1	1,169	1,158	3,000	3,000	4,694	1,194	1,609	3,096	3,096	4,844
C2	2,352	3,340	6,516	3,340	4,694	2,431	3,439	6,725	3,439	4,844
C3	4,694	6,696	13,260	4,694	4,694	4,844	6,921	13,666	4,844	4,844
Packets Received										
C1	1,038	1,401	2,733	2,686	4,247	1,068	1,445	2,786	2,786	4,374
C2	2,101	3,019	5,816	3,011	4,204	2,196	3,107	6,082	3,084	4,338
C3	4,256	6,001	11,943	4,225	4,224	4,360	6,211	12,271	4,361	4,365
Loss Rate (LR)										
C1	11%	10%	9%	10%	10%	11%	10%	10%	10%	10%
C2	11%	10%	11%	10%	10%	10%	10%	10%	10%	10%
C3	9%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Max per Frame LR										
C1	100%	80%	44%	50%	45%	75%	75%	50%	50%	33%
C2	50%	40%	30%	56%	36%	44%	44%	30%	42%	46%
C3	33%	38%	22%	46%	42%	50%	35%	25%	42%	43%

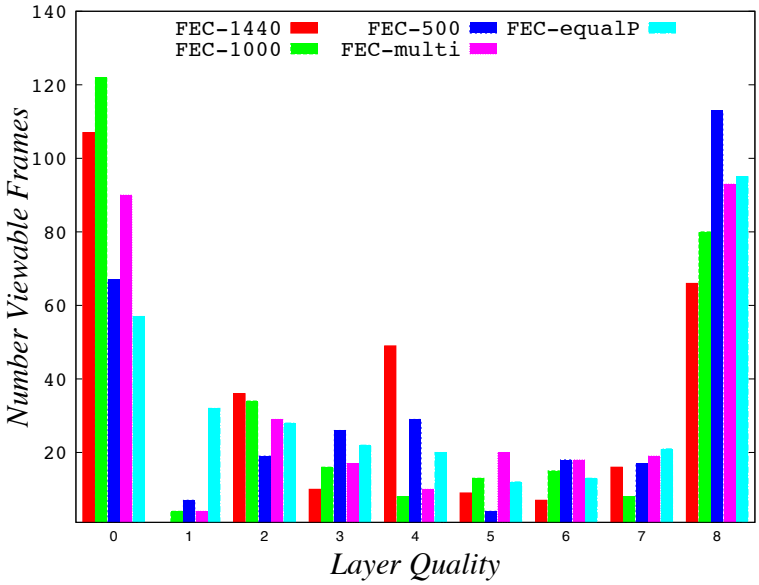


Figure C.3: Example of the number of viewable layers for SVC-SC, for each of the $FECPacket_{thres}$, at a packet loss rate of 10%

approximately 20% for $Packet_{thres}$ of '500' and 'multi', and 13% for 'equalP' is still too high.

One item to note across both FEC models is that the consistency of quality for layers 3 and 6 (lowest layers for C2 and C3) are very similar, with this consistency also evident in layers 4 and 7 (mid layers for C2 and C3) for $FEC_{\max}$ only. We believe this illustrates that lower levels of FEC will provide consistent quality for different class streaming models, but an FEC level higher than $FEC_{\max}$ will be required.

Figure C.5 and Figure C.6 illustrate a two second example of variation in viewable

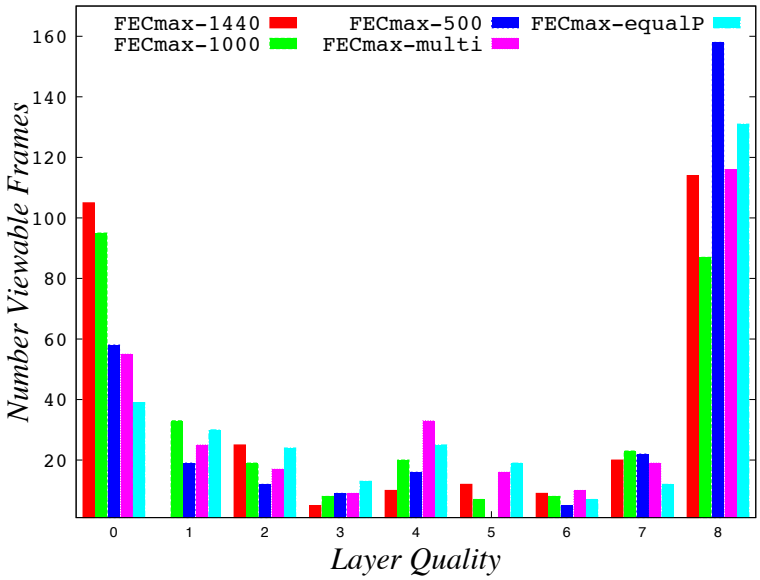


Figure C.4: Example of the number of viewable layers for SVC-SC, for each of the $FEC_{\max} Packet_{thres}$, at a packet loss rate of 10%

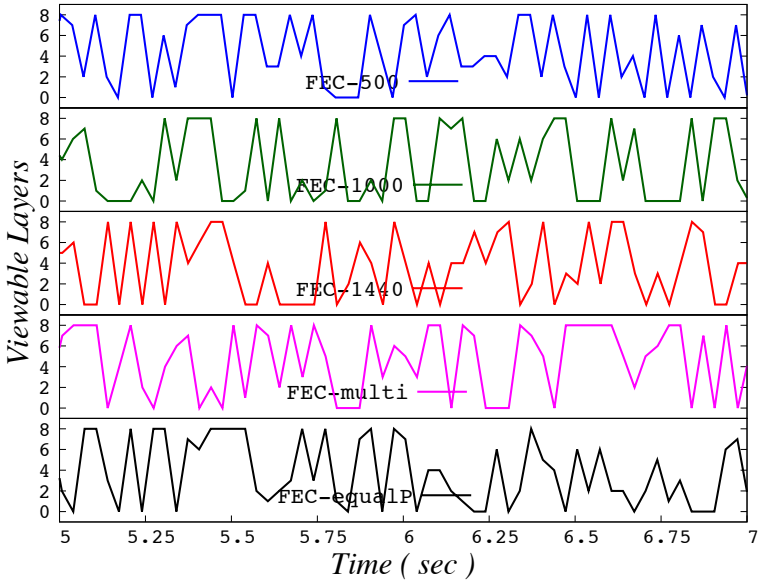


Figure C.5: A two second example of variation in viewable quality for all FEC models, at a packet loss rate of 10%

quality for all packet byte-size allocation schemes with reference to Figure C.3 and Figure C.4 respectively. Note that all schemes fail to provide consistency of quality. Thus it can be determined that for this GOP value and these FEC levels, consistency of quality is unavailable and degradation of quality will occur. All of the models show wide variation in viewable quality over time, while '500' for $FEC_{\max}$ does show some increase in quality but this is limited to only a subset of the viewable frames, and would only provide limited increases in perceived quality.

Figure C.7 and Figure C.8 illustrate an increase in the $FEC/FEC_{\max}$ allocation

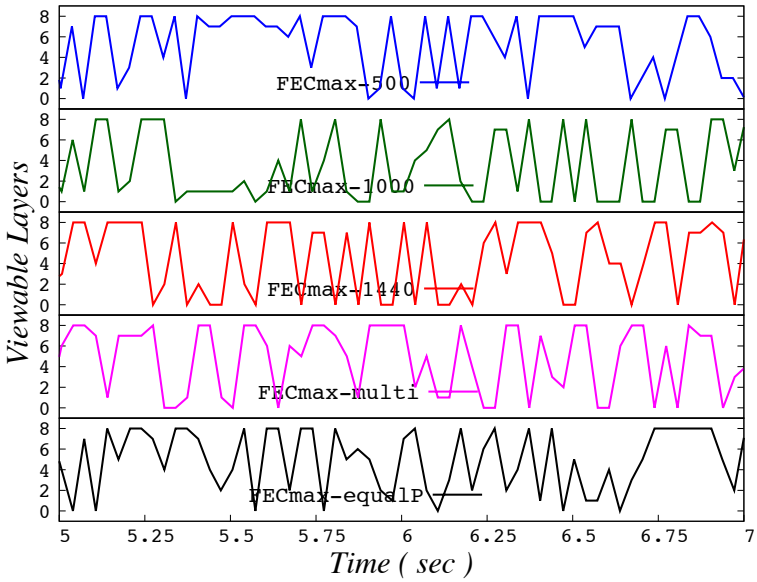


Figure C.6: A two second example of variation in viewable quality for all $FEC_{\max}$ models, at a packet loss rate of 10%

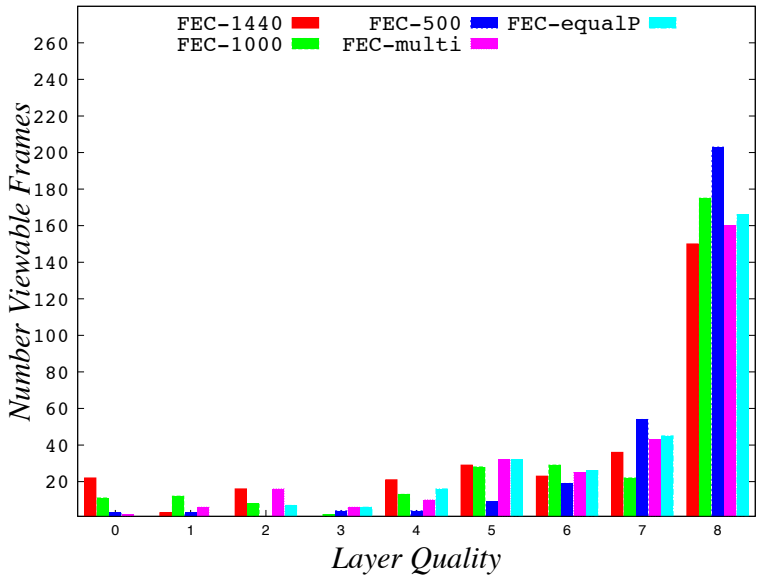


Figure C.7: Example of the number of viewable layers for SVC-SC, for each of the $FECPacket_{thres}$, at a packet loss rate of 10%. This image illustrates an increase in the FEC for C1 and C2 based on the worst case scenario, as shown in Table C.2.

to C1 and to C2. Allocation rates for C3 remain unchanged. The $FEC/FEC_{\max}$ allocation are based on the direct correlation scenario, as shown in Table C.2. Note the large decrease in lower layer viewing and increase for most schemes in the highest class, with reference to Figure C.3 and Figure C.4 respectively, with a $Packet_{thres}$ of 500 bytes for all three classes using $FEC_{\max}$ allocation rates providing near complete C3 viewing.

Finally Table C.5 presents the cumulative transmission cost of the three classes

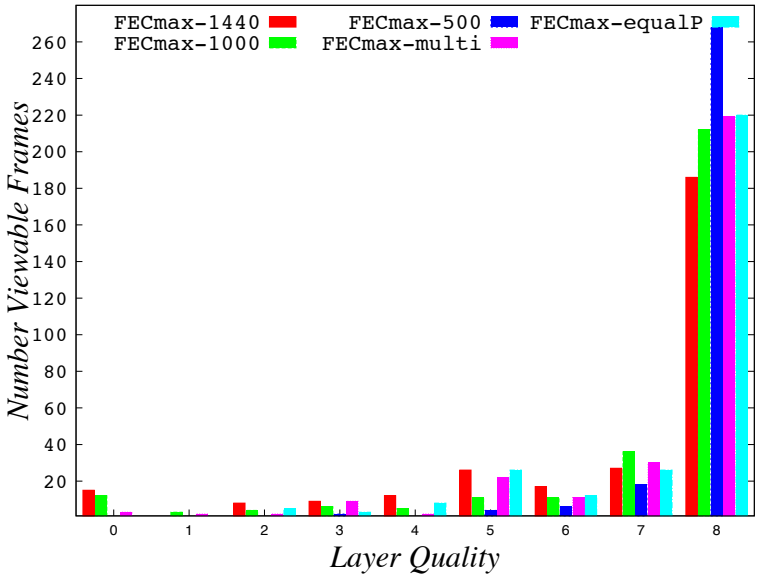


Figure C.8: Example of the number of viewable layers for SVC-SC, for each of the $FEC_{\max}Packet_{thres}$, at a packet loss rate of 10%. This image illustrates an increase in the $FEC_{\max}$ for C1 and C2 based on the worst case scenario, as shown in Table C.2.

Table C.5: Cumulative transmission cost of the three classes based on the respective underlying FEC allocation rate, *e.g.* Table C.1 (best case) and Table C.2 (worst case)

		Table C.1		Table C.2	
		FEC	$FEC_{\max}$	FEC	$FEC_{\max}$
L8	Class 3	11,176,159	11,519,753	12,014,762	12,447,975
L7					
L6					
L5	Class 2	4,619,070	4,760,894	5,457,673	5,689,116
L4					
L3					
L2	Class 1	1,430,363	1,473,636	1,855,808	1,935,993
BL					

based on the respective underlying FEC allocation rate, *e.g.* Table C.1 (best case) and Table C.2 (worst case). We note that for $FEC_{\max}$, Table C.2 has increased the cumulative transmission cost of C1 by 32%, C2 by 19% and C3 by 8%. When we view these costs with respect to ALD from Table C.3 we note a reduction in C1 for $FEC_{\max}$ of 72%, in C2 of 40% but an increase of 3% in C3. While the transmission cost for $FEC_{\max}$ in C1 with respect to SVC is an increase of 57%, but when compared to the C1 cost for MDC (293%) and ALD (460%), this is a dramatic decrease in overall transmission cost for C1. Similar savings can be made in C2.

Table C.6: FEC and $FEC_{\max}$ models for all four $Packet_{thres}$, as well as $Packet_{equal}$. Values are illustrated for packets transmitted, additional packet header cost (60 bytes per packet) and individual packets sent, packets received, loss rate per class and maximum loss rate per frame for SVC-SC, for an overall packet loss rate of 10% and a GOP of eight.

	SVC-SC 10% Different packet loss rates per class, with a GOP of 8									
	FEC					$FEC_{\max}$				
Byte size	1,440	1,000	500	Multi	$Packet_{equal}$	1,440	1,000	500	Multi	$Packet_{equal}$
# C3 Trans Cost	4,679,425	4,679,944	4,681,985	4,679,896	4,680,918	4,824,226	4,824,655	4,826,978	4,824,952	4,825,324
# Packets	3,303	4,736	9,418	4,404	6,021	3,405	4,880	9,702	4,534	6,201
Header Cost	198,180	284,160	565,080	264,240	361,260	204,300	292,800	582,120	272,040	372,060
Packets Sent										
C1	392	561	1,103	1,103	2,007	404	576	1,132	1,132	2,067
C2	904	1,294	2,573	1,294	2,007	934	1,335	2,652	1,335	2,067
C3	2,007	2,881	5,742	2,007	2,007	2,067	2,969	5,918	2,067	2,067
Packets Received										
C1	345	495	984	975	1,791	364	511	1,020	1,016	1,858
C2	806	1,163	2,324	1,153	1,808	833	1,203	2,404	1,201	1,841
C3	1,790	2,559	5,153	1,821	1,812	1,836	2,664	5,285	1,847	1,876
Loss Rate (LR)										
C1	12%	12%	11%	12%	11%	10%	11%	10%	10%	10%
C2	11%	10%	10%	11%	10%	11%	10%	9%	10%	11%
C3	11%	11%	10%	9%	10%	11%	10%	11%	11%	09%
Max per Frame LR										
C1	50%	30%	28%	27%	22%	38%	31%	22%	22%	20%
C2	27%	18%	15%	20%	21%	22%	24%	18%	19%	18%
C3	21%	100%	15%	17%	22%	21%	18%	16%	25%	18%

C.2 Varying Packet Size for a GOP of Eight

Table C.6 is the GOP of eight equivalent of Table C.4. Note the decrease in max loss rate per frame, as well as the decrease in overall transmission cost, as GOP increases. The remainder of this section compares GOP8 plots and figures to the previously shown GOP1 plots and figures.

Figures C.9 to C.10 illustrate a direct comparison to Figures C.3 to C.4. Note how a higher GOP value provides overall higher quantities of the higher layers, but still with excessive numbers of non-decodable frames. This is primarily due to the cascading effect of SD packetisation, where a higher packet loss rate, bursty loss, in one GOP will affect all frames for that GOP, thus mandating low or no decodable quality. Also note how the relatively small increase provided by $FEC_{\max}$ can show vastly improved higher layer viewing numbers, especially noticeable in Figure C.10 for '500' and 'equalP'.

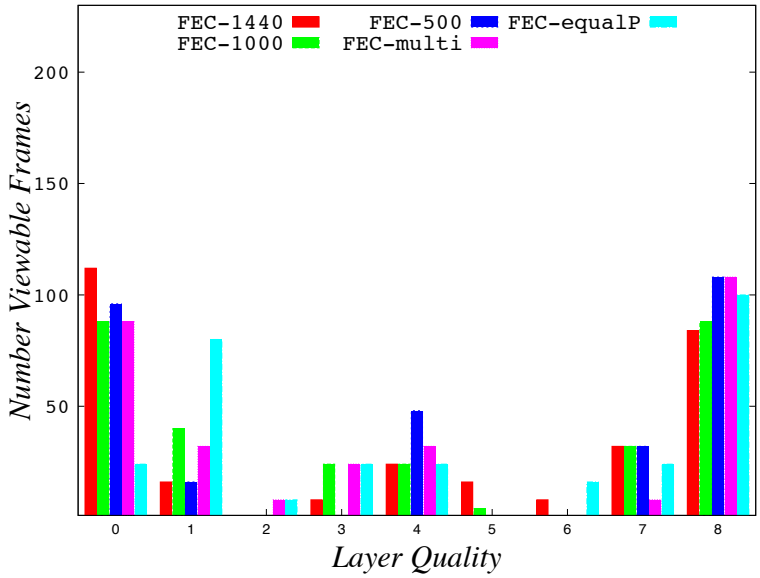


Figure C.9: Example of the number of viewable layers for SVC-SC, for each of the $FECPacket_{thres}$, at a packet loss rate of 10% and a GOP of 8

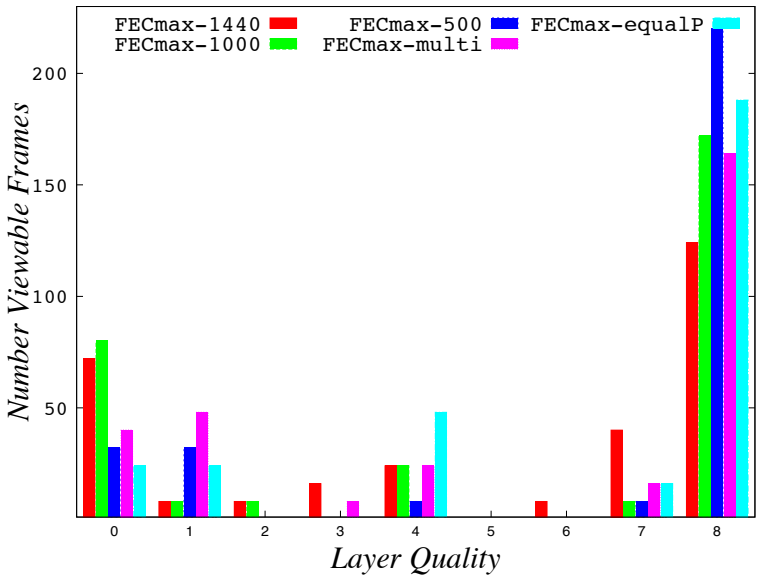


Figure C.10: Example of the number of viewable layers for SVC-SC, for each of the $FEC_{\max}Packet_{thres}$, at a packet loss rate of 10% and a GOP of 8

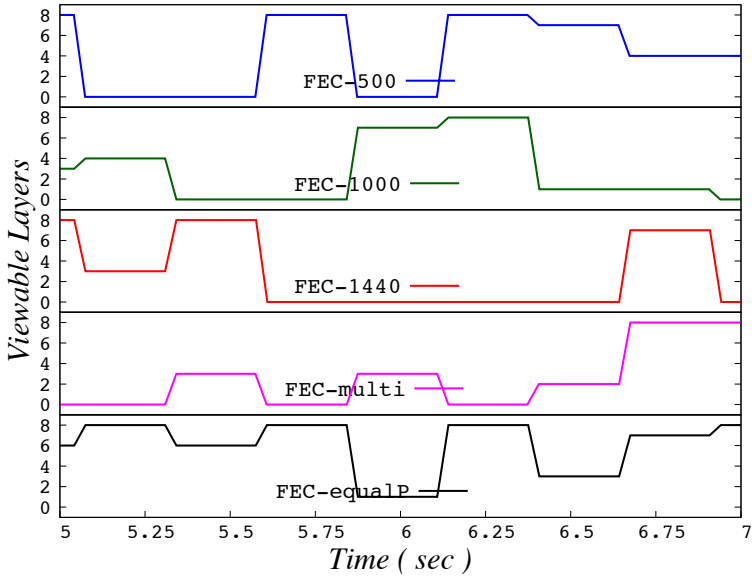


Figure C.11: A two second example of variation in viewable quality for all FEC schemes, at a packet loss rate of 10% and a GOP of 8

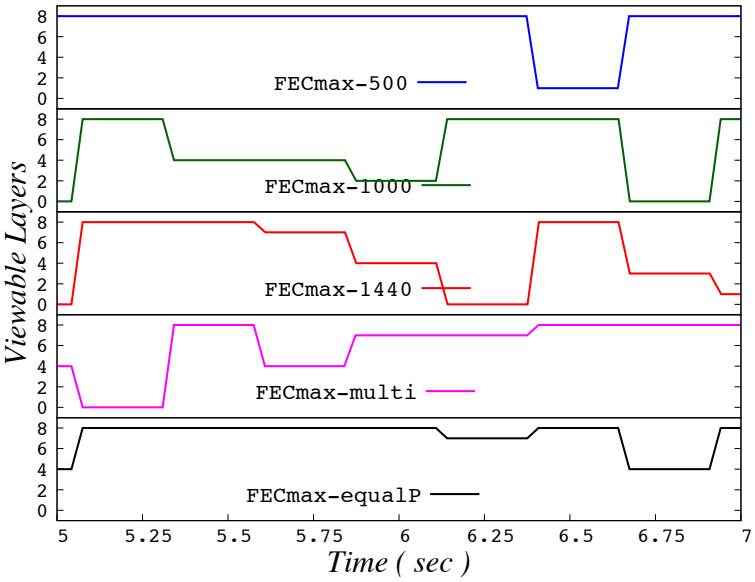


Figure C.12: A two second example of variation in viewable quality for all $FEC_{\max}$ schemes, at a packet loss rate of 10% and a GOP of 8

Plots C.11 and C.12 illustrate a direct comparison to Plots C.5 and C.6. Note the consistency of layer quality that is provided by a larger GOP value (play out over all 8 frames in the GOP), but with noticeably large variation in the quality. Again note the large increase in quality provided by the relatively small increase provided by $FEC_{\max}$, especially for '500', 'multi' and 'equalP'. As previously mentioned, the cascading effect mandated by the frame interdependence within a GOP forces all frames within a GOP to the same layer value. This can be seen to be beneficial for high quality levels, but is detrimental to users when low quality is mandated.

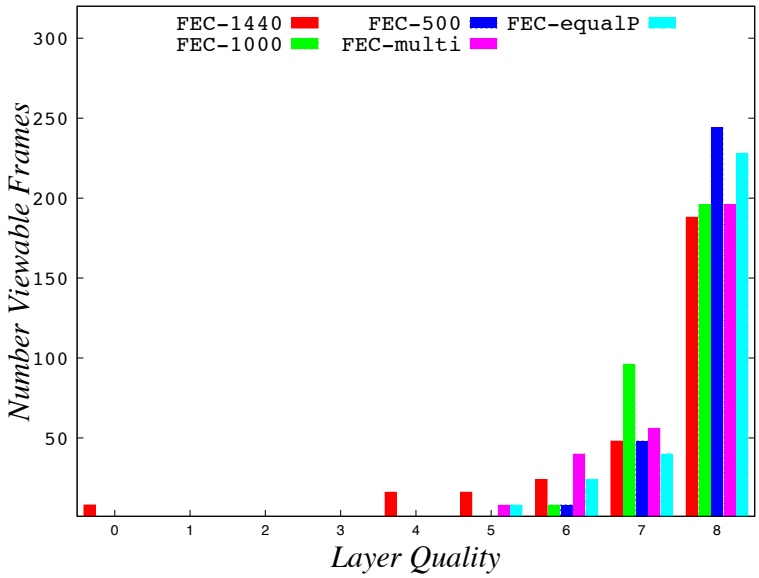


Figure C.13: Example of the number of viewable layers for SVC-SC, for each of the $FECPacket_{thres}$, at a packet loss rate of 10% and a GOP of 8. This image illustrates an increase in the FEC for C1 and C2 based on the worst case scenario, as shown in Table C.2.

Figures C.13 to C.14 illustrate a direct comparison to Figures C.7 to C.8, where the FEC levels have been increased in C1 and C2. For $FEC_{\max}$ we increase the FEC in C1 and C2 to the maximum, while maintaining the standard level of FEC to C3, and note near continuous quality over the layers in C3 for all schemes except for $Packet_{thres}$ 1,440 bytes. This illustrates that adaptation of the FEC levels in the lowest layers, or highest prioritised classes, is most beneficial to overall quality, and that lower levels of FEC can be utilised in the higher classes to increase, or maintain high levels, of viewable quality. Plot C.15 illustrates the near consistency in viewable quality at quality layer 8, for all models with $FEC_{\max}$. We begin now to see that adaptation in the GOP frame level, as well as the FEC level per class can increase overall viewable quality.

The results for GOP-8 illustrate that as GOP value increase, the same level of FEC provides for a noticeable increase in the viewable quality.

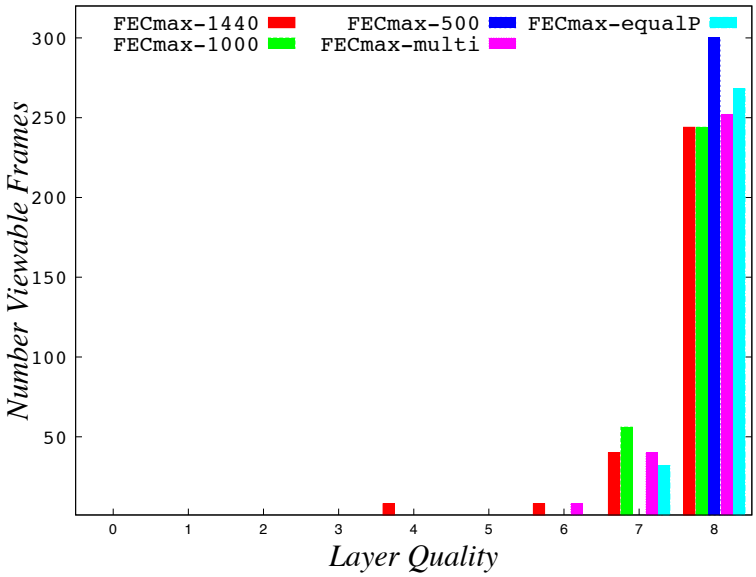


Figure C.14: Example of the number of viewable layers for SVC-SC, for each of the $FEC_{\max}Packet_{thres}$, at a packet loss rate of 10% and a GOP of 8. This image illustrates an increase in the $FEC_{\max}$ for C1 and C2 based on the worst case scenario, as shown in Table C.2.

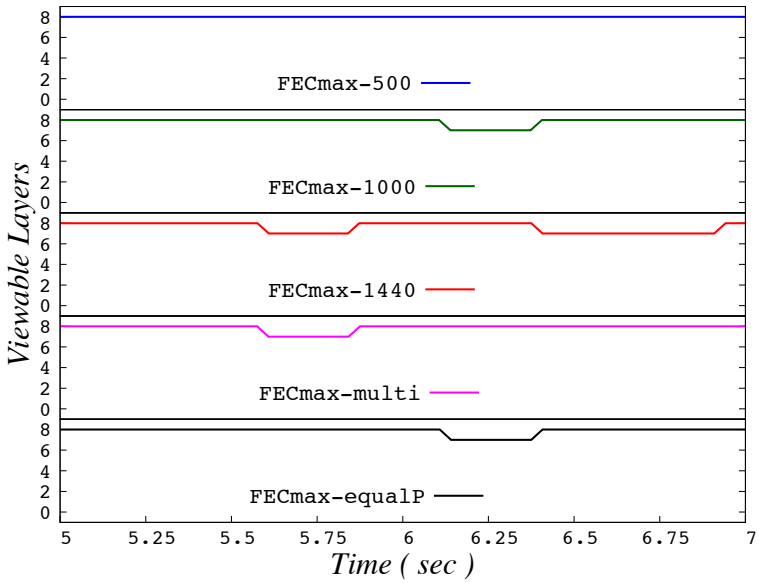


Figure C.15: A two second example of variation in viewable quality for all $FEC_{\max}$ models, at a packet loss rate of 10% and a GOP of 8. This illustrates an increase in the $FEC_{\max}$ for C1 and C2 based on the worst case scenario, as shown in Table C.2

Table C.7: FEC and $FEC_{\max}$ models for all four $Packet_{thres}$, as well as $Packet_{equal}$. Values are illustrated for packets transmitted, additional packet header cost (60 bytes per packet) and individual packets sent, packets received, loss rate per class and maximum loss rate per frame for SVC-SC, for an overall packet loss rate of 10% and a GOP of sixteen.

	SVC-SC 10% - Different packet loss rates per class, with a GOP of 16									
	FEC					$FEC_{\max}$				
Byte size	1,440	1,000	500	Multi	$Packet_{equal}$	1,440	1,000	500	Multi	$Packet_{equal}$
# C3 Trans Cost	3,720,325	3,720,920	3,722,521	3,720,844	3,721,817	3,835,615	3,835,926	3,837,584	3,836,188	3,836,895
# Packets	2,611	3,748	7,471	3,478	4,794	2,692	3,862	7,698	3,583	4,944
Header Cost	156,660	224,880	448,260	208,680	287,640	161,520	231,720	461,880	214,980	296,640
Packets Sent										
C1	307	438	869	869	1,598	317	452	893	893	1,648
C2	706	1,011	2,013	1,011	1,598	727	1,042	2,075	1,042	1,648
C3	1,598	2,299	4,589	1,598	1,598	1,648	2,368	4,730	1,648	1,648
Packets Received										
C1	267	399	781	781	1,445	287	404	804	787	1,475
C2	636	901	1,814	891	1,432	658	931	1,884	928	1,471
C3	1,420	2,051	4,123	1,425	1,426	1,452	2,119	4,235	1,480	1,491
Loss Rate (LR)										
C1	13%	9%	10%	10%	10%	9%	11%	10%	12%	10%
C2	10%	11%	10%	12%	10%	9%	11%	9%	11%	11%
C3	11%	11%	10%	11%	11%	12%	11%	10%	10%	10%
Max per Frame LR										
C1	27%	31%	17%	21%	14%	27%	25%	21%	24%	17%
C2	27%	16%	16%	21%	15%	18%	16%	16%	18%	16%
C3	20%	16%	15%	15%	19%	20%	16%	14%	15%	20%

C.3 Varying Packet Size for a GOP of Sixteen

Table C.7 is the GOP of sixteen equivalent of Table C.4. Note that as we increase GOP value, there is a noticeable decrease in max loss rate per frame, as well as the decrease in overall transmission cost. The remainder of this section compares the GOP of 16 plots and figures to the previously shown GOP of 1 and GOP of 8 plots.

Figures C.16 to C.17 illustrate a direct comparison to Figures C.9 to C.10, based on the best case, lowest level, allocation of FEC as per Table C.1. Note how high levels of layer 8 are available for $FEC_{\max}$, even with the lowest level (best case) of FEC allocation. Also note how the loss is now being forced towards the highest layer in the lower classes, *i.e.* from layer 1, layer 4, such that the SD packetisation is now best equipped to deal with loss over a larger number of frames, as well as over a greater overall bitrate per GOP.

Plots C.18 and C.19 illustrate a two second example of variation in viewable quality for all FEC schemes with minimum FEC allocation. Note how for $FEC_{\max}$, '500' and '1440' are now able to achieve consistency of quality at the highest layer for the duration of the time period shown. Further illustrating how a higher number of frames per GOP as well as an adaptive FEC allocation can be provide consistency of quality over time.

Figures C.20 to C.21 illustrate an increase in the FEC in C1 and C2 to the maximum worst case level, while maintaining the standard level of FEC to C3 as per worst case allocation rates shown in Table C.2. As can be seen in $FEC_{\max}$,

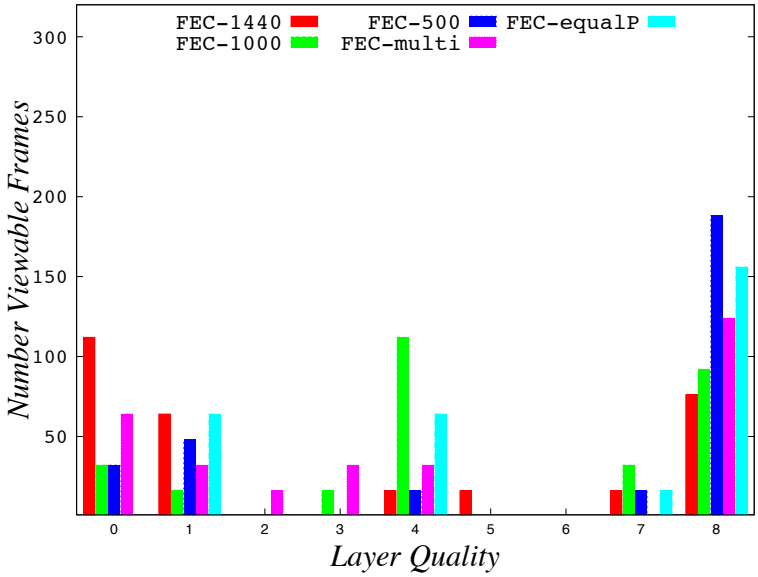


Figure C.16: Example of the number of viewable layers for SVC-SC, for each of the $FECPacket_{thres}$, at a packet loss rate of 10% and a GOP of 16

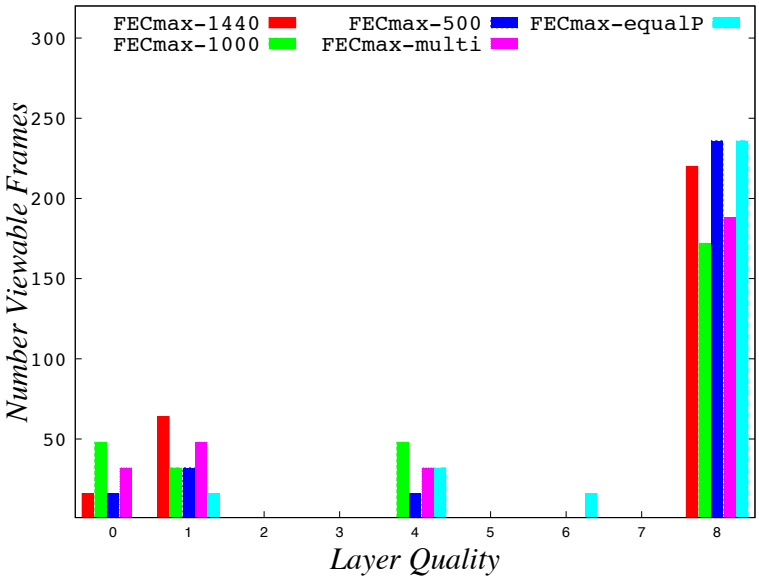


Figure C.17: Example of the number of viewable layers for SVC-SC, for each of the $FEC_{\max}Packet_{thres}$, at a packet loss rate of 10% and a GOP of 16

'500' and '1000' so no degradation in viewable quality, as all 300 frames can be shown at layer 8. This illustrates that for this level of GOP, a lower level of FEC maybe sufficient to mandate the non degradation in viewable quality. It is also important to note that all schemes are now viewable within the highest class, *i.e.* within the highest resolution, thus variations in viewable quality is limited only to fidelity levels, thus further limiting the variation in noticeable perceived quality to spatial rather than temporal issues. Plot C.22 illustrates the near consistency in viewable quality for all models with $FEC_{\max}$ within the time period illustrated.

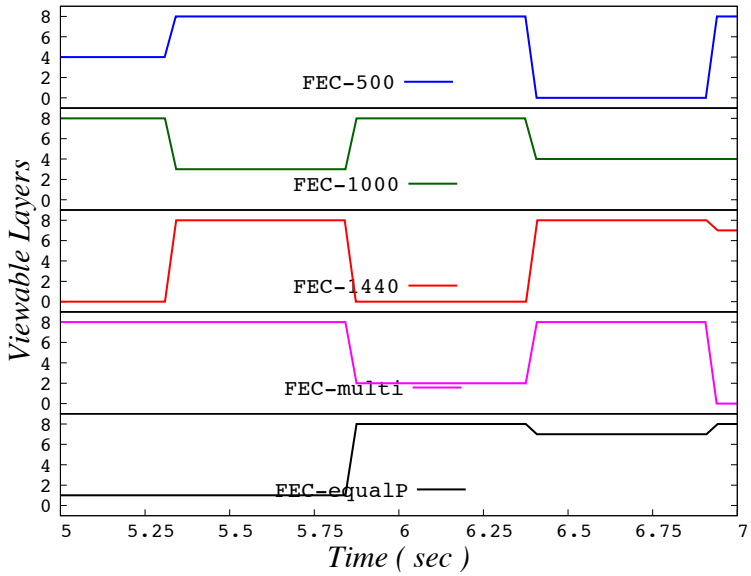


Figure C.18: A two second example of variation in viewable quality for all FEC schemes, at a packet loss rate of 10% and a GOP of 16

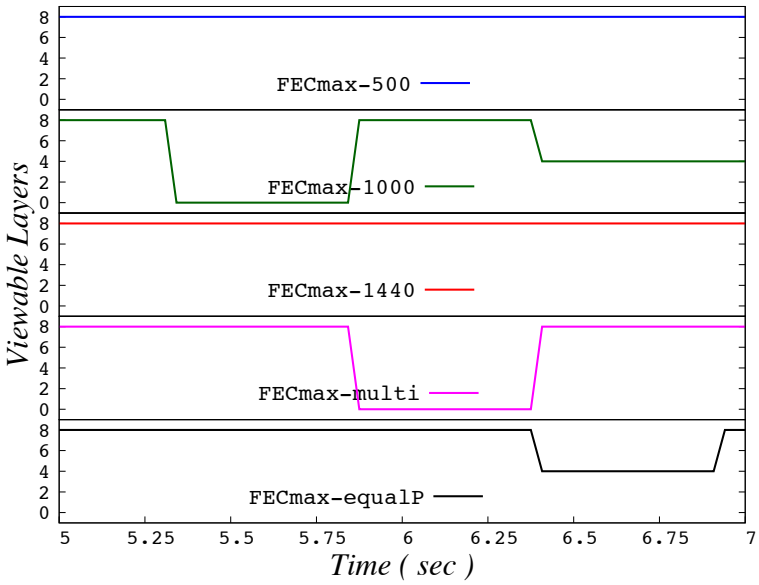


Figure C.19: A two second example of variation in viewable quality for all $FEC_{\max}$ schemes, at a packet loss rate of 10% and a GOP of 16

In Figure C.21, we have reached a error resiliency threshold for $FEC_{\max}$ models 1,000 bytes, 500 bytes using worst case FEC allocation, as we have attained continuous maximum quality decoding of layer 8. In Figures C.20 we also note that the lower FEC allocation rate is sufficient to provide full C1 and C2 decoding, while only C3 requires the additional $FEC_{\max}$ allocation to maximise quality. Thus we now have an adaptive mechanism by which to maximise quality for the individual stream classes. As we have seen before, as we increase the GOP value, the same levels of FEC allocation provide an increase in viewable quality, as loss can now be distributed over more frames, or specifically over more bytes, thus forcing the loss to the FEC allocated levels rather

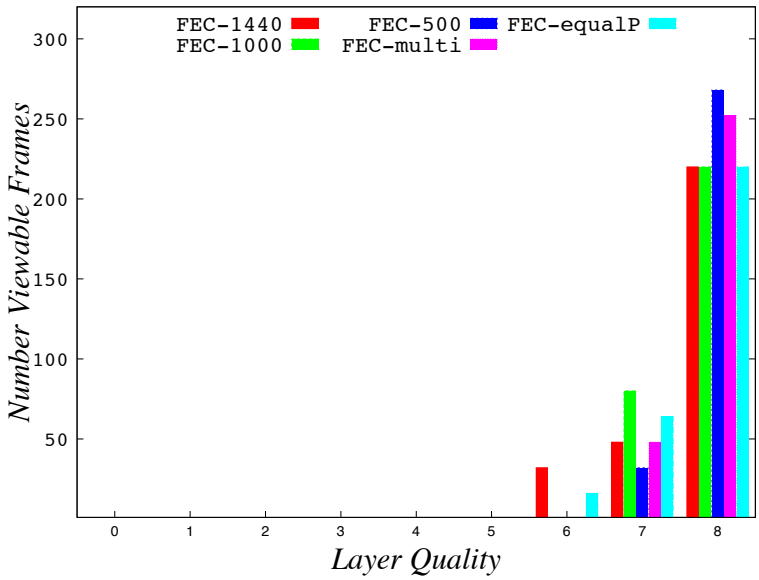


Figure C.20: Example of the number of viewable layers for SVC-SC, for each of the $FECPacket_{thres}$, at a packet loss rate of 10% and a GOP of 16. This image illustrates an increase in the FEC for C1 and C2 based on the worst case scenario, as shown in Table C.2.

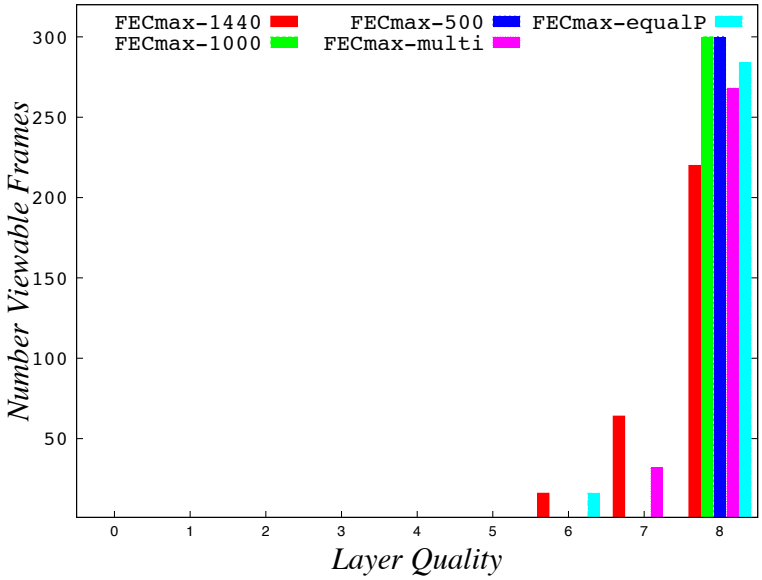


Figure C.21: Example of the number of viewable layers for SVC-SC, for each of the $FEC_{\max}Packet_{thres}$, at a packet loss rate of 10% and a GOP of 16. This image illustrates an increase in the $FEC_{\max}$ for C1 and C2 based on the worst case scenario, as shown in Table C.2.

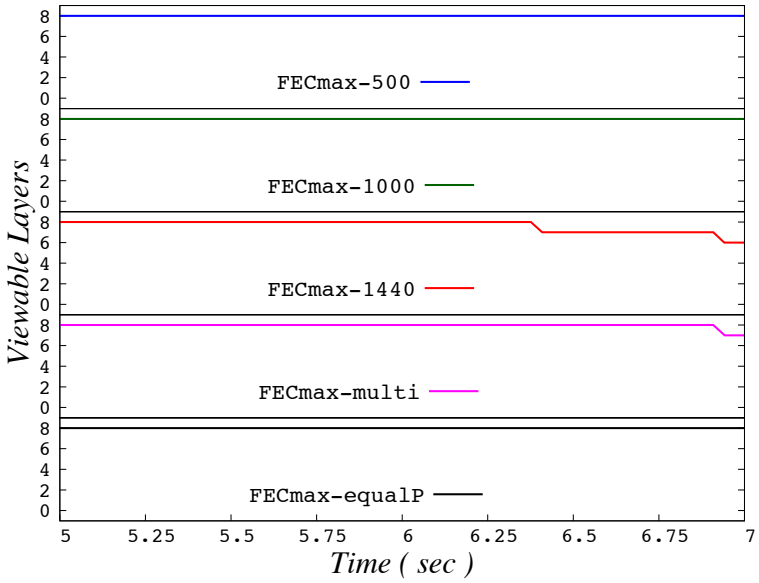


Figure C.22: A two second example of variation in viewable quality for all $FEC_{\max}$ models, at a packet loss rate of 10% and a GOP of 16. This illustrates an increase in the $FEC_{\max}$ for C1 and C2 based on the worst case scenario, as shown in Table C.2

than to specific frames or layers within the GOP.

Table C.8: FEC and $FEC_{\max}$ models for all four $Packet_{thres}$, as well as $Packet_{equal}$. Values are illustrated for packets transmitted, additional packet header cost (60 bytes per packet) and individual packets sent, packets received, loss rate per class and maximum loss rate per frame for SVC-SC, for an overall packet loss rate of 10% and a GOP of thirty two.

	SVC-SC 10% - Different packet loss rates per class, with a GOP of 32									
	FEC					$FEC_{\max}$				
Byte size	1,440	1,000	500	Multi	$Packet_{equal}$	1,440	1,000	500	Multi	$Packet_{equal}$
# C3 Trans Cost	3,158,123	3,158,382	3,160,689	3,158,353	3,159,005	3,255,771	3,256,625	3,258,020	3,256,052	3,256,772
# Packets	2,210	3,174	6,331	2,936	4,098	2,276	3,270	6,524	3,022	4,218
Header Cost	132,600	190,440	379,860	176,160	245,880	136,560	196,200	391,440	181,320	253,080
Packets Sent										
C1	255	364	723	723	1,366	263	373	742	742	1,406
C2	589	847	1,689	847	1,366	607	874	1742	874	1,406
C3	1,366	1,963	3,919	1,366	1,366	1,406	2,023	4,040	1,406	1,406
Packets Received										
C1	219	319	646	636	1,218	230	334	673	652	1,269
C2	522	771	1,524	758	1,215	546	784	1,568	786	1,252
C3	1,231	1,736	3,519	1,223	1,233	1,255	1,796	3,623	1,257	1,255
Loss Rate (LR)										
C1	14%	12%	11%	12%	11%	13%	10%	9%	12%	10%
C2	11%	9%	10%	11%	11%	10%	10%	10%	10%	11%
C3	10%	12%	10%	10%	10%	11%	11%	10%	11%	11%
Max per Frame LR										
C1	36%	23%	16%	19%	15%	22%	27%	14%	18%	15%
C2	21%	13%	14%	16%	16%	16%	14%	17%	15%	16%
C3	14%	16%	12%	14%	14%	16%	14%	12%	15%	14%

C.4 Varying Packet Size for a GOP of Thirty Two

Table C.8 contains the details for a GOP of thirty two. Note the decrease in max loss rate per frame, as well as the decrease in overall transmission cost, as GOP increases.

Rather than illustrate results for both FEC allocation tables, Table C.1 and Table C.2, we will just illustrate the results for Table C.2. We note that for the results based on Table C.1, only one specific GOP was unable to provide layer 8 decoding for $FEC_{\max}$ $Packet_{thres}$ 1,000 bytes and 500 bytes.

Figures C.23 to C.24 illustrate an increase in the FEC in C1 and C2 to the maximum worst case level, while maintaining the standard level of FEC to C3. Plot C.25 illustrates two seconds of consistency in viewable quality for all models with $FEC_{\max}$. As seen in Figure C.24 only $Packet_{thres}$ 1,440 was unable to provide continuos layer 8 (for one GOP layer 7 was decodable). We further evaluated GOP of 32 (not shown) and found that when we increased the $FEC_{\max}$ allocation of C1 from 17% and 22% to 22% and 27% respectively, based on Table C.1, that excluding $Packet_{thres}$ 1,440, all the other models were able to decode all but one of the GOP at layer 8. Further increasing C1 from 22% and 27% to 27% and 33% respectively and increasing C2 from 17%, 22% and 27% to 22%, 27% and 33% respectively increased viewable quality of the three models to full layer 8 decoding, while $Packet_{thres}$ 1,440 bytes and 'multi' were able to decode all but one of the GOP at layer 8. This further illustrates the adaptive FEC allocation, as well as the packetisation byte-allocation, mechanisms proposed can find the optimal level of FEC, but as defined a non-default initial level is required prior to GOP value determination. It is be concluded that higher GOP values will reduce

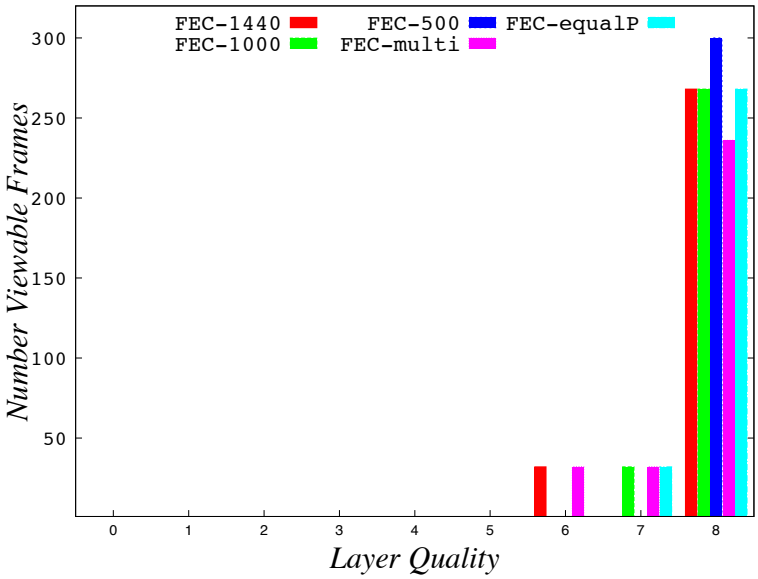


Figure C.23: Example of the number of viewable layers for SVC-SC, for each of the $FECPacket_{thres}$, at a packet loss rate of 10% and a GOP of 32. This image illustrates an increase in the FEC for C1 and C2 based on the worst case scenario, as shown in Table C.2.

overall FEC levels even further.

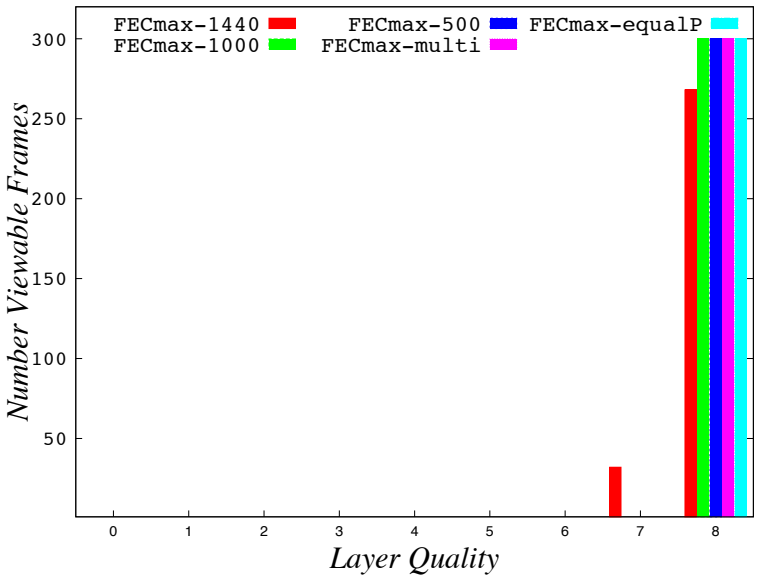


Figure C.24: Example of the number of viewable layers for SVC-SC, for each of the $FEC_{\max} Packet_{thres}$, at a packet loss rate of 10% and a GOP of 32. This image illustrates an increase in the $FEC_{\max}$ for C1 and C2 based on the worst case scenario, as shown in Table C.2.

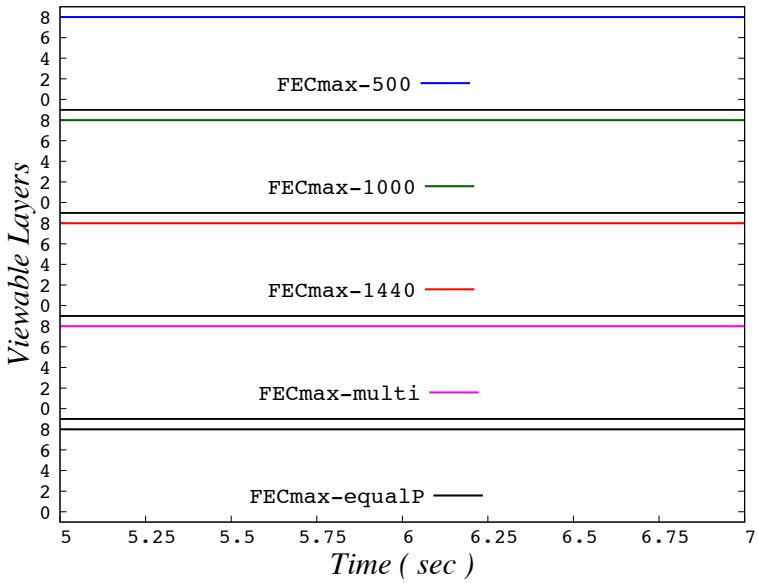


Figure C.25: A two second example of variation in viewable quality for all $FEC_{\max}$ models, at a packet loss rate of 10% and a GOP of 32. This illustrates an increase in the $FEC_{\max}$ for C1 and C2 based on the worst case scenario, as shown in Table C.2

C.5 Conclusion

We noted with a GOP of 1, that the worst case FEC allocation could not provide continuous layer 8 quality, while with a GOP of 8 the same level of FEC allocation provides near continuous layer 8 quality for all schemes. While for a GOP of 16 and GOP of 32 we were able to reduce FEC allocation rates for C1 (GOP of 16) and for C1 and C2 (GOP of 32) while mandating maximised viewable quality. Thus illustrating both adaptation in FEC allocation and packet byte-size allocation by which viewable quality can be preserved and stabilised during moment of packet loss.

As GOP increases, we note that the range of maximum per frame loss rates is smaller. Thus leveraging the benefits of sharing the GOP loss rate over more frames. Note: the maximum loss rate per frame is actually the maximum loss rate per GOP, but because of how we packetise the layer and frames per GOP, both per frame and per GOP values are equal. In these results, we did not increase the FEC or $FEC_{\max}$ for C3. Increasing error resiliency in C3 will only increase viewable quality in C3, while increases in error resiliency in lower classes benefit both the individual lower class and all higher classes. As seen, as GOP values increases, the minimum level of FEC or $FEC_{\max}$ for C3 was adequate for most models to stream at maximum quality.

Finally, we note that for a GOP of 1 that a transmission cost of 12,447,975 bytes is required for $FEC_{\max}$ while a GOP of 32 will mandate a reduced transmission cost of 3,494,525 bytes, thus illustrating that lower FEC rates per layer and a lower overall transmission cost can provide maximised viewable quality at higher GOP values.

Glossary

1080p Resolution of 1920x1080 (WxH).

4CIF Resolution of 704x576 (WxH).

4K Also known as UHD.

ALD Adaptive Layer Distribution.

ANA Application-Layer Network Aware.

AU Access Units - SVC specific.

AVC Advanced Video Coding Standard (H.264).

B-C Best Case.

BL Base Layer - SVC specific.

CDN Content delivery network.

CGS Course-Grained Scalability.

CIF Resolution of 352x288 (WxH).

CMSE Centralised Media Streaming Element.

DASH Dynamic Adaptive Streaming over HTTP.

dB decibels.

Did Dependency identifier - also known as Lid.

Diffserv Differentiated services.

DTQ (dependency_id, temporal_id, quality_id).

EL Enhancement Layers - SVC specific.

ESS Extended Spatial Scalability.

FEC Forward Error Correction.

GOF Group of Frames.

GOP Group of Pictures.

HD High Definition - see 1080p for resolution size.

HEVC High Efficiency Video Coding Standard (H.265).

ICC Independent Class Composition.

ICI Increased Class Interdependency.

IDD Incremental Datagram Delivery.

IER Improved Error Resiliency.

IPTV Internet Protocol Television.

IRP Improved Resiliency Packetisation.

I-SDC SDC with Incremental Datagram Delivery.

JSVM Joint Scalable Video Model.

LId Dependency identifier - also known as Did.

MBR Multi-bitrate.

MDC Multiple Description Coding.

MGS Medium-Grained Scalability.

MLS Multi-protocol Label Switching.

MOS Mean Opinion Score.

MSE Mean Squared Error.

MTAP Multiple Time Aggregation Packet.

NAL Network Abstraction Layer.

P2P Peer to Peer.

PET Priority Encoding Transmission.

PSNR Peak Signal-to Noise Ratio.

PSS 3GPP Packet-switched Streaming Service.

QCIF Resolution of 176x144 (WxH).

Qid Quality identifier.

QOE Quality of Experience.

QoP Quality of Perception.

QoS Quality of Service.

QP Quantisation Parameter.

ROP Reduced Overhead Packetisation.

RTCP Real-Time Control Protocol.

RTP Real-Time Transport Protocol.

SC Streaming Class.

SD Section Distribution.

SDC Scalable Description Coding.

SDC-NC Network Coding for SDC.

SDN Software Defined Networks.

SDP Section-based Description Packetisation.

SMDC Scalable Multiple Description Coding.

SSIM Structural Similarity Index Metric.

STAP Single Time Aggregation Packet.

STF Section Thinning Factor.

SVC Scalable Video Coding.

SVC-SC SVC using a Streaming Class model.

TCP Transmission Control Protocol.

TId Temporal identifier.

UDP User Datagram Protocol.

UEP Unequal Error Protection.

UHDTV Ultra High Definition TV - approximately four times the resolution of HD.

Video Coding Layer Video Coding Layer.

W-C Worst Case.

XOR exclusive disjunction.