Performance of Single Layer H.264 SVC Video Over Error Prone Networks

by

Hari Sundararaman

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2011 by the
Graduate Supervisory Committee:

Martin Reisslein, Chair
Patrick Seeling
Cihan Tepedelenlioglu

ARIZONA STATE UNIVERSITY

May 2011

ABSTRACT

With tremendous increase in the popularity of networked multimedia applications, video data is expected to account for a large portion of the traffic on the Internet and more importantly next-generation wireless systems. To be able to satisfy a broad range of customers requirements, two major problems need to be solved. The first problem is the need for a scalable representation of the input video. The recently developed scalable extension of the state-of-the art H.264/MPEG-4 AVC video coding standard, also known as H.264/SVC (Scalable Video Coding) provides a solution to this problem. The second problem is that wireless transmission medium typically introduce errors in the bit stream due to noise, congestion and fading on the channel. Protection against these channel impairments can be realized by the use of forward error correcting (FEC) codes. In this research study, the performance of scalable video coding in the presence of bit errors is studied. The encoded video is channel coded using Reed Solomon codes to provide acceptable performance in the presence of channel impairments. In the scalable bit stream, some parts of the bit stream are more important than other parts. Parity bytes are assigned to the video packets based on their importance in unequal error protection scheme. In equal error protection scheme, parity bytes are assigned based on the length of the message. A quantitative comparison of the two schemes, along with the case where no channel coding is employed is performed. H.264 SVC single layer video streams for long video sequences of different genres is considered in this study which serves as a means of effective video characterization. JSVM reference software, in its current

version, does not support decoding of erroneous bit streams. A framework to obtain H.264 SVC compatible bit stream is modeled in this study. It is concluded that assigning of parity bytes based on the distribution of data for different types of frames provides optimum performance. Application of error protection to the bit stream enhances the quality of the decoded video with minimal overhead added to the bit stream.

DEDICATION

To my affectionate parents,

Respectful grandparents,

Lovable sister and

Dear friends

ACKNOWLEDGMENTS

Firstly I would like to acknowledge the love and support of my parents, loving sister and my grandparents who have always wished for my success. Secondly, I would like to thank 'God' for his support and shoulder, whenever I needed them.

I would like to express my sincere gratitude to my advisor Professor Martin Reisslein for his invaluable guidance and persistent support. Working with him has been my true pleasure and has trained me both professionally and personally. Without his advice, this work would not have been possible. I would like to specially thank Dr.Patrick Seeling for his support whenever I faced problems and constant encouragement and for his gracious willingness to be part of my Master's Thesis committee. I would like to thank Dr.Cihan Tepedelenlioglu for helping me better understand the concepts in Digital and Wireless communications and for his gracious willingness to be part of my Master's Thesis committee. I am grateful to all of them for their interest in my work, and their valuable suggestions and insights.

I would like to thank all my friends and colleagues in the Multimedia and Wireless Networking labs. I have learned how to work and share with them during my studies. I would like to thank them for every help. I would like to thank all my friends and well-wishers at ASU and beyond, for their love, encouragement, and constant support. Specifically, I would like to thank Akshay, Pulipaka, Vikram Neelakandan, Rohit Raghunathan, Harish Kumar and Keshav Narayanan for their help rendered during the course of completion of my thesis.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF TABLES

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction

With tremendous increase in the popularity of networked multimedia applications, video data is expected to account for a large portion of the traffic in the Internet and next-generation wireless systems. Transmitting uncompressed video content is not feasible as the amount of data to be transmitted will be very high (in the range of gigabytes). For ease of transport over networks, video is encoded (i.e., compressed) to reduce the bandwidth requirements. Even compressed video requires large bandwidths to accommodate large data rates of the order of hundreds of Kilobits per second (Kbps) or Megabits per second (Mbps). In the present day, end-users have a large choice of multimedia devices at their disposal, going from high-resolution HDTVs to low-resolution mobile phones.

To be able to satisfy a broad range of customers requirements, two major problems need to be solved. The first problem is that, due to diversity of end user requirements arising from a broad range of mobile terminals and the variety of available screen resolutions, we require a scalable representation of the input video. The recently developed scalable extension of the state-of-the art H.264/MPEG-4 AVC video coding standard, also known as H.264/SVC (Scalable Video Coding) [12] provides a solution to this problem. It aims to bring varying scalability, to provide a high degree of coding efficiency without significant increase in the decoder complexity compared to their corresponding non-scalable

profiles. H.264/SVC supports a variety of scalability modes such as temporal, spatial and quality scalability. The video stream can be encoded into multiple layers each with different frame rates (temporal scalability), different spatial resolutions (spatial scalability) or different fidelity levels (quality or SNR scalability). The second problem is that wireless transmission medium typically introduce errors in the bit stream due to noise, congestion and fading on the channel. Protection against these channel impairments can be realized by the use of forward-error correcting (FEC) codes. Using FEC codes, parity symbols are added to the original bit stream to achieve improvement in the quality of the reconstructed vide at the decoder side.

Stringent Quality of Service (QoS) requirements (loss and delay) of video traffic makes the transport of video traffic over communication networks a challenging problem. This has led to a tremendous interest in the networking research community to study all aspects of video transport. Performance evaluation of H.264/SVC single layer video traffic in the presence of bit errors introduced by the channel is being investigated in this research study. Long video sequences are used in this research study to derive meaningful statistical estimates of the video traffic. Reed Solomon encoding [8] is used in this simulation model to correct errors caused due to channel impairments. The reason for choosing Reed Solomon codes among various error control coding schemes is because the DVB-T standards specifies the use of Reed Solomon encoding as a preferred physical layer FEC scheme [31]. More specifically, a quantitative study of the

improvement in quality of the reconstructed video due to the application of error protection is performed. Actual video bit streams have a limitation that they are usually proprietary and/or protected by copyright [13]. This limits the access of networking researchers to bit streams, and also limits the exchange of bit streams among different research groups. The bit streams give the actual bits carrying the video information, but the traces only give the number of bits used for encoding each individual video frame. So, the simulation model is replicated with video data generated using the video trace information. The performance degradation of the video after transport over the network using the offset distortion approach [13] is measured and compared with the results obtained using the actual bit stream approach.

1.2 Related Works

Research on application of Forward Error Correction (FEC) schemes to improve the performance of transmission of video over error prone networks has gained prominence in the last few years with rapid increase in multimedia transmission over wireless networks [2], [3], [7], [4], [6], [9], [17]. The focus has been on testing streaming models for short video sequences of frame length around 100 frames over wired and wireless channels. No prior models have tested with transmission of long video sequences over the network which would provide meaningful statistical estimates. For transmission over wireless networks, FEC schemes are used on top of the video data to provide for correction of errors due to channel impairments. With increasing popularity on scalable video coding, lot

of research has been directed towards comparing unequal error protection versus equal error protection of packets for different packet loss scenarios. Very few studies have focused on evaluating the performance due to the effect of bit errors [4]. [2], [7], [3] have focused on comparing equal error protection versus unequal error protection for different packet loss scenarios. In [2], the case of wired transmission is considered where packets losses are not high. It is observed that using unequal error protection, graceful degradation of the video quality is achieved when the targeted packet loss probability is exceeded. [7], [18] presents a simulation model for performance of the system due to transmission of packets over error prone wireless networks. They have not considered the case where the bits of a packet are in error, which is quite common, whenever noise is present in the channel. [6] provides a framework for transmission of MGS video sequences over real time networks. Here, the packets are dropped based on their arrival time at the receiver end. [4] proposes the use of hierarchical QAM modulation to provide unequal error protection to a data partitioned H.264 SVC video streams. A study of Bit Error Rate (BER) v Signal to Noise Ratio (SNR) for different bit rates for only 1 CIF video sequence of 100 frames is performed. [9] proposes an unequal error protection scheme for different packet loss scenarios where more important layers are fully protected while according to rate limitations, no extra data is added to less important layers. This study examines the performance of FGS video sequence of length 32 frames. [7] proposes a simulation model for transmission of unequally error protected H.264 video over 802.11 b/g wireless networks based on Reed Solomon codes. It examines only one video sequence of

180 frames length. [3] propose an unequal error protection scheme for FGS video sequences of length 50 frames. [17] analyzes the use of unequal protection schemes for enhancing the video quality over a wider range of error rates compared with single layer protection schemes by using a "graceful degradation" like functionality. This study uses FGS video sequences which at present have been removed from the H.264/SVC standard due to high complexity of encoding. [16] exploits the traffic prioritization capabilities offered by 802.11e to provide better protection to the most perceptually important parts of a video while achieving efficient network resource usage. This study uses 300 length frame sequences with spatial scalability.

1.3 Outline

The effect of application of error protection on the quality of reconstructed video, after passing through a communication channel, is examined in this study. We perform a quantitative comparison of the reconstructed video quality for three different scenarios- firstly when unequal error protection is applied based on the importance of different NAL units, secondly when equal error protection is applied, keeping the overhead (in terms of parity bytes) constant, and assigning parity bytes based on the size of different NAL units and finally comparing it with the case where no channel coding is used. The study is organized as follows.

Chapter 2 gives a very brief overview of H.264/SVC video coding standard in view of understanding the video encoding process. Chapter 3 emphasizes about

two important concepts of H.264/SVC namely NAL units and scalability, based on which our error protection scheme is formulated. Chapter 4 provides a description of the proposed simulation model covering the video encoding process, error protection scheme used in this study, some basics about Reed Solomon codes, transport of channel coded data over the wireless channel and reconstruction of the erroneous bit stream by decoding the NAL units. In chapter 5, the reconstruction of the decoded sequence from the erroneous bit stream is discussed in detail as the current version of the JSVM reference software doesn't support error concealment techniques. Chapter 6 discusses the results obtained for Sony Demo, Star Wars and Silence of the Lambs sequences for the case of equal error protection, unequal error protection and no error protection. Also directions for future research are discussed.

Chapter 2

OVERVIEW OF H.264 SVC VIDEO CODING STANDARD

This section presents a brief overview of the H.264/SVC standard to understand the concept of video encoding, the various types of frames and the concept of NAL units. For a detailed discussion on H.264/SVC, H.264/AVC, previous standards of the MPEG-4 family and the concepts involved in them, please refer [11, 12, 19, 20, 22].

2.1 The H.264/SVC Standard

The SVC standard was developed as the Annex G extension to the H.264/MPEG-4 AVC video compression standard [12], preserving its well-designed core coding tools. The idea of subdivision of pictures into macro blocks (MBs) and slices is used in this standard. It supports three basic slice coding types namely I-Slice (intra-picture predictive coding using spatial prediction from neighboring regions), P-Slice (intra-picture predictive coding and inter-picture predictive coding with one prediction signal for each predicted region) and the B-Slice (intra-picture predictive coding, inter-picture predictive coding, and inter-picture bipredictive coding with two prediction signals that are combined with a weighted average to form the prediction region). For I-slices, the standard provides several directional spatial intra prediction modes and for P-slice and B-slice, in addition to the features provided for I-slice, variable block size motion-compensated prediction with multiple reference pictures is permitted [21]. The H.264/AVC standard, states a set of integer transforms of different block sizes for transform

coding [22]. The standard uses uniform reconstruction quantizers. The Quantization Parameter (QP) allows the selection of one of the 52 quantization step-sizes for each macro-block [22]. Two entropy coding schemes are supported by the standard namely Context Adaptive Binary Arithmetic Coding (CABAC) and the Context Adaptive Variable Length Coding (CAVLC). CABAC uses arithmetic coding and more sophisticated mechanisms for employing statistical dependencies to achieve around 10-15% bit rate savings over CAVLC [22, 24]. An Adaptive de-blocking filter which helps in reducing blocking artifacts is specified by the H.264/AVC standard [22]. Some of the other standard tools included in the H.264/AVC standard are spatial intra-frame prediction, variable block sizes and Lagrangian-based rate-distortion optimization [25].

The design of H.264/AVC into a video coding layer (VCL) and a Network Abstraction layer (NAL) has also been carried forward into the SVC extension [12]. The VCL makes a coded representation of the original content, while the NAL facilitates the ability to map H.264/SVC VCL data to transport layers so that various systems can use the VCL data effectively. Network Abstraction Layer (NAL) units help adapt the video bit stream to a packet network by creating the natural packet boundaries of the data packets to be used. A NAL unit is effectively a packet that contains a certain integer number of bytes of the coded video data. In H.264/AVC standard, a NAL unit consists of a 1-byte header and a NAL unit payload of varying size which is modified in H.264/SVC into a 4-byte header to include the scalability information. More information on the NAL unit

header format is presented in Chapter 3. The coded video data is in effect a group of NAL units. The NAL units are organized into access units, where an access unit is defined as a set of consecutive NAL units with specific properties that make up one decoded picture. Each access unit contains a set of VCL NAL units that together compose a primary coded picture. An access unit delimiter is prefixed before the access unit specifying the start of the access unit. This information is used in our parser implementation, discussed later in Chapter 4, to split the bit stream into individual NAL units. Redundant coded pictures may follow the primary coded picture. They contain redundant representations of the same video picture and are useful in the event of loss or corruption of the data in the primary coded pictures. A coded video sequence contains a set of consecutive access units and use only one sequence parameter set. It is always preceded by an instantaneous decoding refresh (IDR) access unit. An end of sequence NAL unit is used to indicate the end of the sequence, in the event that the coded picture is the last picture of a coded video sequence. An end of stream NAL unit may be used to indicate that the stream is ending if the coded picture is the end of the NAL unit stream [24]. An IDR access unit contains an intra picture and specifies that the set of access units that follow can be decoded without the need for decoding any previous pictures in the bit stream.

The most important feature of H.264/SVC standard that supports features which are required for efficiently supporting the required types of scalability namely temporal scalability, spatial scalability and quality scalability [22]. The

H.264/SVC allows for limited inter-layer prediction from all the layers of the scalable layer representation, down to the base layer, as against the older scalable codec which only allowed prediction within a layer and from the next lower layer [23]. The H.264/SVC employs the concept of single-loop decoding to reduce the decoding complexity. If there is no single-loop decoding, then a separate decoding loop has to be employed for each scalable layer to be decoded thereby increasing the computational demands on the decoder. So only one decoding loop needs to be used by the decoder to decode any number of scalable layers. In H.264/SVC, motion compensation is carried out only once, as the motion compensation loop will be shared by the decoding loops [23]. Another important feature of the SVC design is that it supports a scalable bit stream that can be decoded by a H.264/AVC decoder, to obtain the basic base layer features [22].

Chapter 3

CONCEPT OF NAL UNITS AND SCALABILITY

In this chapter, we take a closer look at two important concepts of H.264/SVC namely NAL units and Scalability. A good understanding of these concepts is required for better understanding of the proposed simulation model.

3.1 Concept of NAL units

The design of H.264/SVC contains two layers, the Video Coding Layer (VCL) and the Network Abstraction Layer (NAL) [15, 23]. The VCL makes a coded representation of the original content, while the NAL facilitates the ability to map H.264/SVC VCL data to transport layers so that various systems can use the VCL data effectively. The NAL unit is effectively a packet that contains a certain number of bytes of the coded video data. The NAL units are organized into access units, where an access unit is a set of consecutive NAL units with specific properties. The decoding of an access unit results in exactly one decoded picture. A set of consecutive access units with certain properties is referred to as a coded video sequence. A coded video sequence represents an independently decodable part of a NAL unit bit stream. A coded video sequence always starts with an instantaneous decoding refresh (IDR) access unit, which signals that the IDR access unit and all following access units can be decoded without the need for decoding any previous pictures of the bit stream.

NAL units can be categorized into three types –

- Video Coding Layer (VCL) NALU - VCL NALUs can be a coded slice of an Instantaneous Decoding Refresh (IDR) picture, coded slice of an SVC enhancement layer or a Prefix NAL unit. The prefix NAL units are used to attach SVC related information to non-SVC NAL units and these type of NAL units directly precede all non-SVC VCL NAL units (NAL units corresponding to the base layer VCL data) in an SVC bit stream and contain the SVC NAL unit header extension. These NAL units are usually between 8-10 bytes in size.
- Parameter Set NALU - Parameter set NALUs contain the sequence-level header information (sequence parameter sets (SPS)) and the picture-level header information (picture parameter sets (PPS)) header.
- SEI NALU - An SEI NAL unit contains one or more SEI messages. SEI messages are not required for the decoding of the output picture but they aid in related processes such as picture output timing, rendering, error detection, error concealment and resource reservation.

Without reception of the coded slice of IDR picture, SPS, PPS and SEI, the video information cannot be decoded properly. In our experiments, assumption is made that these packets are transmitted separately through a reliable channel and decoded correctly. In SVC, coded pictures from different scalable layers may use different SPS. 12 additional types of SEI messages have been specified in SVC, in addition to the original 24 in the H.264/AVC standard [23].

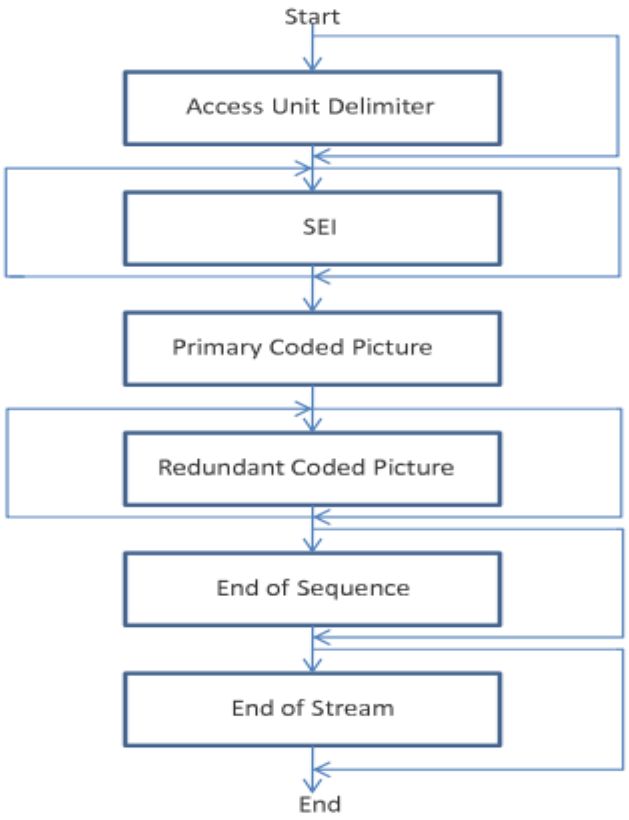


Figure 3.1: Structure of a NAL access unit [24]

The semantics of the components of header information of a H.264/SVC NAL unit is explained in [14]. The first 4 bytes of each NAL unit is the header of the NALU that contains information such as type of data in the NAL unit and scalability information. Figure 3.2 shows the format of the H.264/SVC NAL unit header [14].

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
F	NRI			NUT				R	I			PID				N	DID			QID				TID				U	D	O	R2

Figure 3.2: Structure of a NAL unit header [14]

F: 1 bit *forbidden_zero_bit.*

This bit is used to support wire line/wireless gateways. The H.264 specification declares a value of 1 as a syntax violation. Instead of disposing of NAL units with known bit errors that occurred on a wireless link, a gateway may indicate the presence of errors with the help of this bit.

NRI: 2 bits *nal_ref_idc.*

A value of "00" (in binary form) indicates that the content of the NAL unit is not used to reconstruct reference pictures for inter picture prediction. Such NAL units can be discarded without risking the integrity of the reference pictures in the same layer. A value greater than "00" indicates that the decoding of the NAL unit is required to maintain the integrity of reference pictures in the same layer, or that the NAL unit contains parameter sets. For SVC enhancement layers, a slice or slice data partitioning NAL unit with an NRI value of 11 indicates that it belongs to a key picture. A key picture is the first picture in decoding order within each group of pictures (GOP).

NUT: 5 bits *nal_unit_type.*

This component specifies the NAL unit type. NAL unit type 5 is used for Coded slice of Instantaneous Decoding Refresh (IDR) pictures and NAL unit type 1 is used for all other base layer pictures. NAL unit type 6 is used for SEI NAL units. NAL unit type 14 is used for prefix NAL unit, NAL unit type 15 is used for subset sequence parameter set, and NAL unit type 20 is used for coded slice in scalable extension. NAL unit types 14 and 20 indicate the presence of three additional bytes in the NAL unit header which contains the scalability information.

R: 1 bit *reserved_one_bit.*

Reserved bit for future extension. R must be equal to 1. The value of R must be ignored by decoders.

I: 1 bit *idr_flag.*

This one bit flag specifies whether the layer representation is an instantaneous decoding refresh (IDR) layer representation (when equal to 1) or not (when equal to 0).

PRID: 6 bits *priority_id.*

This flag specifies a priority identifier for the NAL unit. A lower value of PRID indicates a higher priority. PRID is used for inferring the values of Dependency ID (DID), temporal level ID (TID), and Quality level ID (QID)

N: 1 bit *no_inter_layer_pred_flag*.

This flag when set to 1 specifies whether inter-layer prediction may be used for decoding the coded slice.

DID: 3 bits *Dependency ID*

This component indicates the inter-layer coding dependency level of a layer representation. At any access unit, a layer representation with a given dependency ID may be used for inter-layer prediction for coding of a layer representation with a higher dependency ID, while a layer representation with a given dependency ID shall not be used for inter-layer prediction for coding of a layer representation with a lower dependency ID. This component indicates the spatial resolution level in spatial scalability. This component indicates the quality level of a CGS layer representation.

QID: 4 bits Quality ID

This component indicates the quality level of an MGS layer representation. At any access unit and for identical dependency ID values, a layer representation with quality ID equal to ql uses a layer representation with quality ID equal to (ql-1) for inter-layer prediction.

TID: 3 bits Temporal ID.

This component indicates the temporal level of a layer representation. The temporal ID is associated with the frame rate. A lower value of temporal ID corresponds to lower frame rates and higher value of temporal ID corresponds to higher frame rates. A layer representation at a given temporal ID typically depends on layer representations with lower temporal ID values, but it never depends on layer representations with higher temporal ID values.

U: 1 bit use_ref_base_pic_flag.

A value of 1 indicates that only reference base pictures are used during the inter prediction process. A value of 0 indicates that the reference base pictures are not used during the inter prediction process.

D: 1 bit discardable_flag.

A value of 1 indicates that the current NAL unit is not used for decoding NAL units with values of dependency ID higher than the one of the current NAL unit, in the current and all subsequent access units. Such NAL units can be discarded without risking the integrity of layers with higher dependency ID values. Discardable_flag equal to 0 indicates that the decoding of the NAL unit is required to maintain the integrity of layers with higher dependency ID.

O: 1 bit output_flag

This bit affects the decoded picture output process. The output timestamp is utilized to decide whether a decoded picture stored in the Double Picture Buffer is needed for future output.

RR: 2 bits reserved_three_2bits.

Reserved bits for future extension. RR must be equal to "11" (in binary form). The value of RR must be ignored by decoders.

3.2 Concept of Scalability

The H.264/SVC standard is an extension of H.264/AVC standard that supports features which are required for efficiently supporting the required types of scalability namely temporal scalability, spatial scalability and quality scalability [26]. The H.264/AVC standard offers only temporal scalability.

3.2.1. Scalability

The H.264/SVC standard includes spatial (resolution), temporal (frame rate), and quality (fidelity levels) scalability. Each layer is identified by a layer identifier. There are separate layer identifiers for temporal, spatial and quality layers namely TID, DID and QID in the NAL unit header. A reference layer is used to predict the coding of another layer with a higher layer identifier. The layer with the layer identifier equal to zero is called the base layer. The layers that employ data of other layers for coding are called enhancement layers [27]. An enhancement layer is a called a spatial enhancement layer when the spatial resolution changes

relative to the reference layer and is called a quality enhancement layer if the spatial resolution is the same as that of the reference layer however the quality level changes in this case [27]. The number of layers in an SVC bit stream depends on the application needs and demands. H.264 SVC standard supports up to 128 layers in a bit stream [27]. However, with the existing profiles present in the standard, the maximum number of enhancement layers is limited to 47 layers with a maximum of two spatial enhancement layers [27].

3.2.2. Temporal Scalability

For Temporal scalability, the temporal layers are each identified by a temporal layer identifier, TID, in the NAL unit header. The base layer is represented by TID value=0, and the temporal layer identifier increases by 1, for each subsequent temporal enhancement layer. The concepts used to attain temporal scalability are motion-compensated prediction, hierarchical prediction structures and reference picture memory control [22]. The motion-compensated prediction as shown in [22] is restricted to reference pictures with a temporal layer identifier that is less than or equal to the temporal layer identifier of the picture to be predicted. H.264/SVC provides improved temporal scalability compared to previous standards because it allows coding of picture sequences with arbitrary temporal dependencies. This is known as the reference picture memory control. More than one reference picture can be used to construct reference picture lists, when this concept is carried forward with the hierarchical prediction structures. It can also include pictures from the same temporal level as the picture to be predicted [23].

3.2.3. Spatial Scalability

Spatial scalability is supported by the H.264/SVC extension, which uses the multi-layer coding approach. Each spatial layer corresponds to a supported spatial resolution and is associated with a dependency identifier DID in the NAL unit header. The base layer has DID=0 and it increases by 1 for every subsequent spatial enhancement layer. The maximum number of enhancement layers supported presently by the H.264/SVC standard is 2 [27]. Motion compensated prediction and intra-prediction are used in all spatial layers. In order to increase coding efficiency of enhancement layers, H.264/SVC introduces additional inter-layer prediction mechanisms like prediction of macro block modes and associated motion parameters, and predication of the residue signal [22]. A detailed discussion on spatial scalability is present in [22, 27]. SVC design supports spatial scalability with arbitrary resolution ratios. The various resolution ratios are QCIF and CIF resolutions. The base layer is usually encoded at QCIF resolution and the enhancement layer is encoded at CIF resolution.

3.2.4. Quality Scalability

Quality scalability is providing scalability in terms of the quality of the picture. Quality scalable layers have the same spatio-temporal resolution but differ in fidelity. Two quality scalable modes namely Coarse-Grain Scalability (CGS) and Medium-Grain Scalability (MGS) are supported in the H.264/SVC standard. An obsolete version of granular scalability called fine-grained scalability (FGS) has

recently been removed from the SVC extension, due to its high computational complexity [28].

FGS provides finer quality refinements relative to the number of bits. The FGS enhancement layer has the special property that it can be cut at any bit rate and the received information can be decoded to add upon the quality of the base layer [28]. Even though it provides flexibility, it comes at the cost of low coding efficiency.

CGS layers differ in the layer identifiers, so each CGS layer is identified by a unique value of DID. H.264/SVC supports up to eight CGS layers, corresponding to eight quality extraction points [22]. CGS scalability provides only a few limited bit rates in a scalable bit stream compared to MGS and FGS schemes. These rate points depend on the number of CGS layers, as each CGS layer corresponds to a specific rate point. All the enhancement layer packets have to be received by the decoder to construct a quality enhancement layer. The Instantaneous decoding refresh (IDR) Access units act as switching points between different CGS layers, because these CGS layers can switch only at these specified points in the bit stream depending on the target layer. However, relying on such access unit causes reduced coding efficiency due to frequent coding of such access units. To solve this problem, SVC allows encoding of IDR pictures independently for each layer [22]. The CGS layers all have the same spatial resolution but differ from each other in terms of the PSNR quality. Each CGS

layer has a specific value of DID and QID=0. For CGS, the same inter-layer prediction mechanisms employed for spatial scalable coding are used, except that the corresponding up-sampling operations and the inter-layer de-blocking for intra-coded reference layer macro-blocks are neglected. The inter-layer intra and residual prediction are directly performed in the transform domain to reduce the decoding complexity. In every layer, quality refinement of the transform coefficients are stored by using a decreasing quantization step size [10]. Though a CGS bit-stream can have seven enhancement layers, the inter-layer prediction to obtain a sub bit-stream is limited to any three layers, out of which one has to be the base layer.

MGS is a modification of the coarse grain scalability. To increase the granularity, SVC provides quality refinement layers. The different MGS layers have the same dependency identifier DID, but have different quality identifiers QID. So MGS layers represent the different quality layers inside a particular dependency layer MGS. Switching between different dependency layers can only take place at IDR access units in the case of CGS, but there is no such restriction for switching between different quality refinement layers in the case of MGS. In the case of MGS, each dependency layer can contain one or more quality levels, each identified by a quality identifier Q. MGS allows the use of up to 16 quality levels per each dependency layer [22].

Chapter 4

SIMULATION MODEL FOR TRANSPORT OF VIDEO DATA OVER ERROR PRONE CHANNELS

4.1 Introduction

A simulation model for transmission of single layer SVC video data over an error prone channel is proposed in this research study. The aim is to compare the performance enhancement provided by applying unequal error protection and equal error protection to the video bit stream compared to the scenario where no error protection is applied. The video is first encoded using the JSVM reference software version 9.15 [11] and error protection is applied to the video data using Reed Solomon (RS) codes. The encoded data is interleaved using a row-column interleaver and transmitted through a communication channel. The received data is passed through a deinterleaver, followed by RS decoding. The NAL units are reconstructed from the decoded data to form the erroneous bit stream. The block diagram of the model is given in figure 4.1. In the following sections, we look into the detail each of the blocks in the model.

4.2 Encoding of the video sequence

Three video sequences namely SONY DEMO (17680 frames), STAR WARS (54000 frames) and SILENCE OF THE LAMBS (24000 frames) are encoded at a rate of 30 frames per second using JSVM reference software version 9.15 [11] in this study. The videos are encoded at CIF resolution (352×288 pixels) with different GOP structure of G16B0, G16B3 and G16B7. The encoder settings used

in this experiment are based on the settings used in [10] for single layer SVC experiments. These settings have shown to achieve good RD performance [10]. QP value of 30 is chosen for this experiment. The reason behind this choice is that the PSNR obtained is in the range of 35-40 dB which corresponds to good quality video [29]. The encoded video is passed through a Network Abstraction Layer Unit (NALU) header parser to obtain the header information of individual NAL units [1]. The individual NALUs are extracted using the NALU parser in byte-stream format.

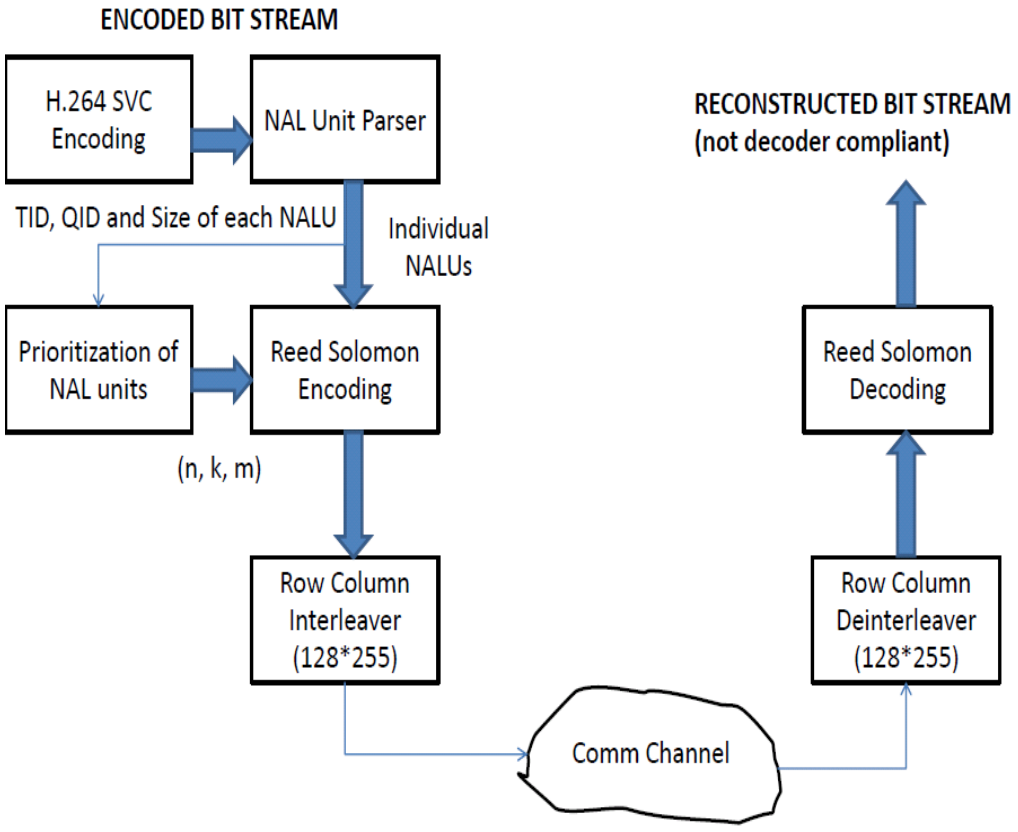


Figure 4.1: Proposed simulation model for transport of video over error prone channels

4.2.1. Video Sequences

We consider video sequences of CIF resolution (352×288) in this work. We consider 3 CIF video sequences in this work.

- Ten-minute Sony Digital Video Camera Recorder demo sequence (17,680 frames at 30 frames/sec), which we refer to as Sony Demo sequence, genre complex texture and range of motion activity
- First half hour of the Star Wars IV movie (54,000 frames at 30 frames/sec), genre science fiction/action
- First fifteen minutes of Silence of the Lambs movie (24,000 frames at 30 frames/sec). genre drama/thriller

The Sony Demo sequence is originally a high definition (HD) video sequence with resolution (1280×720) that has been down sampled to CIF (352×288) resolution. It consists of scenes with complex texture and a wide range of low to high motion activity, so encoding complexity is higher compared to that for other sequences. Hence the encodings for the Sony Demo sequence required more time and effort as a result of the higher use of the motion compensation prediction tools.

Star Wars IV and Silence of the Lambs movie sequences, on the other hand are not that complex to encode, compared to Sony Demo sequence. They have short bursts of high motion and high activity amidst relatively quieter scenes without too much motion.

4.2.2 Encoding tools

All the long videos are originally from high quality DVD source material, except Sony Demo which is a demo sequence decoded from a high definition capture camera where the original source material is in High Definition (HD) resolution, which is then decoded and down sampled to CIF resolution using MEncoder tool (http://www.mplayer.hu) as with other sequences which are originally in CIF. The JSVM H.264 SVC reference encoder (version 9.15) is used for all the experiments [11]. Some of the important parameters are discussed in brief here.

Configuration files are used to change the parameters for video encoding. There are two types of configuration files namely main configuration file and layer configuration file. The layer configuration file is present for individual layers. This research study deals with only single layer encodings, so there exists only one layer configuration file. The configuration files represent a collection of configuration parameters, which can be changed according to the needs of the user. In the event of not changing a particular parameter in a configuration file, the default values are used automatically for those parameters. Output file specifies the file to which the encoded bit stream is to be written. Frame rate specified the maximum rate in Hertz (Hz). The SNR enhancements are set CGS mode by specifying "CgsSnrRefinement" to 0. "PreAndSuffixUnitEnable" specifies whether to add prefix NAL units before the NAL units of AVC slices. When this parameter is 1, prefix NAL units are added. This parameter is always on in scalable video coding.

"GOPSize" parameter specifies the number of hierarchical B frames plus one key picture, either of type I or P. The GOPSize is varied in this study. We use GOPSize values of 1, 4 and 8. "IntraPeriod" specifies the number of frames between 2 I frames. It is set to 16 in this study. "NumLayers" specifies the number of layers inclusive of the base layer. In this study, since we deal only with single layer SVC, the NumLayers is set to 1 always. For each layer a layer configuration file shall be specified by using the parameter "LayerCfg". The macro block adaptive inter-layer prediction has been used. As suggested in the JSVM software manual [11], this mechanism uses an R-D optimization framework. Using this, significant RD performance improvement has been observed, at the price of an increased encoding time. Also the values of MeQP are set to values just smaller than the QP values as suggested in the manual for an improved RD performance. This parameter is set to 6 less than the QP value specified. The MeQP values are used for determining the Lagrangian parameters for motion estimation and mode decision of key pictures. CABAC coding scheme is used and the (8 × 8) transform is enabled for a better RD performance. The deblocking filter is applied to all block edges with the exception of slice boundaries, which marginally improved the RD performance. Fast Search and Fast bi-directional search are employed. Sum of absolute difference (SAD) for the luminance component is used as the distortion measure, which is applied for the motion search on integer-sample positions. SAD is used in the Hadamard transform domain for the luminance component as the distortion measure for motion search on sub-sample positions. These have been slightly changed

depending on the video sequence used. The Search range has also been changed according to the video sequence to achieve a good trade-off between RD performance and the encoding time. In the layer configuration file, source width and source height are set to CIF resolution (352 × 288).

4.2.3 GOP structures

Three different GOP structures, namely IPPPPPPPPPPPPPPI (16 frames, with 0 B frame per I/P frame), which is denoted by G16-B0, IBBBPBBBPBBBPBBBI (16 frames, with 3 B frames per I/P frame) denoted by G16-B3 and IBBBBBBBPBBBBBBBI (16 frames, with 7 B frames per I/P frame) denoted by G16-B7 are used in our experiments. In the context of H.264 SVC, these three GOP structures are defined by their "GOP size" which is the number of hierarchical B frames plus one key picture, either of type I or P. Hence, G16-B0 has a GOP size 1, G16-B3 has a GOP size 4 and G16-B7 has a GOP size 8.

4.2.4 Video Traffic Metrics

Here a brief overview of the essential video traffic metrics calculated during the statistical analysis is provided [29]. Let us assume that a video sequence consisting of M frames encoded with a fixed quantization parameter (QP). Let X_m (m = 1. . . M) denotes the sizes [in bits] of each encoded video frame.

The mean frame size $\bar{X}$ [bits] of the encoded video sequence is defined as

$$\bar{X} = \frac{1}{M}\sum_{m=1}^{M} X_m \qquad (4.1)$$

.

If each video frame is transmitted during one frame period T (e.g., 33 ms for 30 frames per second), then the bit rate R_m [bits per second] required to transmit frame X_m is given by,

$$R_m = \frac{X_m}{T} \qquad (4.2)$$

The average bit rate $\bar{R}$ [bits per second] is given by

$$\bar{R} = \frac{\bar{X}}{T} \qquad (4.3)$$

The Peak Signal-to-Noise Ratio (PSNR) is used as the objective measure of the quality of a reconstructed video frame R (x, y) with respect to the uncompressed video frame F (x, y). The larger the difference between R (x, y) and F (x, y), or equivalently, the lower the PSNR value. The PSNR is expressed in decibels [dB] to accommodate the logarithmic sensitivity of the human visual system. The PSNR is typically obtained for the luminance video frame and in case of a $N_x \times N_y$ frame consisting of 8-bit pixel values; it is computed as a function of the mean squared error (MSE) as [29]

$$MSE = \frac{1}{N_x \times N_y} \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} [F(x,y) - R(x,y)]^2 \qquad (4.4)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \qquad (4.5)$$

Let PSNR quality of m^{th} video frame m is denoted by Q_m and the average PSNR quality of a video sequence, denoted by $\bar{Q}$ of a video sequence is given by

$$\bar{Q} = \frac{1}{M} \sum_{m=1}^{M} Q_m \qquad (4.6)$$

4.3 Prioritization of NAL units

In scalable video coding, some parts of the video data are more important than other parts based on the type of frame (I, P or B frame), temporal ID and quality ID (in case of MGS encoding) and dependency ID (in case of CGS encoding). Quantitative comparison of two error protection schemes is performed in this study namely unequal error protection and equal error protection.

4.3.1 Unequal error protection

The idea in unequal error protection is to assign more weight to more important NAL units and lesser weight to less important NAL units. Based on these weights, the more important NAL units are assigned more number of parity bytes when compared to NAL units of lesser importance. In unequal error protection, the weight of the NAL unit is calculated as given in [3].

$$pr_{NALU} = TID_{NALU} \times (QID_{max} + 1) + QID_{NALU} \quad (4.7)\ [3]$$

where pr_{NALU} represents the priority of the NAL unit, TID_{NALU} represents the TID of the NAL unit, QID_{NALU} represents the QID of the NAL unit and QID_{max} represents the maximum value of QID in the entire bit stream. In the case of CGS encoded video,LID_{NALU} and LID_{max} could be used instead of QID_{NALU} and QID_{max} to obtain different priority levels. For temporal scalability, the QID_{NALU} is always 0.

Each base layer frame has a prefix NAL unit and a data NAL unit. The prefix NAL unit is typically 8-10 bytes in size and data NAL unit is a few hundred

bytes. The prefix NAL unit and the corresponding data NAL unit can be combined to form a single logical NAL unit [30]. From the priority information, the length of the codeword (n) is computed using the equation given below,

$$\frac{k}{n} = (1 - p_{max}^{loss}) + \left(\frac{pr_{NALU}}{pr_{MAX}}\right) \times (p_{max}^{loss} - p_{avg}^{loss}) \qquad (4.8) \qquad [3]$$

The k/n rate allocation used here is based on the formulae given in [3]. Here k represents the length of the original data sequence, pr_{NALU} represents the priority of the NALU, p_{max}^{loss} represent the maximum bit error rate of the channel, p_{avg}^{loss} represents the average bit error rate of the channel and pr_{MAX} represents the maximum value assigned to pr in the whole bit stream.

By using this scheme, more important NALUs will have a lower pr_{NALU} value, and so a higher priority. This means that the more important NALUs will be strongly encoded with more parity bytes compared to lesser significant NALUs. The values of p_{max}^{loss} and p_{avg}^{loss} set the overhead added to the bit stream in terms of parity bytes.

The overhead added to the video traffic in terms of parity bytes is computed using the formula,

$$Overhead\ (in\ \%) = \frac{Total\ number\ of\ parity\ bytes}{Total\ length\ of\ the\ bit\ stream\ (in\ bytes)} \times 100 \qquad (4.9)$$

More details on the experimental results are given in chapter 6 .The aim of the error protection scheme is to keep the overhead added to the bit rate, due to channel coding , to be minimal and at the same time provide reasonable

performance in terms of error correction. Chapter 5 explains the need for an unequal error protection scheme.

4.3.2 Equal error protection

We compare the performance of unequal error protection scheme described in the above section with equal error protection. The goal is to obtain a scheme that offers the least degradation in PSNR value when the channel BER is kept constant. For comparing the error protection schemes, the overhead added to the video traffic in terms of parity bytes is kept constant.

Based on the percentage of overhead added to the original bit stream calculated using equation , the number of parity bytes added to each logical NAL unit is given by

$$n_{NALU} = k_{NALU} + \left(\frac{Overhead\ (in\ \%)}{100}\right) \times k_{NALU} \qquad (4.10)$$

For example, if an overhead of 0.75% is added to the bit stream in the case of unequal error protection, then the length of the code word (n) for each NAL unit is given by, $n_{NALU} = 1.0075 \times k_{NALU}$.

4.4 Reed Solomon Encoding

Reed Solomon codes [8, 31] are a class of linear block codes with parameters (n, k) where n represents the length of the codeword and k represents the length of the original message. For a Reed-Solomon code of n symbols, the first k symbols is the data part, which is the information to be protected against corruption, and

the last (n-k) symbols is the parity part, which is calculated based on the data part. The number of parity symbols, (n-k), is always a multiple of 2, and is denoted by 2t. A Reed-Solomon codeword with 2t parity symbols has the capability of correcting up to t errors in the codeword. A large value of t means that a large number of errors can be corrected but requires more computational power than a small value of t. The length of the codeword (n) needs to be less than $2^m - 1$ where m represents the maximum number of bits that can be used to represent the symbol. The reason for choosing Reed Solomon codes for error protection is because DVB-T and DVB-H standard uses Reed-Solomon code for physical layer protection of MPEG-2 transport packets [31].

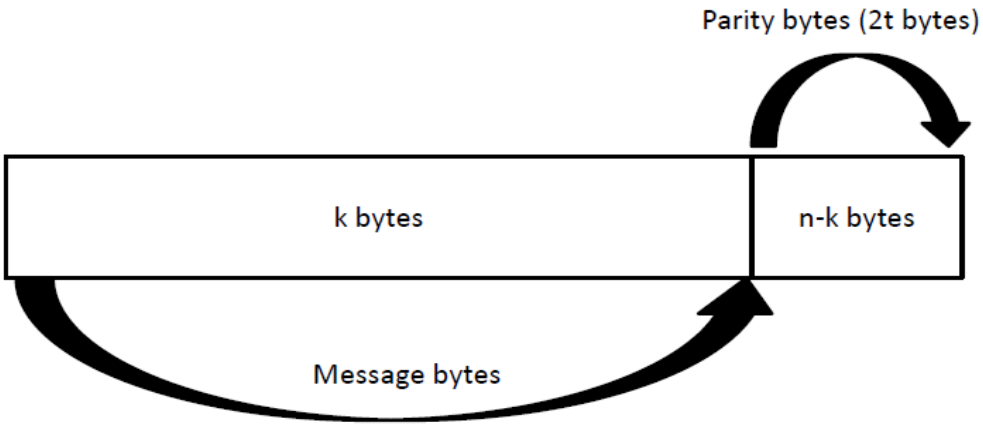


Figure 4.2: Reed Solomon encoding

The k bytes of the original message to be encoded as one block can be represented by a polynomial M(x) of order (k – 1) given by,

$$M(x) = M_{k-1}x^{k-1} + M_{k-2}x^{k-2} + \ldots + M_1x^1 + M_0 \qquad (4.11)$$

where each coefficient $M_{k-1}, M_{k-2}, \ldots, M_1, M_0$ are m-bit message symbols. To encode the message, the message polynomial is first multiplied by x^{n-k} and the

result is divided by the generator polynomial, g(x). Division by g(x) produces a quotient q(x) and a remainder r(x), where r(x) is of degree up to n – k – 1.

The remainder polynomial r(x) is appended to the message polynomial M(x) to form the codeword polynomial C(x) given by,

$$C(x) = M(x) + \frac{r(x)}{g(x)} \qquad (4.12)$$

Adding the remainder, r(x) to M(x) ensures that the encoded message polynomial will always be divisible by the generator polynomial without any remainder. If the division of the encoded message polynomial by the generator polynomial at the decoder side gives a remainder, then that indicates the presence of error in the codeword. Upto $\frac{n-k}{2}$ errors can be corrected using RS codes.

4.5 Interleaving

A row-column interleaver is used to improve the performance of Reed Solomon codes. In communication channels, burst errors occur more frequently than individual bit errors. More specifically in wireless communication channels, the transmission errors due to the wireless channel impairment are highly correlated due to multipath signal fading and shadowing effects. These channel impairments combined with the impulse interferences causes burst error patterns.

Interleaving is done to protect the data from random burst errors. If the number of errors exceeds the error correcting capability of the error correcting code, then the

original codeword is not recovered. The advantage of using an interleaver is that random burst errors do not affect whole parts of a single encoded unit but small parts of different encoded units. The burst errors in the interleaved data are averaged out and uniformly distributed after deinterleaving; thereby the effectiveness of using FEC is enhanced. 128 x 255 row-column interleaver is used in these experiments. There is a trade-off in using a particular buffer size. If the number of rows is increased, it takes more time to fill up the buffer but amounts to having a greater protection. The 128 x 255 interleaver is used because packets with a payload of 128 bytes are well suited for wireless communication when Real-time Transport Protocol (RTP) is used for encapsulating the packets [15]. Compressed video streams typically exhibit Variable Bit Rate (VBR) but constant quality. In this experimental setup, the encoded units are consecutively written row wise into the interleaver. Once the buffer is full, the packets are read column wise out of the buffer and transmitted through the channel. Once the buffer is empty, the next sets of encoded units are written row wise into the interleaver. In our experiment, we assume that the buffer is always full by padding additional zeros at the end of the buffer in case it is partially full.

4.6 Transmission of packets and deinterleaving

The packets of size 128 bytes are transmitted through the channel. Simulations are performed for Bit Error Rates (BER) ranging from 10^{-2} to 10^{-5}. Errors are introduced into the channel by corrupting pseudorandom bytes of the packets transmitted. For example, to obtain a BER of 10^{-3}, one-thousandths of the total

number of bytes transmitted through the channel are corrupted. The corrupted data packets are buffered and deinterleaved. Deinterleaving is the reverse process of interleaving. In deinterleaving, the received data packets are written column wise into a 128 x 255 buffer until it is full and the data is read row wise from the buffer to form the individual logical units.

4.7 Reed Solomon Decoding and reconstruction of erroneous bit stream

RS decoding is performed using Berlekamp - Massey algorithm [5], which is employed in the rsdec ( ) in-built function in MATLAB. The RS decoder aims to correct the errors in the given packets and form the original code word. For more information regarding the Berlekamp-Massey algorithm please refer [5, 8]. If the number of errors is greater than $\frac{n-k}{2}$, the RS decoder fails to decode the packet correctly. This leads to an error in a NAL unit. The individual NAL units are then reconstructed by splitting the prefix NAL unit and the data NAL unit from the decoded message. The reconstructed video sequence is not decoder-ready due to the presence of errors in the received bit stream. H.264/ SVC decoding tool specified in the JSVM reference software in its current version 9.15 [11] does not support decoding of a bit stream when NALUs are corrupt, arrive out of order [6]. To overcome this problem, we develop an approach based on [6] to reconstruct the bit stream such that it is decoder-compatible. Chapter 5 presents more details about this process.

4.8 Offset Distortion Approach

The video bit stream is the actual output of the video encoding and contains the entire video information in bytes. The advantage of using bit streams is that it allows for networking experiments where the quality of the video, after passing through the network, can be visually evaluated. However, the disadvantages of using the actual video bit stream are they are large in size; usually copyright protected and/or proprietary and so cannot be shared among the research groups [34, 35, 13]. Video traces provide a good alternative to the actual video bit stream. Video Traces are typically in simple text format and carry only the video frame sizes and the video frame qualities. In contrast to encoded video data, video traces do not carry the actual video information and are therefore exchangeable among researchers without copyright issues.

In this study, we test our model with the video trace approach. In case of the frame loss, offset distortion tool [13], is used to calculate the PSNR of the missed frames. For more details on the offset distortion approach to estimate the PSNR of lost frames, please refer [13, 35]. In this approach, the trace information is generated as shown in Figure 5.1. The trace file contains information about the NALU sizes of the bit stream. Using that information, random bytes of data are generated to resemble dummy NAL units. Once the dummy NAL units are generated, the steps explained in the above sections are followed. The individual NAL units are provided protection using Reed Solomon codes and then interleaved to form packets of data. Here the aim is to compare how accurately

the offset distortion traces can calculate the PSNR metric. In order to compare the bit stream approach to the offset distortion approach, the errors are injected at the same places where they were introduced in the bit stream approach. The received packets are deinterleaved and RS decoding is performed. The reconstructed dummy NAL units are thus obtained. The received NAL units are compared with the dummy NAL units that were generated initially to detect the NAL units in error. Details on the process of obtaining the PSNR based on the offset distortion approach are further explained in Chapter 5.

Chapter 5

RECONSTRUCTION OF H.264 SVC COMPATIBLE BIT STREAM

5.1 Received trace file generation

Bit stream extractor tool in the JSVM reference software version 9.15 [11] is used to generate a trace file. A sample version of the trace file is as shown in figure 5.1. The trace file contains information regarding the size of NAL unit, the TID, LID and QID of the NAL unit, type of NALU.

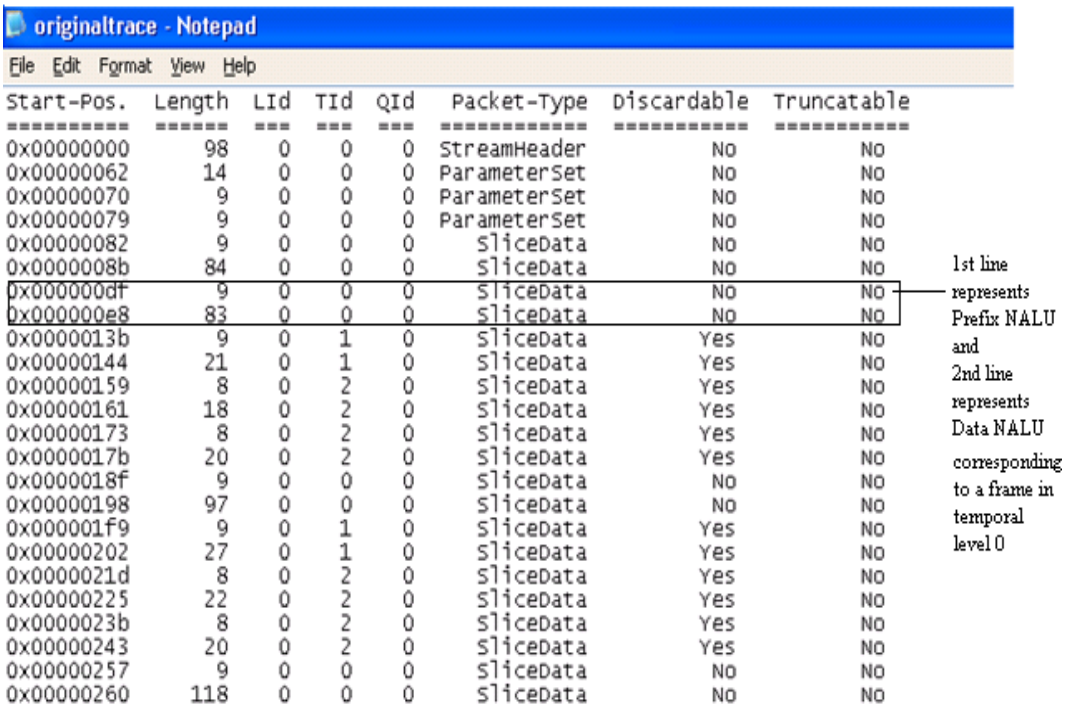


Figure 5.1: Original trace file showing length, LID, TID and QID of NAL units

A modified version of the trace file is created by appending a column to the trace file indicating the frame number corresponding to each line in the trace file. By

doing this, parsing for the frames in error becomes easier. The original NAL unit is compared byte-by-byte to their corresponding reconstructed NAL unit to check for errors. In its current version, the JSVM software version 9.15, considers that the NAL unit is in error even if one byte of data in the NAL unit is corrupted. In the first pass, if either the prefix NAL unit or the data NAL unit corresponding to a particular frame is in error, both the prefix and data NAL units corresponding to that frame need to be discarded. For example, if the data NAL unit of frame 3 is in error, in the first pass, the lines corresponding to frame 3 in the trace file (both the prefix and data NAL unit corresponding to frame 3) is deleted as shown in the figure 5.3.

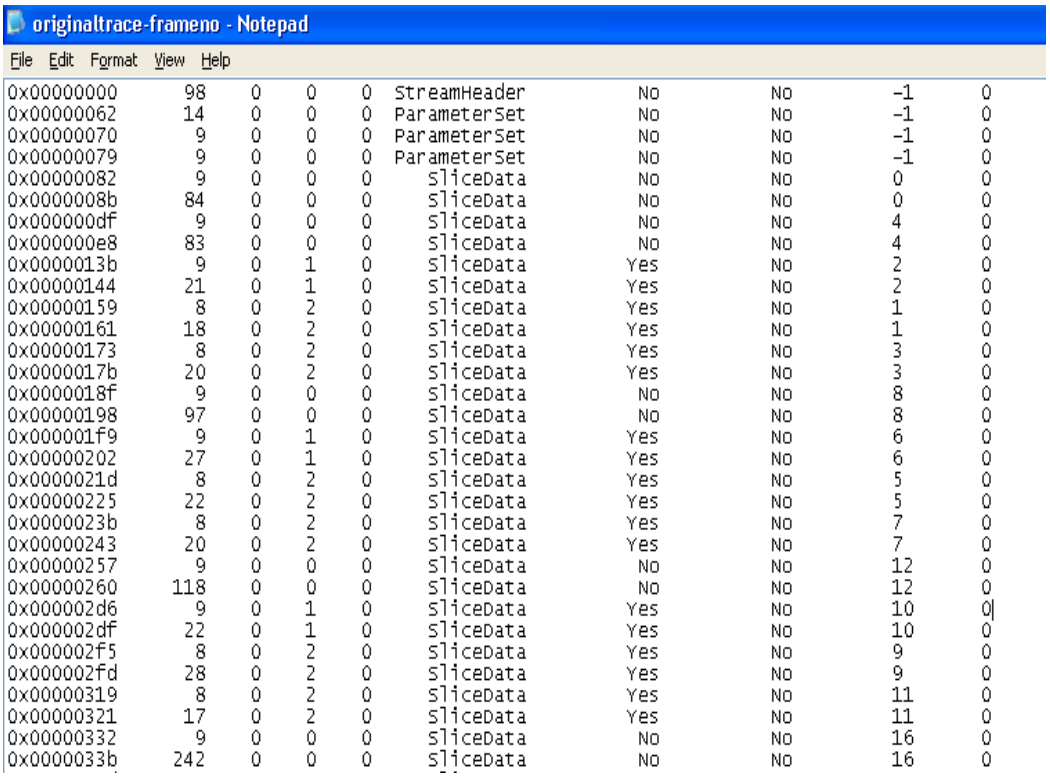
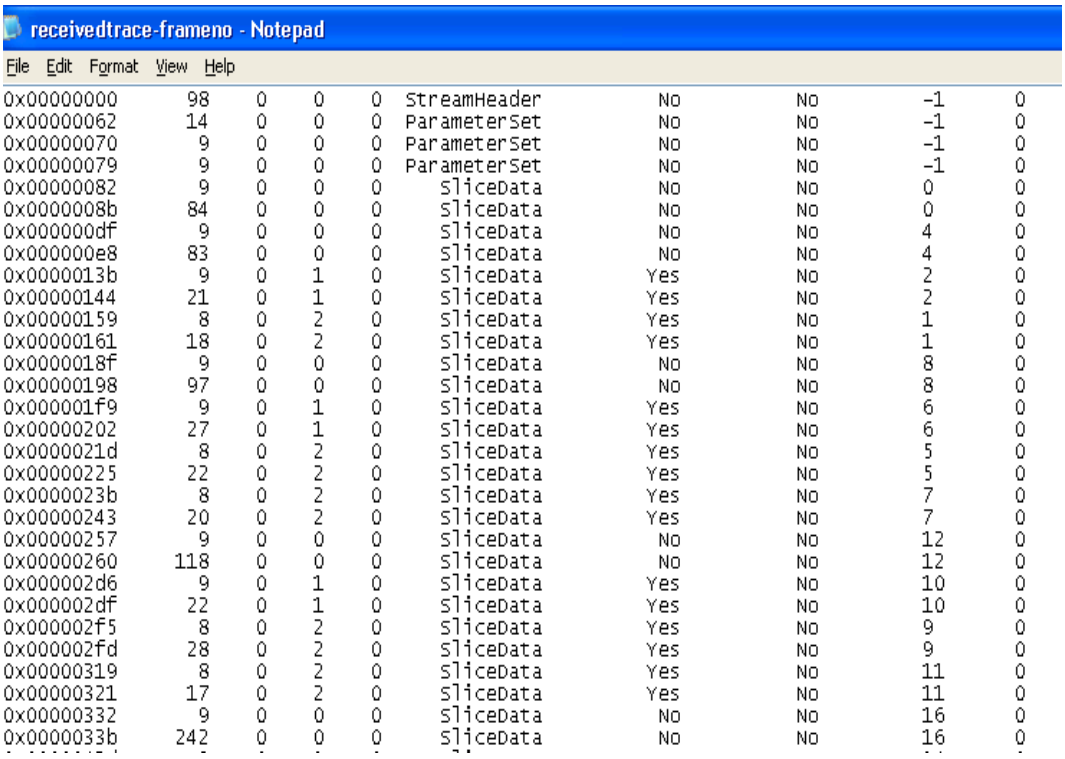


Figure 5.2: Trace file with frame numbers before transmission


receivedtrace-frameno - Notepad

File Edit Format View Help

0x00000000	98	0	0	0	StreamHeader	No	No	-1	0
0x00000062	14	0	0	0	ParameterSet	No	No	-1	0
0x00000070	9	0	0	0	ParameterSet	No	No	-1	0
0x00000079	9	0	0	0	ParameterSet	No	No	-1	0
0x00000082	9	0	0	0	SliceData	No	No	0	0
0x0000008b	84	0	0	0	SliceData	No	No	0	0
0x000000df	9	0	0	0	SliceData	No	No	4	0
0x000000e8	83	0	0	0	SliceData	No	No	4	0
0x0000013b	9	0	1	0	SliceData	Yes	No	2	0
0x00000144	21	0	1	0	SliceData	Yes	No	2	0
0x00000159	8	0	2	0	SliceData	Yes	No	1	0
0x00000161	18	0	2	0	SliceData	Yes	No	1	0
0x0000018f	9	0	0	0	SliceData	No	No	8	0
0x00000198	97	0	0	0	SliceData	No	No	8	0
0x000001f9	9	0	1	0	SliceData	Yes	No	6	0
0x00000202	27	0	1	0	SliceData	Yes	No	6	0
0x0000021d	8	0	2	0	SliceData	Yes	No	5	0
0x00000225	22	0	2	0	SliceData	Yes	No	5	0
0x0000023b	8	0	2	0	SliceData	Yes	No	7	0
0x00000243	20	0	2	0	SliceData	Yes	No	7	0
0x00000257	9	0	0	0	SliceData	No	No	12	0
0x00000260	118	0	0	0	SliceData	No	No	12	0
0x000002d6	9	0	1	0	SliceData	Yes	No	10	0
0x000002df	22	0	1	0	SliceData	Yes	No	10	0
0x000002f5	8	0	2	0	SliceData	Yes	No	9	0
0x000002fd	28	0	2	0	SliceData	Yes	No	9	0
0x00000319	8	0	2	0	SliceData	Yes	No	11	0
0x00000321	17	0	2	0	SliceData	Yes	No	11	0
0x00000332	9	0	0	0	SliceData	No	No	16	0
0x0000033b	242	0	0	0	SliceData	No	No	16	0

Figure 5.3: Received trace file with frame 3 missing

In H.264/SVC, based on the GOP structure, the P and B frames are predicted from either I, P or B frames. So, if a NAL unit corresponding to a frame is in error, we need to eliminate the NAL units corresponding to the frames that are dependent on the frames in error for decoding purposes from the bit stream. We call this the filtered bit stream. In the second pass, the NALUs dependent on the NAL units in error is discarded from the bit stream. The NAL unit dependency is based on which GOP structure is being used for the encoding process. In the experiments conducted, G16B7, G16B3 and G16B0 are used, so we take a closer look into the NAL unit dependencies for GOP structures G16B7, G16B3 and G16B0.

5.2 G16B0 GOP structure

A G16B0 GOP structure is shown in figure 5.4. The arrows indicate the prediction structure. In a G16B0 GOP structure there are 15 P frames between 2 I frames (IPPPPPPPPPPPPPPI). All the frames are at temporal level 0 as indicated in the figure 5.4.

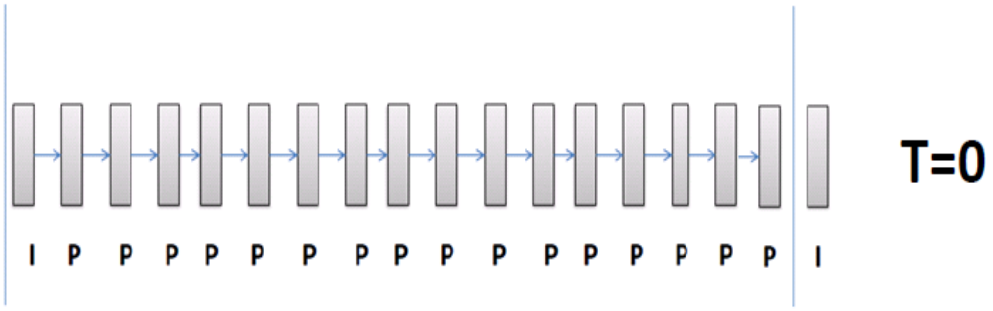


Figure 5.4: GOP Structure of G16B0

The first P frame is predicted from the starting I frame, the second P frame is predicted from the first P frame, the third P frame from the second P frame and so on.

- If any of the NAL unit corresponding to the I frame is in error, then all NAL units corresponding to the entire GOP is discarded.
- If any of the NAL unit corresponding to one of the P frames is in error, say the data NAL unit of P1 frame in a particular GOP is in error, the NAL units corresponding to all the P frames, occurring after the P frame in error, of the entire GOP are discarded. So NAL units corresponding to frames P2,P3…P15 are discarded in this case

5.3 G16B3 GOP structure

A G16B3 GOP structure is shown in figure 5.5. The arrows indicate the prediction structure. In a G16B3 GOP structure there are 3 B frames between 2 I/P frames (IBBBPBBBPBBBPBBBI). This GOP structure has 3 temporal levels as seen in the figure. The I and P frames are at temporal level 0 and the B frames are at temporal levels 1 and 2. The NALU dependencies for a G16B3 are explained below.

- If any of the NAL unit corresponding to an I frame is in error, the NAL units corresponding to the remaining frames (P and B frames) in the entire GOP are discarded.
- If any of the NAL units corresponding to P4 frame is in error, the NAL units corresponding to the remaining P and B frames in the GOP are discarded. Only the NAL units corresponding to the I frame of that particular GOP remains in the filtered bit stream.

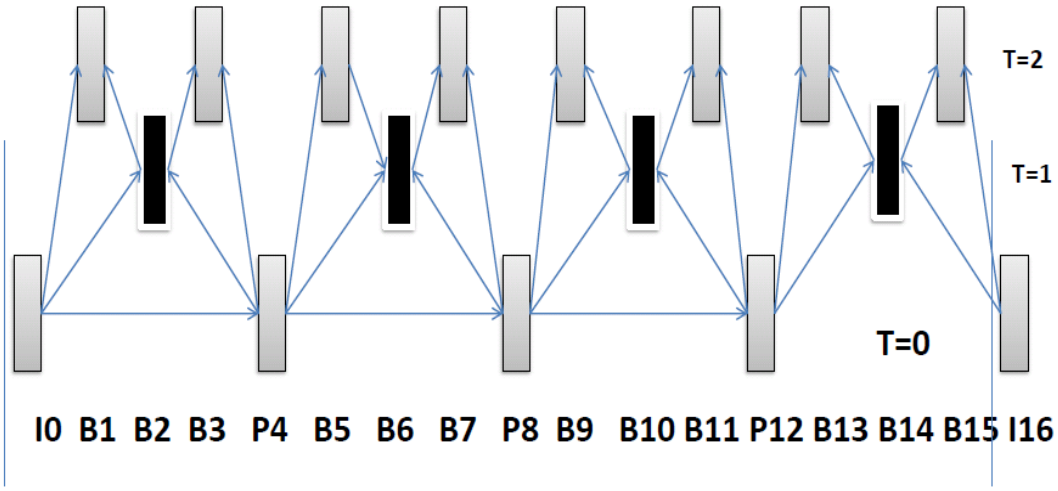


Figure 5.5: GOP Structure of G16B3

- If any of the NAL units corresponding to a P8 frame is in error, the NAL units corresponding to P12 frame and B5, B6, B7, B9, B10, B11, B13, B14, B15 frames of that particular GOP are discarded.
- If any of the NAL units corresponding to a P12 frame is in error, the NAL units corresponding to B9, B10, B11, B13, B14 and B15 frames in that particular GOP are discarded.
- If one of the NAL units corresponding to a B frame at temporal level 1 is in error, the NAL units corresponding to the previous B frame and the next B frame in the GOP are discarded. Suppose, the data NAL unit corresponding to B6 frame in a particular GOP is in error, the NAL units corresponding to B5 and B7 frames are discarded.
- If a NAL unit corresponding to a B frame at temporal level 2 is in error, no other NAL unit is discarded other than the ones discarded at the first pass. This is because the B frame at the highest temporal level is not used to predict any other frame in the GOP.

5.4 G16B7 GOP structure

A G16B7 GOP structure is shown in figure 5.6. The arrows indicate the prediction structure. In a G16B7 GOP structure there are 7 B frames between 2 I/P frames (IBBBBBBBPBBBBBBBI). This GOP structure has 4 temporal levels as seen in the figure 5.6. The I and P frames are at temporal level 0 and the B frames are at temporal levels 1, 2 and 3.

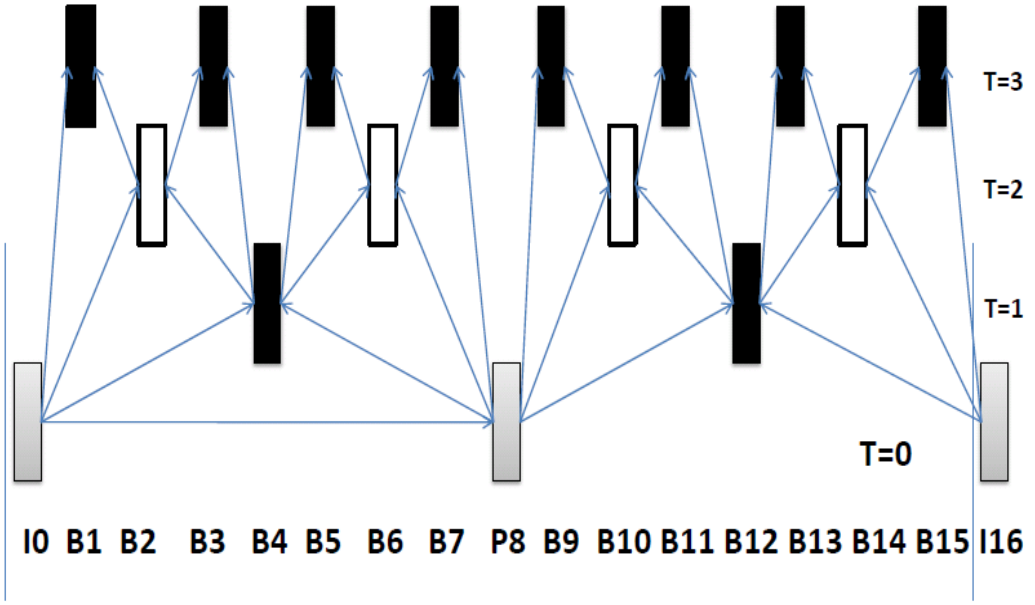


Figure 5.6: GOP structure of G16B7

The NALU dependencies for a G16B7 GOP structure are explained below.

- If any of the NAL unit corresponding to an I frame is in error, the NAL units corresponding to the remaining frames (P and B frames) in the entire GOP are discarded.
- If any of the NAL units corresponding to the P frame is in error, the NAL units corresponding to all the B frames in the GOP are discarded. Only the NAL units corresponding to the I frame of that particular GOP remains in the filtered bit stream.
- If one of the NAL units corresponding to a B frame at temporal level 1 is in error, the NAL units corresponding to the preceding three B frame and succeeding three B frames in the GOP are dropped. Suppose, the data

NAL unit corresponding to B4 frame in a particular GOP is in error, the NAL units corresponding to B1, B2, B3 and B5, B6, B7 frames are discarded.

- If one of the NAL units corresponding to a B frame at temporal level 2 is in error, the NAL units corresponding to the previous B frame and the next B frame in the GOP are discarded. Suppose, the data NAL unit corresponding to B6 frame in a particular GOP is in error, the NAL units corresponding to B5 and B7 frames are discarded.
- If a NAL unit corresponding to a B frame at temporal level 3 is in error, no other NAL unit is discarded other than the ones discarded at the first pass. This is because the B frame at the highest temporal level is not used to predict any other frame in the GOP.

The dependency structures of different GOPs explained above is the main reason behind using unequal error protection for the NAL units. Assigning more parity bytes to an I frame or a P frame, ensures that an I or a P frame is lost less frequently compared to the other frames in the GOP. When one byte of an I frame is in error , the entire GOP is dropped (16 frames) whereas when a byte of a B frame in the highest temporal level is lost, no other frame is discarded.

5.5 Reconstruction of YUV sequence using frame copy

The NAL units in error are first identified by comparing the reconstructed NAL units with the NAL units of the original bit stream. The lines corresponding to the frames in error are deleted from the trace file in the first pass and the received trace is obtained. In the second pass, the dependent NAL units, corresponding to the NAL units in error, are filtered out from the received trace file to obtain the filtered trace file.

The filtered bit stream is constructed with the aid of the filtered trace information using the Bit Stream Extractor tool present in the JSVM reference software version 9.15 [11]. The filtered bit stream is H.264/SVC compatible and can be passed to the H.264/SVC decoder. The filtered bit stream is decoded using the H264AVCDecoderTestLibStatic tool present in the JSVM reference software version 9.15 [11]. Thus, the filtered YUV sequence is obtained. The metric used to calculate the objective video quality is termed as peak signal to noise ratio (PSNR) between the unencoded original YUV video data and the encoded and subsequently decoded YUV video data as explained in Chapter 4. The filtered YUV sequence cannot be directly used for the PSNR computations due to the presence of dropped frames in the filtered sequence. For example, if frame 3 is in error, then the decoded picture corresponding to frame 3 would be missing from the filtered YUV sequence. This leads to comparison of frame 3 of the original sequence with Frame 4 of the filtered sequence and Frame 4 of the original sequence with Frame 5 of the reconstructed sequence and so on, which is not the

correct approach. Error concealment is done to conceal the effect of missing coded pictures in the filtered bit stream. For more details regarding error concealment strategies please refer [32], [33]. Frame Copy approach [32] is adopted to account for the missing coded pictures due to transmission errors. In the frame copy method, each pixel value of the concealed frame is copied from the corresponding pixel of the first frame in Reference Picture List 0 (RefPicList0). In effect, the last successfully played out frame is used to conceal the missing frame until a new video frame is successfully decoded and displayed at the receiver. In the above example, since frame 3 is in error, frame 2 (which is the last played out successful frame) is copied in its place and PNSR measurements are taken. Offset distortion tool [13] is used to calculate the PSNR between the original and played out sequences because the PSNRStatic tool in JSVM reference software version 9.15 [11] is unable to produce the PSNR measurements for all frames in scenarios where frame losses occur.

5.6 Obtaining PSNR metric using offset distortion approach

The filtered trace which is a modified version of the original trace file which does not contain the lines corresponding to the NAL units in error and their dependent NAL units is obtained as explained in the previous sections. From the filtered trace information, the frames which are in error are obtained. The offset distortion tool requires 3 parameters – the original YUV file, the decoded YUV file and the offset value to generate the PSNR statistics for the missing frames. The offset value is set to the maximum number of consecutive frames that are lost. The

PSNR statistics are computed and from the missing frames information, the PSNR of the missing frame is calculated from the PSNR statistics of the last played out frame. A section of the output from the offset distortion tool is shown in Figure 5.7. The offset value used here is 8.

0	55.39896	54.61748	54.08897	51.07326	49.16773	46.31816	45.03572	43.78948	43.43282
1	56.19068	55.94249	53.59231	51.37575	48.08919	46.57088	45.13209	44.72219	42.88811
2	55.60474	54.9835	52.93828	49.51648	47.78176	46.19088	45.72788	43.72166	42.98586
3	55.28202	54.37882	51.54007	49.51042	47.69801	47.15855	44.90104	44.07394	42.75219
4	54.03458	53.86126	51.7576	49.73415	49.08583	46.49127	45.52645	44.00139	42.87682
5	54.79485	53.7017	52.0674	51.32192	48.47304	47.32976	45.53	44.21846	42.64067
6	53.8491	53.76598	53.14232	50.64973	49.32074	47.20489	45.66364	43.88981	43.05333
7	53.98202	53.65254	52.06719	50.69418	48.37541	46.66247	44.75534	43.84464	42.73967
8	53.34118	53.2778	52.01711	49.59031	47.7104	45.67786	44.68407	43.48332	42.17985
9	54.1727	53.64232	51.45515	49.37694	47.16434	46.0262	44.65544	43.17967	42.02292
10	53.97955	53.24652	51.3056	49.114	47.78746	46.18083	44.4548	43.15523	42.51392
11	53.45251	52.80015	51.60756	50.07923	48.16012	46.07898	44.59782	43.85665	42.9119
12	52.70135	53.7663	52.42882	50.28744	47.79439	46.11836	45.25976	44.17656	43.56871
13	54.4	53.71454	51.78554	49.06863	47.2769	46.32246	45.12775	44.44819	42.10047
14	54.27869	53.07132	50.32158	48.49064	47.43447	46.12384	45.35956	42.83258	42.68083
15	53.94869	52.07632	50.56337	49.36064	47.85821	46.92443	44.09412	43.92268	42.61925
16	52.80165	52.92108	51.84871	50.26561	49.0932	45.89106	45.68739	44.14466	42.83265
17	53.53871	53.24485	52.18228	50.87749	47.61658	47.37765	45.58915	44.06731	42.93175
18	52.97616	53.21616	52.15622	49.59349	49.31984	47.2531	45.46181	44.19132	43.31986
19	52.90935	52.28776	50.99411	50.71347	48.48706	46.48546	45.11965	44.16888	43.0515
20	51.82648	52.24988	51.98848	49.72777	47.52403	46.07217	45.03772	43.82358	42.87786
21	53.38793	53.24207	51.36257	48.94762	47.41568	46.25852	44.90031	43.84155	41.74588

Figure 5.7: Section of the PSNR output from Offset distortion tool

The first column denotes the frame number. The second column denotes the PSNR value of the frame if it was decoded properly and the consecutive columns contain the PSNR values for the missing frames. Suppose frame 3 and frame 4 are in error, frame 2 is played out instead of frame 3 and frame 4 and the PSNR of frame 3 and 4 are calculated from the PSNR values corresponding to frame 2. The PSNR of frame 3 is computed to be 3rd column corresponding to frame 2 (i.e. 54.9835) and the PSNR of frame 4 is computed to be the 4th column corresponding to frame 2 (52.93828) instead of the 2nd columns corresponding to

frames 3 and 4 respectively. Experiments are conducted for SONY G16B3 using offset distortion approach and the results are presented in Chapter 6.

Chapter 6

RESULTS AND CONCLUSION

We consider 3 long video sequences namely Sony Demo sequence, which contains 17680 frames, Star Wars movie sequence which contains 54000 video frames and Silence of the Lambs movie sequence which contains 24000 frames. We compare the performance of the system, in terms of PSNR and number of missed frames, in the presence of bit errors in this research study. In the first approach, unequal error protection is applied based on the importance of the NAL units. In the second approach, equal error protection is applied to the NAL units based on their size by keeping the overhead (in terms of parity bytes added) constant to that used in unequal error protection. The aim is to keep the overhead to be small and at the same time achieving reasonable performance. We quantify the improvement in terms of PSNR obtained in these two schemes compared to the case where no FEC being used.

6.1 Results for SONY DEMO sequence

6.1.1 SONY DEMO G16B3 sequence

For the SONY demo sequence, three schemes with different overheads (in terms of error correction bytes) are used. Here, the video is encoded using G16B3 GOP structures. The BER of the channel is fixed at 10^{-3}. The overhead used in the error protection scheme are suitable only for this BER range. When the BER is increased to 10^{-2}, the number of frames in error is huge (around 16000 frames out of the 17680 frames are dropped). So, in all these experiments, we consider

that the BER is fixed at 10^{-3}. The statistics for G16B3 sequence encoded at QP=30 is given in Table 6.1 and the distribution of the video traffic among the different types of frames namely T0, T1 and T2 frames is shown in figure 6.1.

Average Bit Rate (in Kbps)	381.8	
Average PSNR of decoded video(dB)	35.65575	
Type of frame and number of frames	Number of frames	Average frame size (in bytes)
T0 frames	4420	4154
T1 frames	4420	995
T2 frames	8840	622
Total	17680	1600

Table 6.1: Video traffic statistics of SONY DEMO G16B3 sequence (QP=30)

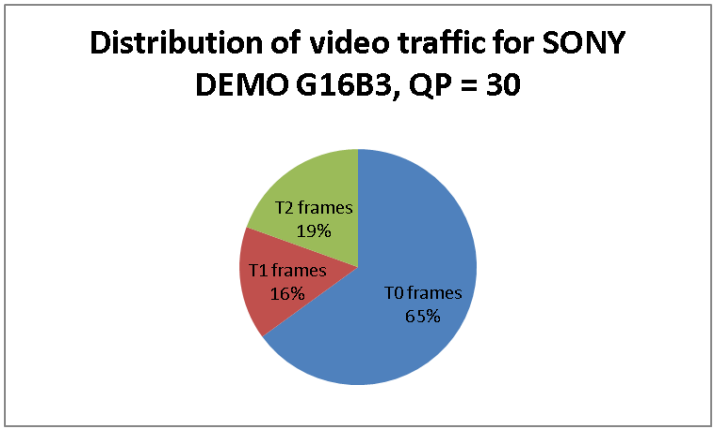


Figure 6.1: Distribution of video traffic for SONY DEMO G16B3 sequence (QP=30)

T0 frames comprising of I and P frames constitute 65% of the video traffic, T1 frames comprise of temporal level 1 B frames constitute 16% of the video traffic and T2 frames consisting of temporal level 2 B frames constitute 19% of the video traffic. I and P frames form the most important part of the traffic and are assigned higher protection in unequal error protection scheme. These frames are required for decoding other P frames and B frames as explained in Chapter 5. The next in the priority list are the B frames in temporal level 1 which aid in the decoding of temporal level 2 B frames. The least important frames are the B frames in temporal level 2 as they do not have any dependencies.

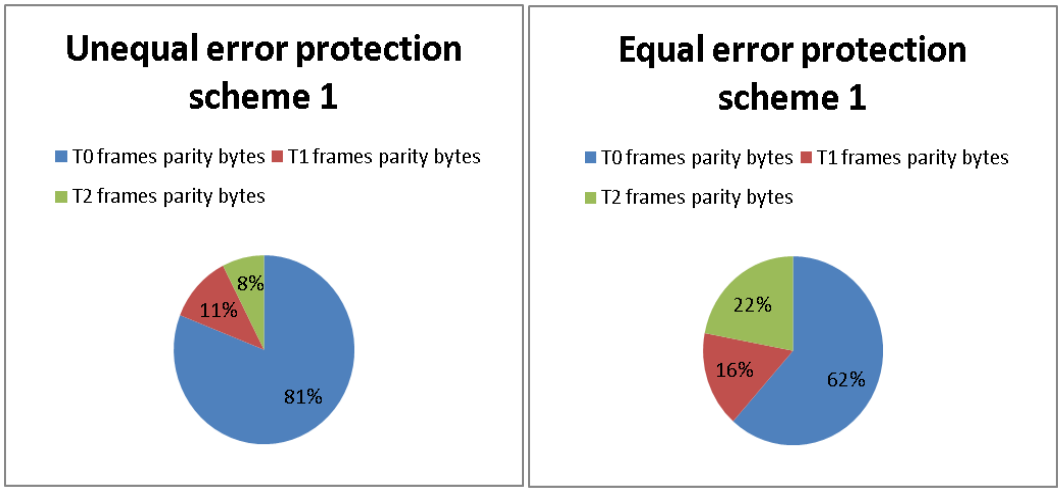


Figure 6.2: UEP and EEP schemes for SONY DEMO G16B3 with overhead of 0.80%

In the first scheme, the overhead added to the video traffic is constant for both UEP and EEP and is equal to 0.80% of the video traffic. In the UEP scheme, parity data is assigned to the NALUs, based on their level of importance as explained in chapter 4. The distribution of parity bytes assigned to T0, T1 and T2 frames is given in figure 6.2. For T0 frames, 81% of the total number of parity

bytes, assigned to the entire video traffic, is provided. Thus strong protection is given to these frames due to their higher importance. For T1 frames, 11% of the total parity bytes are assigned and for T2 frames 8% of the parity bytes are assigned. In the case of EEP, the parity bytes are assigned based on the length of the NAL units as explained in Chapter 4. The distribution of parity bytes assigned to T0, T1 and T2 frames in scheme 1 are 62%, 16% and 22% respectively.

Type of frame	Number of frames lost (UEP)	Number of frames lost (EEP)	Offset distortion approach (UEP)	Offset distortion approach (EEP)
T0	3	21		
T1	84	58		
T2	1290	214		
Total	1477	293	1477	293
PSNR of reconstructed video (dB)	35.962	36.528	35.961	36.528
Degradation in PSNR (dB)	0.694	0.128	0.693	0.128

Table 6.2: Frame loss and PSNR statistics for SONY DEMO G16B3 with overhead of 0.80%

From table 6.2, it is observed that the number of missing frames in the case of UEP is more compared to the case of EEP. In the case of UEP, total number of frames lost is 1477 (T0 frames lost=3, T1 frames lost =84, T2 frames lost=1290). In the case of EEP, total number of frames lost is 293 (T0 frames lost=21, T1 frames lost= 58, T2 frames lost=214).PSNR degradation in the case of UEP is greater than in the case of EEP due to larger number of dropped frames. An interesting observation is that even though a larger number of T0 frames are lost in the case of EEP due to lesser protection for T0 frames in EEP (62% of parity bytes) as compared to UEP (81% of parity bytes). The reason for more degradation, in terms of PSNR, in the case of UEP is due to lesser protection for the T2 frames (8%) as compared to EEP (22%). This observation is further validated in the second error protection scheme.

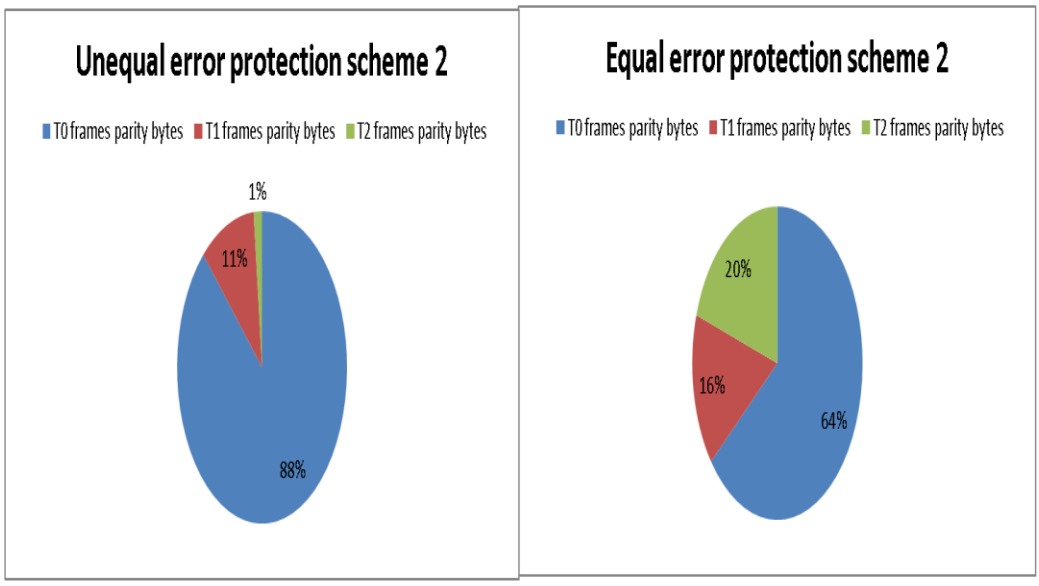


Figure 6.3: UEP and EEP schemes for SONY DEMO G16B3 with overhead of 3.91%

In the second scheme, the overhead is increased to 3.91% from 0.80%. This corresponds to approximately five-fold increase in the number of parity bytes. The distribution of parity bytes is as shown in figure 6.3

The distribution of parity bytes follows a distribution based on the number of bytes lost for each type of frame in error. From Table 6.1, the weight of the T0, T1 and T2 frames are calculated from the average frame sizes given by $Size_{T0} = 4154$, $Size_{T1} = 995$, $Size_{T2} = 622$. Based on this information and the NALU dependencies, the protection ratios are calculated as follows. For example, protection of a T1 frame is calculated as follows:

$$Wt_{T1} = (1 \times 995) + (2 \times 622) \qquad (6.1)$$

The weight arises from the fact that, if a T1 frame (B frame in T1) is in error, then the dependent frames are also dropped. (2 B frames in T2). Similarly the weights of T0 and T2 frames are calculated. Similarly the weights of T0 frame and T2 frame are computed to be $Wt_{T2} = 622\ bytes$ and $Wt_{T0} = 17662$. Based on these values, the protection of T1 is given by

$$Protection_{T1} = \frac{Wt_{T_1}}{Wt_{T_1} + Wt_{T_2} + Wt_{T0}} \qquad (6.2)$$

Similarly the protection ratios of T0 and T2 frames are calculated. It is observed that the ratios are 0.86%, 0.11% and 0.03 % which are close to the actual values obtained using the unequal error protection algorithm.

Type of frame	Number of frames lost (UEP)	Number of frames lost (EEP)	Offset distortion approach (UEP)	Offset distortion approach (EEP)
T0	0	0		
T1	0	1		
T2	1191	2		
Total	1191	3	1191	3
PSNR of reconstructed video (dB)	36.076	36.674	36.075	36.674
Degradation in PSNR (dB)	0.58	0.001	0.581	0.001

Table 6.3: Frame loss and PSNR statistics for SONY DEMO G16B3 with overhead of 3.91%

It is observed that only 3 frames are lost in the case of EEP as compared to 1191 frames in UEP. The reason behind such a mismatch is because, with increase in number of parity bytes, the distribution shifts more towards assigning parity bytes to T0 frames than to T1 and T2 frames. In this case very weak protection (only 2% of total number of parity bytes) is assigned to T2 and hence 1191 T2 frames are dropped.

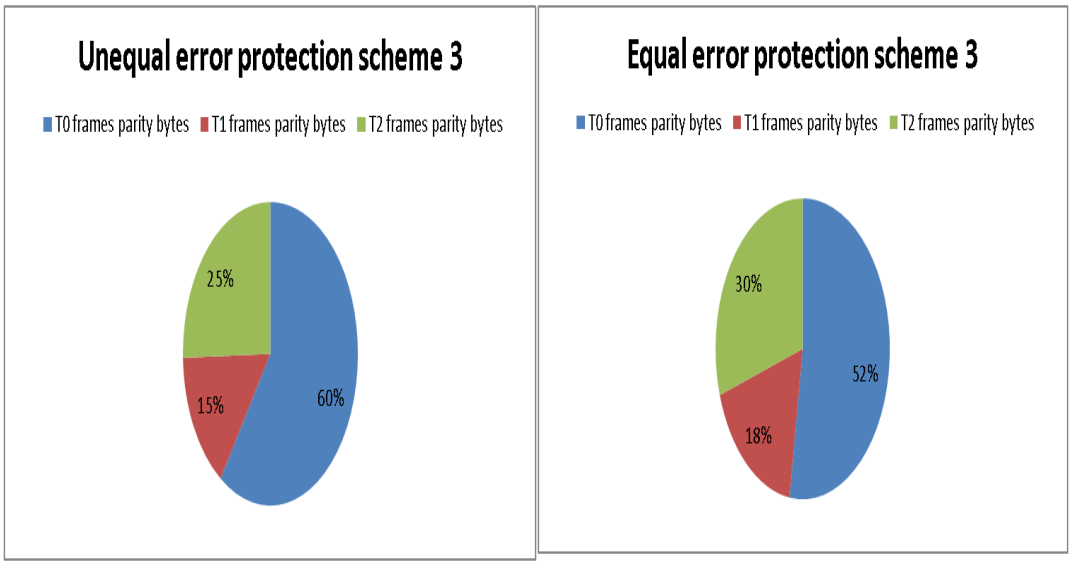


Figure 6.4: UEP and EEP schemes for Sony Demo G16B3 with overhead 0.25%

Type of frame	Number of frames lost (UEP)	Number of frames lost (EEP)
T0	244	530
T1	620	908
T2	1617	2031
Total	2481	3469
PSNR of reconstructed video (dB)	35.032	34.244
Degradation in PSNR (dB)	1.624	2.412

Table 6.4: Frame loss and PSNR statistics for Sony Demo G16B3 with overhead 0.25 % (BER 0.5 x 10^{-3})

In the third error protection scheme, the overhead is kept minimal (0.25%). The overhead is too low that, at BER of 10^{-3}, the number of frames lost in the case of EEP is 13418 and in the case of UEP it is 10938. Since the number of frames lost is huge, the PSNR degradation is very high and the visual quality will be worse. Thus with decrease in overhead, the performance worsens and the video sequence can be reconstructed for acceptable video quality only at a BER range of 0.5 x 10^{-3}, which is half the number of errors that can occur due to the channel compared to schemes 1 and 2.

At a BER of 0.5 x 10^{-3}, in the case of UEP performance degradation is 1.624 dB (2481 frames lost) whereas in the case of EEP, the performance degradation is 2.412 dB (3469 frames lost). The reason behind this is that the distribution of parity bytes in the case of UEP follows the distribution of actual data bytes in the bit stream more closely than in the case of EEP. In scheme 3 UEP, the distribution of parity bytes is 60% for T0, 15% for T1 and 25% for T2 and in the case of scheme 3 EEP, it is 52% for T0, 18% for T1 and 30% for T2 as shown in Figure 6.4. The distribution of the actual video traffic, as shown in Figure 6.1, is 65% for T0, 15% for T1 and 20% for T2. This distribution matches closely with UEP than EEP in the third scheme and hence the UEP scheme performs better in terms of number of missing frames and PSNR degradation.

From all three schemes for the SONY DEMO sequence, it is observed that when the distribution of parity bytes allocated for the different types of frames namely

T0, T1 and T2, follows the distribution of bytes among the different frames in the video traffic, the performance degradation is minimal. From the above results, it is also inferred that if the overhead is too low, the number of lost frames would be high and the system would not be able to operate at higher BER values as seen in scheme 3 where the over head is only 0.25%. With increase in number of parity bytes added to the bit stream, the distribution of parity bytes in the case of unequal error protection becomes more skewed towards the more important frames leading to performance degradation due to dropping of large number of lesser important frames. This is evident from the distribution of parity bytes in scheme 2 where the T2 frames are assigned a very small proportion of the parity bytes.

BER	10^{-4}	10^{-5}	10^{-6}
Number of missing frames	12080	2289	211
PSNR	25.75	34.998	36.51
PSNR degradation	10.906	1.658	0.146

Table 6.5: Frame loss and PSNR statistics for SONY DEMO G16B3 with no error protection

In the case where FEC codes are not employed, for BER upto 10^{-5}, the quality of the reconstructed video sequence is bad. . The frame losses observed are very high and the decoded video will not be fit for viewing. In the case where no FEC

is employed, we eliminate the RS Encoder/Decoder blocks and Interleaver/Deinterleaver blocks in Figure 4.1 in Chapter 4. Interleaving is effective only when used with FEC. If interleaving is employed when FEC is not used, it leads to corruption of more number of packets and in turn more number of NALUs are dropped. Affecting small parts of different NALUs will not help in this case as there is no error protection scheme to recover the corrupted data. Table 6.5 shows the performance of the system when no error protection scheme is employed.

This clearly shows the benefit of applying physical layer error protection to the NAL units. The reconstructed video provides acceptable performance for a BER of 10^{-3} in the case where error protection is applied compared to the case where no error protection is applied where the acceptable BER is 10^{-5}. The results obtained using the offset distortion approach match with that obtained using the actual bit stream. This shows that our model can be used for simulating the transport of the video frames over a lossy network and by just using the trace information, the statistics corresponding to the missing frames can be computed.

6.1.2 SONY DEMO G16B7 sequence

The SONY DEMO video is encoded using G16B7 GOP structures. The BER of the channel is fixed at 10^{-3}. The statistics for G16B7 encoded at QP=30 is given in Table 6.6 and the distribution of the video traffic among the different types of frames namely T0, T1, T2 and T3 frames is shown in figure 6.5.

Average Bit Rate (in Kbps)	361.1	
Average PSNR of decoded video(dB)	36.745	
Type of frame and number of frames	Number of frames	Average frame size (in bytes)
T0 frames	2210	6312
T1 frames	2210	1442
T2 frames	4420	967
T3 frames	8840	605
Total	17680	1600

Table 6.6: Video traffic statistics of SONY DEMO G16B7 sequence (QP=30)

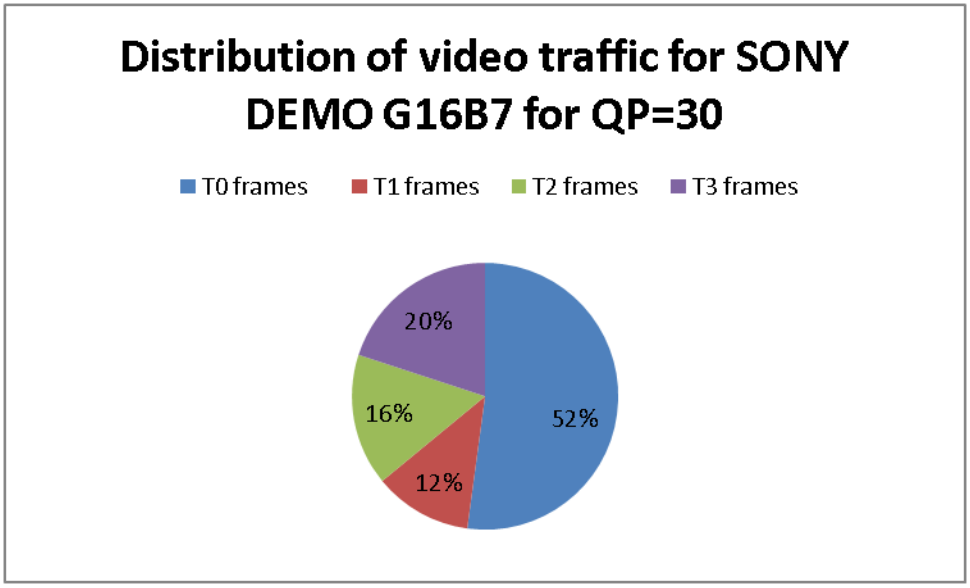


Figure 6.5: Distribution of video traffic for SONY DEMO G16B7 for QP=30

Experiments are constructed for EEP and UEP with an overhead of 0.8% and 3.78%. In the scheme 1, the over head added in terms of parity bytes is 0.8% of the overall video traffic. The distribution of parity bytes added in the case of EEP and UEP is shown in Figure 6.6. Table 6.7 provides the frame loss and PSNR statistics for EEP and UEP in scheme 1.

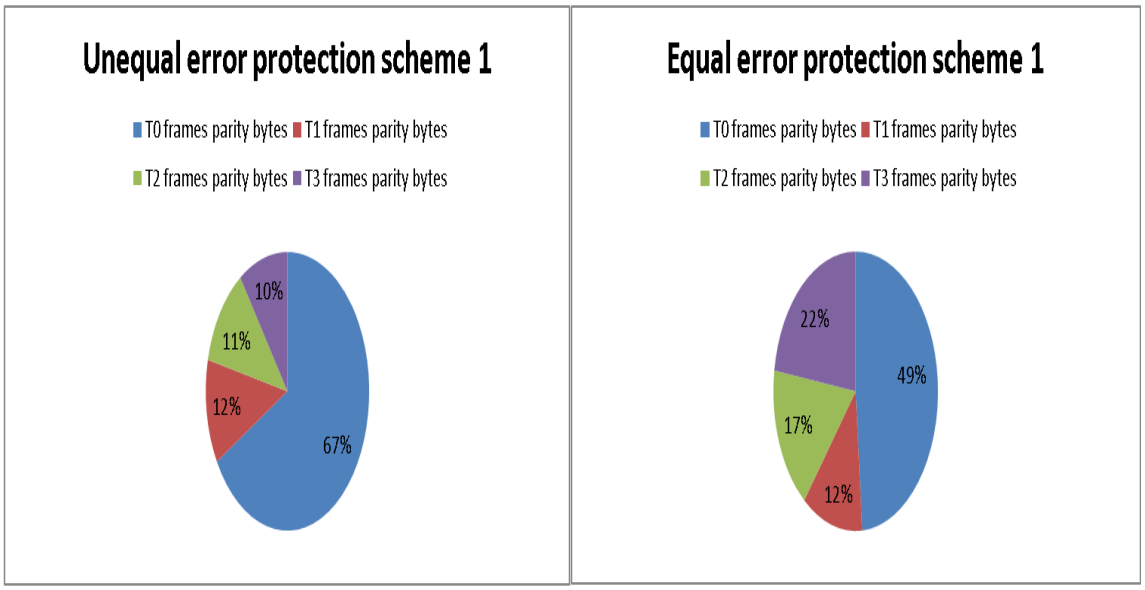


Figure 6.6: UEP and EEP schemes for SONY DEMO G16B7 with overhead of 0.8%

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	2	1
T1	9	13
T2	52	118

T3	180	1016
Total	243	1148
PSNR of reconstructed video (dB)	36.661	36.245
Degradation in PSNR (dB)	0.084	0.5

Table 6.7: Frame loss and PSNR statistics for SONY DEMO G16B7 with overhead of 0.8%

EEP performs better than UEP. In the case of EEP, PNSR degradation obtained is 0.084 dB (243 missed frames). In the case of UEP, PSNR degradation is 0.5 dB (1148 missing frames).

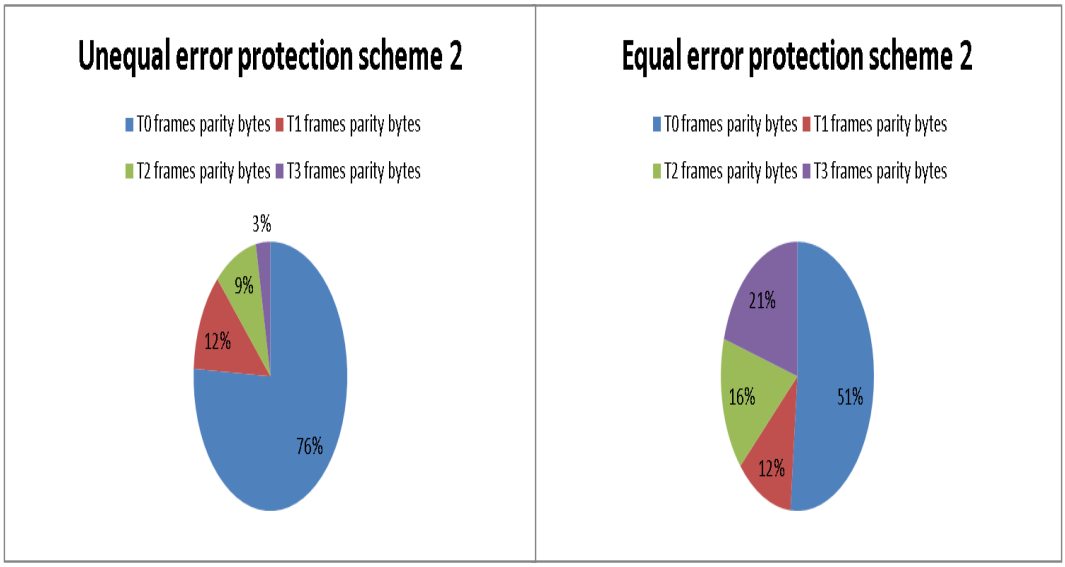


Figure 6.7: UEP and EEP schemes for SONY DEMO G16B7 with overhead of 3.78%

In scheme 2, overhead is kept at 3.78% for both EEP and UEP and PSNR of reconstructed videos are calculated. The distribution of parity bytes in case of EEP and UEP is as shown in figure 6.7. The frame loss and PSNR statistics are given in Table 6.8.

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	0	0
T1	0	0
T2	0	2
T3	1	307
Total	1	309
PSNR of reconstructed video (dB)	36.744	36.626
Degradation in PSNR (dB)	0.001	0.12

Table 6.8: Frame loss and PSNR statistics for SONY DEMO G16B7 with overhead of 3.78%

6.1.3 Sony Demo G16B0 sequence

The Sony G16B0 sequence, encoded at QP=31, has only 1 temporal level and it consists of only I and P frames. Since the number of temporal levels and quality

levels is 1 in this case, we use Equal error protection to assign parity bytes to the different NAL units as explained in Chapter 4. The overhead used in this scheme is 0.87%. The video and frame loss/PSNR statistics is given in Table 6.9. Table 6.10 shows the performance of the G16B0 sequence in the absence of error protection. It is observed that G16B0 GOP structure performs worse than G16B3 GOP structures in the absence of error protection.

Average Bit rate (Kbps)	373.98	
Average PSNR of decoded video (dB)	35.632	
Type of frame	Number of frames	Average frame size (bytes)
T0	17680	1567
Number of lost frames	341	
PSNR of reconstructed video (dB)	35.452	
Degradation in PSNR (dB)	0.18	

Table 6.9: Statistics for Sony Demo G16B0 sequence at a BER of 10^{-3}

(QP=31, Overhead =0.87%)

BER	10^{-4}	10^{-5}	10^{-6}
Number of missing frames	12643	2702	638
PSNR	24.382	33.562	35.18
PSNR degradation	11.25	2.07	0.452

Table 6.10: Frame loss and PSNR statistics for SONY DEMO G16B0 with no error protection

6.2 Results for Star Wars sequence

6.2.1 Star Wars G16B3 sequence

For Star Wars G16B3 video sequence, two schemes are used with different overhead (in terms of parity bytes added). In the first scheme, the overhead is at 0.82% and in the second scheme a 3.78% overhead is chosen for EEP and UEP. The BER of the channel is fixed at 10^{-3}. The statistics for Star Wars G16B3 sequence encoded at QP=30 is given in Table 6.11 and the distribution of the video traffic among the different types of frames namely T0, T1 and T2 is shown in figure 6.8.

Average Bit Rate (in Kbps)	167.3	
Average PSNR of decoded video(dB)	41.088	
Type of frame and number of frames	Number of frames	Average frame size (in bytes)
T0 frames	13496	1462
T1 frames	13496	595
T2 frames	26992	383
Total	53984	706

Table 6.11: Video traffic statistics of Star Wars G16B3 sequence (QP=30)

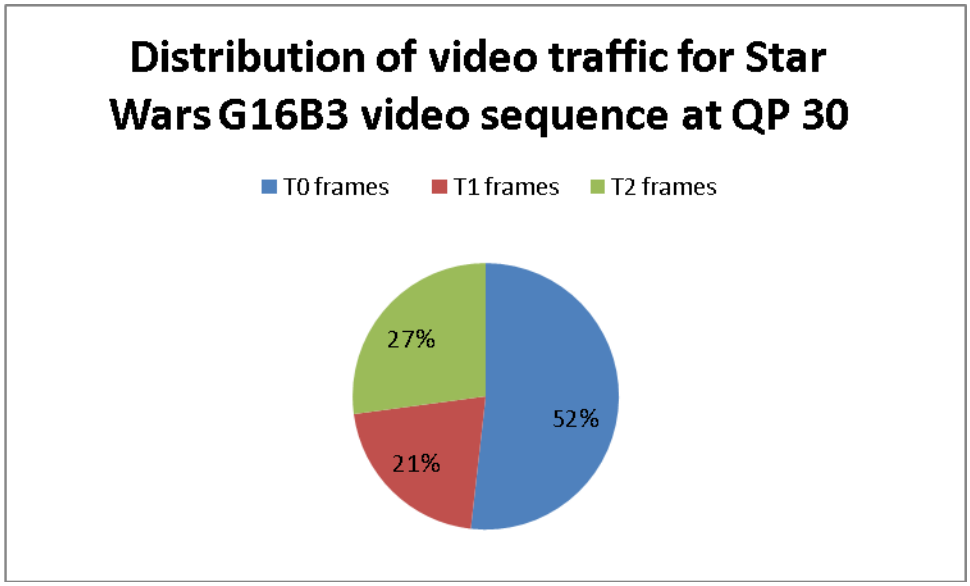


Figure 6.8: Distribution of video traffic for Star Wars G16B3 (QP=30)

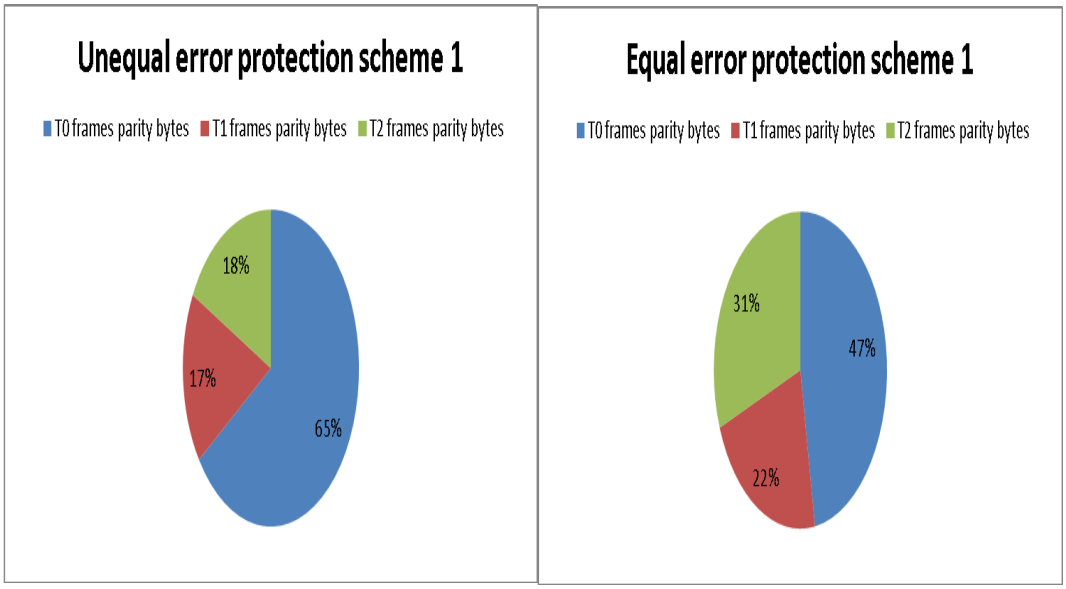


Figure 6.9: UEP and EEP schemes for Star Wars G16B3 with overhead of 0.82%

T0 frames comprising of I and P frames constitute 52% of the video traffic, T1 frames comprise of temporal level 1 B frame constitute 17% of the video traffic and T2 frames consisting of temporal level 2 B frames constitute 21% of the video traffic.

The distribution of parity bytes assigned to T0, T1 and T2 frames for scheme 1 is given in figure 6.9. In UEP, 65% of the total number of parity bytes is allocated to T0 frames, 17% to T1 frames and 18% to T2 frames. In the case of EEP, the distribution of parity bytes is 47%, 22% and 31% to T0, T1 and T2 frames respectively.

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	158	32
T1	359	366
T2	1043	2455
Total	1560	2853
PSNR of reconstructed video (dB)	40.872	40.589
PSNR Degradation (dB)	0.216	0.5

Table 6.12: Frame loss and PSNR statistics for Star Wars G16B3 with overhead of 0.82%

In scheme 1, the distribution of parity bytes for EEP scheme follows the distribution of video traffic among different layers namely T0, T1 and T2 more closely as compared to UEP. In the case of EEP, PSNR degradation obtained is 0.216 dB (1560 missed frames). In the case of UEP, PSNR degradation is 0.5 dB (2853 missing frames).

In scheme 2, the overhead is fixed at 3.78 %. The distribution of parity bytes among T0, T1 and T2 frames is shown in Figure 6.10 for the case of both EEP and UEP.

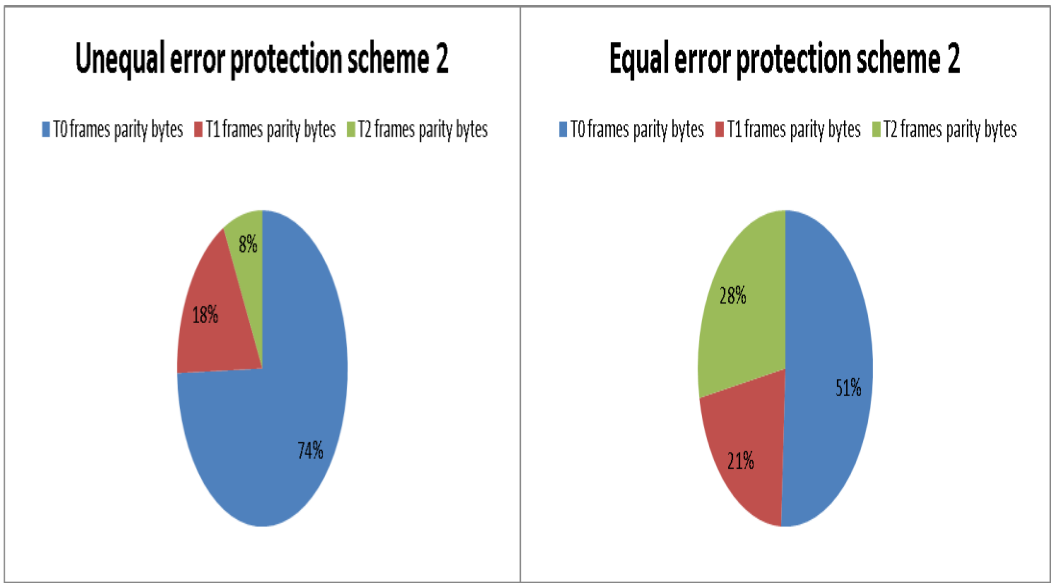


Figure 6.10: UEP and EEP schemes for STAR WARS G16B3 with overhead 3.78%

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	0	0
T1	0	0
T2	2	202
Total	2	202
PSNR of reconstructed video (dB)	41.087	41.018

Degradation in PSNR (dB)	0.001	0.070

Table 6.13: Frame loss and PSNR statistics for Star Wars G16B3 with overhead 3.78%

From the frame loss and PSNR statistics shown in Table 6.13, it is observed that EEP scheme loses only 2 frames whereas UEP loses 202 frames. It is observed that the allocation of parity bytes to T0, T1 and T2 frames follows the same distribution of the bytes in the Star Wars G16B3 video traffic. Hence, the performance degradation in the case of EEP is negligible.

6.2.2 Star Wars G16B7 sequence

The Star Wars video is encoded using G16B7 GOP structure in this experiment. The BER of the channel is fixed at 10^{-3}.

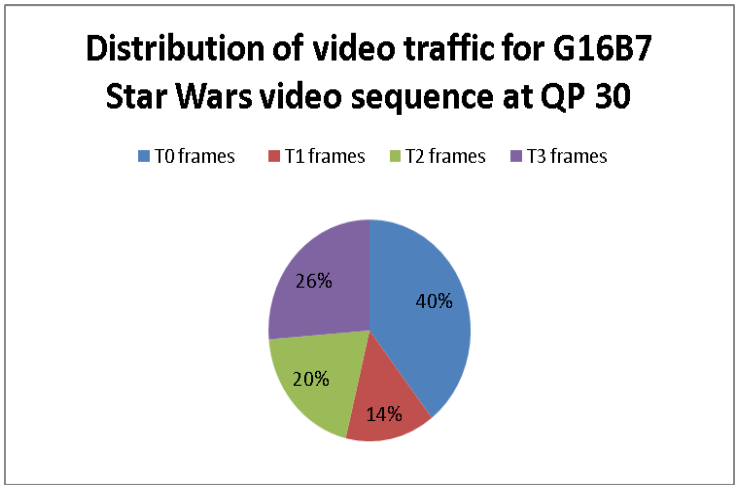


Figure 6.11: Distribution of video traffic for Star Wars G16B7 video sequence (QP= 30)

Average Bit Rate (in Kbps)	167.3	
Average PSNR of decoded video(dB)	41.184	
Type of frame and number of frames	Number of frames	Average frame size (in bytes)
T0 frames	6748	2147
T1 frames	6748	782
T2 frames	13496	542
T3 frames	26992	358
Total	53984	680

Table 6.14: Video traffic statistics for Star Wars G16B7 sequence (QP=30)

The statistics for Star Wars G16B7 video encoded at QP=30 is given in Table 6.14 and the distribution of the video traffic among the different types of frames namely T0, T1, T2 and T3 frames is shown in figure 6.11.

For Star Wars G16B7 video sequence, two schemes are used with different overheads (in terms of parity bytes added). In the first scheme, the overhead is at 0.8% and in the second scheme a 3.8% overhead is chosen for EEP and UEP. The BER of the channel is fixed at 10^{-3}. T0 frames comprising of I and P frames constitute 40% of the video traffic, T1 frames comprise of temporal level 1 B

frames constitute 14% of the video traffic, T2 frames consisting of temporal level 2 B frames constitute 20% of the video traffic and T3 frames consisting of temporal level 3 B frames constitute 26% of the traffic. In scheme 1, the allocation of error protection bytes to T0, T1, T2 and T3 frames is as shown in Figure 6.12.

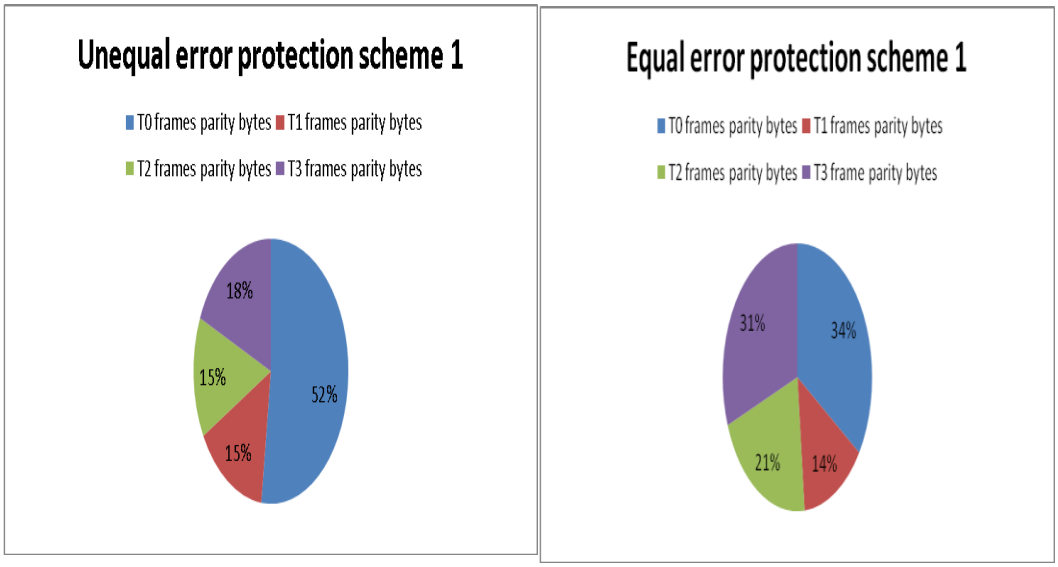


Figure 6.12: UEP and EEP schemes for STAR WARS G16B7 with overhead 0.8%

The frame loss and PSNR statistics are provided in Table 6.15. It is observed that the performance of EEP is better than that of UEP by 0.41 dB. In scheme 2, the overhead considered is 3.8% of the total video traffic. Figure 6.13 shows the distribution of parity bytes for UEP and EEP approaches.

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	32	7
T1	130	68
T2	425	630
T3	1182	2890
Total	1769	3595
PSNR of reconstructed video (dB)	40.952	40.541
Degradation in PSNR (dB)	0.233	0.643

Table 6.15: Frame loss and PSNR statistics for Star Wars G16B7 sequence with overhead 0.8%

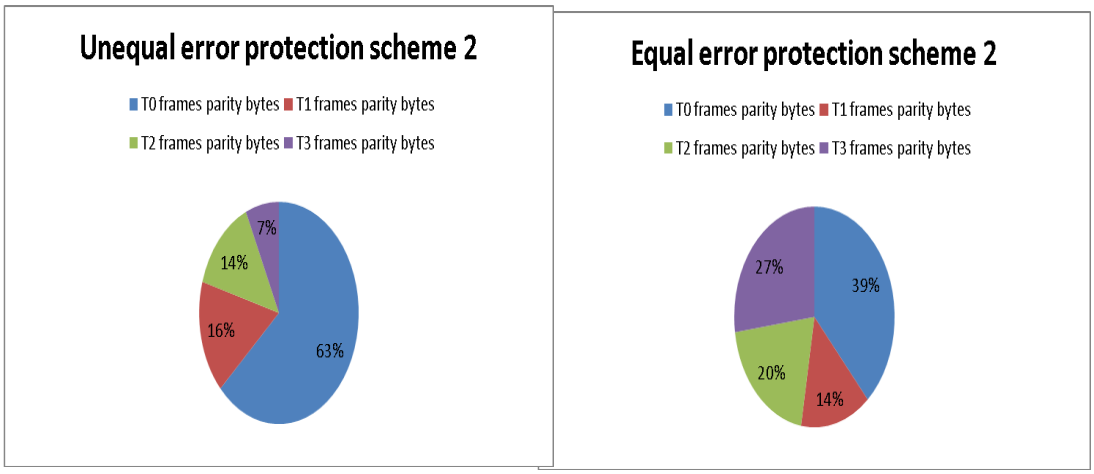


Figure 6.13: UEP and EEP schemes for Star Wars G16B7 with overhead 3.8%

The addition of parity bytes is skewed more towards the more important frames when the overhead is increased from 0.8% to 3.8% in the case of UEP. Even though, the amount of parity information added is approximately a five-fold increase, there still exists performance degradation in the case of UEP. The reason for the degradation is that the amount of parity information added to the lesser important frames (T3 frames in this case) in the case of scheme 2 are inadequate to correct the errors. EEP performs better than UEP when the amount of parity added increases as the distribution approaches the actual distribution of the video traffic. In both the Sony demo and Star Wars, the same trends are observed. Table 6.16 provides details on the performance degradation in terms of PSNR and number of lost frames for scheme 2.

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	0	0
T1	0	0
T2	1	4
T3	10	255
Total	11	259
PSNR of reconstructed video (dB)	41.18	41.161
Degradation in PSNR (dB)	0.004	0.023

Table 6.16: Frame loss and PSNR statistics for Star Wars G16B7 sequence with overhead 3.8%

6.3 Results for Silence of the Lambs sequence

6.3.1 Silence of the Lambs G16B3 sequence

For Silence of the Lambs G16B3 video sequence, two schemes are used with different overhead (in terms of parity bytes added). In the first scheme, the overhead is at 0.8% and in the second scheme a 3.8% overhead is chosen for EEP and UEP. The BER of the channel is fixed at 10^{-3}. The statistics for Silence of

the Lambs G16B3 sequence encoded at QP=30 is given in Table 6.17 and the distribution of the video traffic among the different types of frames namely T0, T1 and T2 is shown in figure 6.14.

Average Bit Rate (in Kbps)	211.28	
Average PSNR of decoded video(dB)	40.0861	
Type of frame and number of frames	Number of frames	Average frame size (in bytes)
T0 frames	6000	1816
T1 frames	6000	788
T2 frames	12000	476
Total	24000	889

Table 6.17: Video traffic statistics of Silence of the Lambs G16B3 sequence

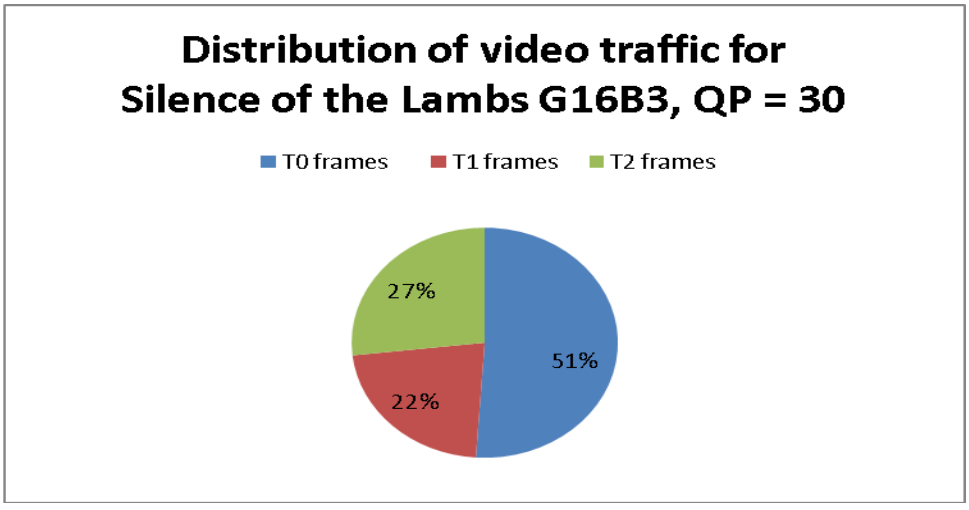


Figure 6.14: Distribution of video traffic for Silence of the Lambs G16B3 (QP=30)

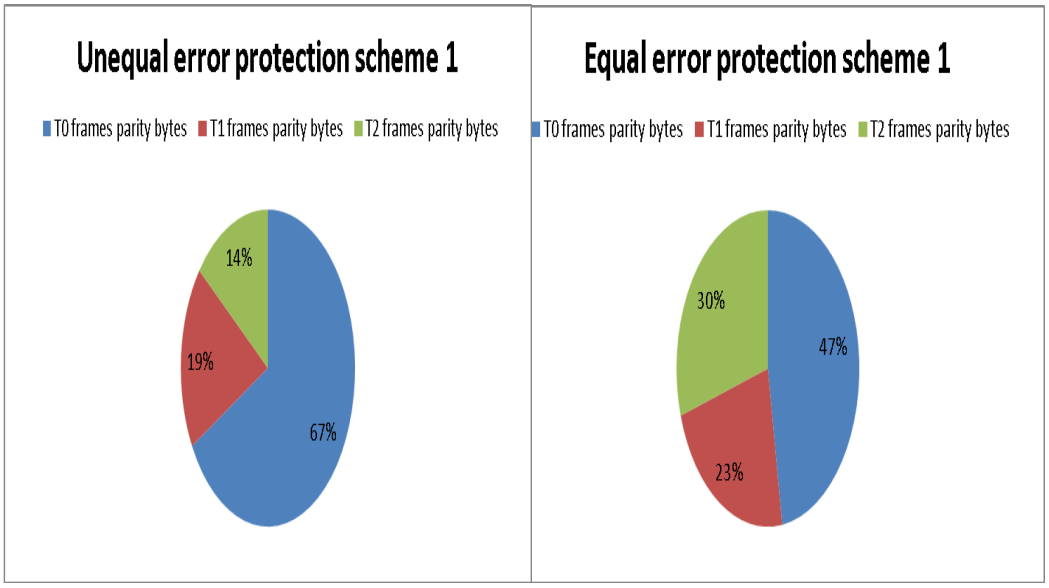


Figure 6.15: UEP and EEP schemes for Silence of the Lambs G16B3 with overhead of 0.8%

T0 frames comprising of I and P frames constitute 51% of the video traffic, T1 frames comprise of temporal level 1 B frame constitute 22% of the video traffic and T2 frames consisting of temporal level 2 B frames constitute 27% of the video traffic.

The distribution of parity bytes assigned to T0, T1 and T2 frames for scheme 1 is given in figure 6.15. In UEP, 67% of the total number of parity bytes is allocated to T0 frames, 19% to T1 frames and 14% to T2 frames. In the case of EEP, the distribution of parity bytes is 47%, 23% and 30% to T0, T1 and T2 frames respectively.

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	65	25
T1	144	149
T2	428	1301
Total	637	1475
PSNR of reconstructed video (dB)	39.8671	39.5034
PSNR Degradation (dB)	0.219	0.5827

Table 6.18: Frame loss and PSNR statistics for Silence of the Lambs G16B3 with overhead of 0.8%

In scheme 1, the distribution of parity bytes for EEP scheme follows the distribution of video traffic among different layers namely T0, T1 and T2 more closely as compared to UEP. In the case of EEP, PSNR degradation obtained is 0.219 dB. In the case of UEP, PSNR degradation is 0.583 dB.

In scheme 2, the overhead is fixed at 3.8 %. The distribution of parity bytes among T0, T1 and T2 frames is shown in Figure 6.16 for the case of both EEP and UEP.

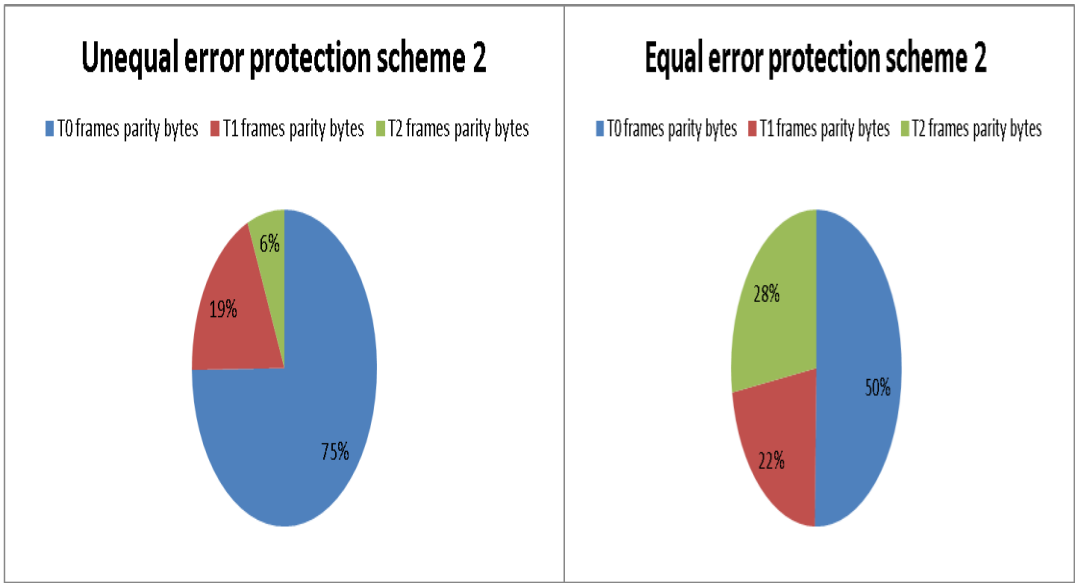


Figure 6.16: UEP and EEP schemes for Silence of the Lambs G16B3 with overhead 3.8%

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	0	0
T1	1	0
T2	3	126
Total	4	126
PSNR of reconstructed video (dB)	40.0859	40.0598
Degradation in PSNR (dB)	0.0002	0.0272

Table 6.19: Frame loss and PSNR statistics for Silence of the Lambs G16B3 with overhead 3.8%

From the frame loss and PSNR statistics shown in Table 6.19, it is observed that EEP scheme loses only 4 frames whereas UEP loses 126 frames. It is observed that the allocation of parity bytes to T0, T1 and T2 frames follows the distribution of the bytes in the Silence of the Lambs G16B3 video traffic very closely. Hence, the performance degradation in the case of EEP is negligible.

6.3.2 Silence of the Lambs G16B7 sequence

The Silence of the Lambs video is encoded using G16B7 GOP structure in this experiment. The BER of the channel is fixed at 10^{-3}.

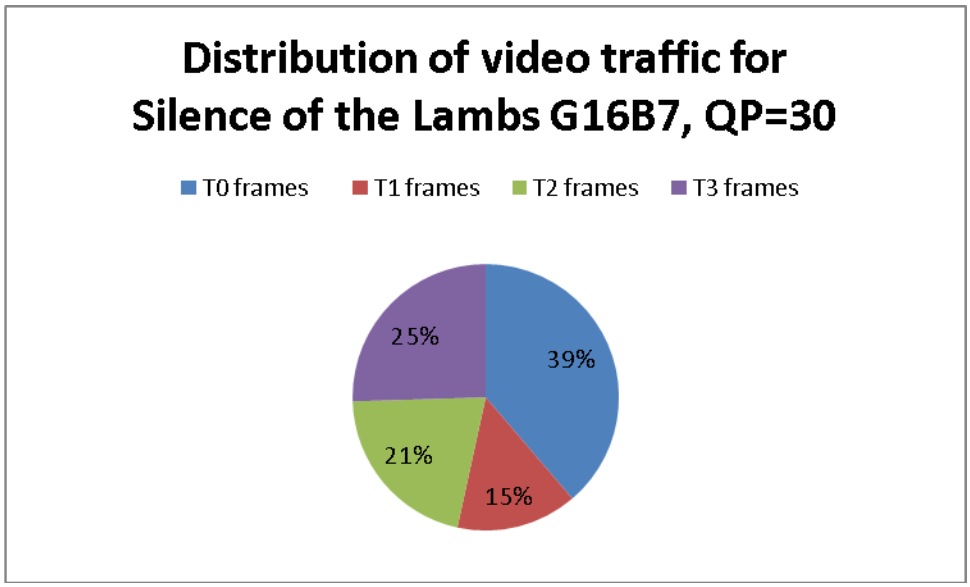


Figure 6.17: Distribution of video traffic for Silence of the Lambs G16B7 video (QP= 30)

Average Bit Rate (in Kbps)	204.7	
Average PSNR of decoded video(dB)	40.2510	
Type of frame and number of frames	Number of frames	Average frame size (in bytes)
T0 frames	3000	2668
T1 frames	3000	1013
T2 frames	6000	727
T3 frames	12000	439
Total	24000	861

Table 6.20: Video traffic statistics for Silence of the Lambs G16B7 sequence (QP=30)

The statistics for Silence of the Lambs G16B7 video encoded at QP=30 is given in Table 6.20 and the distribution of the video traffic among the different types of frames namely T0, T1, T2 and T3 frames is shown in figure 6.17.

For Silence of the Lambs G16B7 video sequence, two schemes are used with different overheads (in terms of parity bytes added). In the first scheme, the overhead is at 0.8% and in the second scheme a 3.8% overhead is chosen for EEP and UEP. The BER of the channel is fixed at 10^{-3}. T0 frames comprising of I and P frames constitute 39% of the video traffic, T1 frames comprise of temporal

level 1 B frames constitute 15% of the video traffic, T2 frames consisting of temporal level 2 B frames constitute 21% of the video traffic and T3 frames consisting of temporal level 3 B frames constitute 25% of the traffic. In scheme 1, the allocation of error protection bytes to T0, T1, T2 and T3 frames is as shown in Figure 6.18.

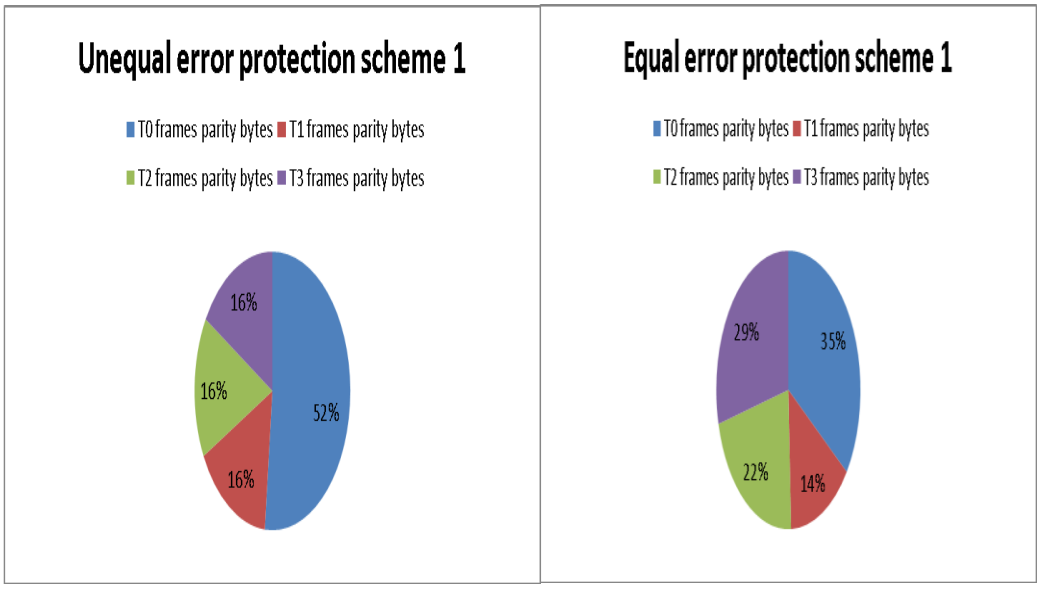


Figure 6.18: UEP and EEP schemes for Silence of the Lambs G16B7 with overhead 0.8%

The frame loss and PSNR statistics are provided in Table 6.21. It is observed that the performance of EEP is better than that of UEP by 0.27 dB. In scheme 2, the overhead considered is 3.8% of the total video traffic. Figure 6.19 shows the distribution of parity bytes for UEP and EEP approaches.

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	13	1
T1	39	12
T2	146	183
T3	429	1005
Total	627	1201
PSNR of reconstructed video (dB)	40.0646	39.7991
Degradation in PSNR (dB)	0.1864	0.452

Table 6.21: Frame loss and PSNR statistics for Silence of the Lambs G16B7 sequence with overhead 0.8%

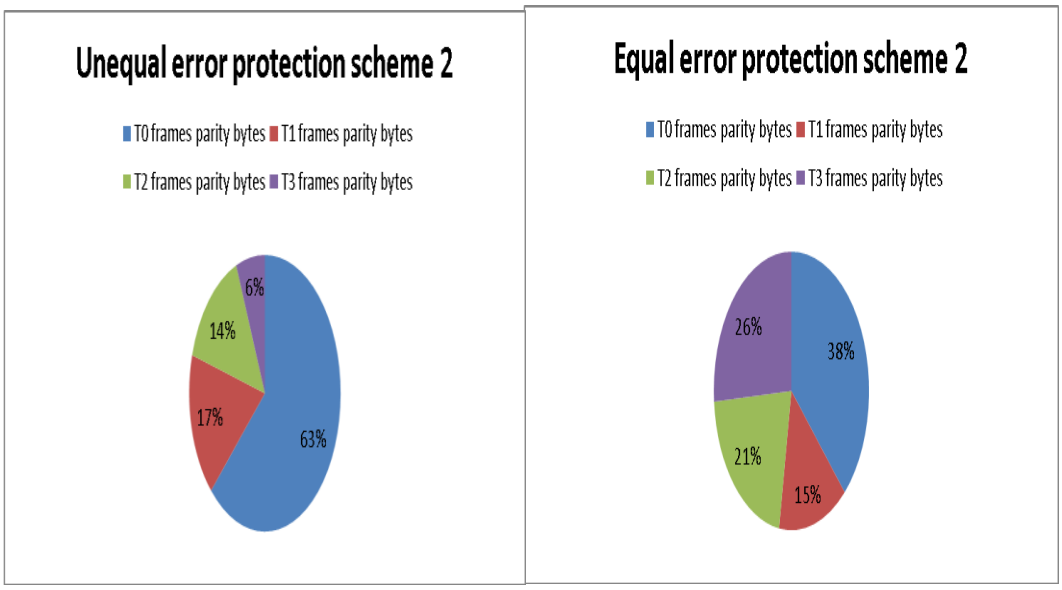


Figure 6.19: UEP and EEP schemes for Silence of the Lambs G16B7 with overhead 3.8%

The addition of parity bytes is skewed more towards the more important frames when the overhead is increased from 0.8% to 3.8% in the case of UEP. Even though, the amount of parity information added is approximately a five-fold increase, there still exists performance degradation in the case of UEP. The reason for the degradation is that the amount of parity information added to the lesser important frames (T3 frames in this case) in the case of scheme 2 are inadequate to correct the errors. EEP performs better than UEP when the amount of parity added increases as the distribution approaches the actual distribution of the video traffic. In all the three video sequences, the same trends are observed. Table 6.22 provides details on the performance degradation in terms of PSNR and number of lost frames for scheme 2.

Type of frame	Number of frames lost (EEP)	Number of frames lost (UEP)
T0	0	0
T1	0	0
T2	0	0
T3	2	191
Total	2	191
PSNR of reconstructed video (dB)	40.2509	40.2206
Degradation in PSNR (dB)	0.0001	0.0304

Table 6.22: Frame loss and PSNR statistics for Silence of the Lambs G16B7 sequence with overhead 3.8%

6.4 Conclusion

The performance of single layer H.264 SVC video error prone networks is evaluated in this study. Three long video sequences are used in this study for evaluation of the performance degradation. Three different schemes are compared – equal error protection, unequal error protection and no error protection for different overheads. Comparison of actual bit stream approach with offset distortion approach is performed for Sony Demo G16B3 sequence. The PSNR

and frame loss results show that both the methods provide the same results and our simulation model can be used to evaluate the degradation in performance by using the trace information. Applying error protection to the video stream has only negligible overhead, but the improvement in performance is significant. The operating range of the system increases from 10^{-5} to 10^{-3} when error protection is applied. It is observed that when the overhead is increased from around 1% to 4%, equal error protection assigns parity bytes closer to the distribution of the data among the different types of frames and performs better than unequal error protection. When the overhead increases in unequal error protection, the allocation of parity bytes is more skewed towards the more important frames and amount of parity bytes added to the least important frames is inadequate to recover the video data from the bit errors.

As part of future work, it would be interesting to evaluate the performance of the system for CGS and MGS encoded video. The number of priority levels in single layer SVC is limited because they have only temporal scalability. By using CGS and MGS, quality scalability is employed and hence more priority levels are present. For error concealment, this model uses Frame copy where the last played out frame is copied in place of the missing frames. This doesn't take into account the motion information of the lost frame and hence degradation in PSNR is more, especially in the case of high motion video sequences. Evaluation of the performance of the system using better error concealment approaches that take

into account the motion information of the lost frames would be an interesting area of study.

REFERENCES

[1] http://r2d2n3po.tistory.com/.
[2] D. Bakker, D. Cromboom, T. Dams, A. Munteanu, and J. Barbarien. Priority-based error protection for the scalable extension of H. 264/SVC. In Proc. of SPIE Conference on Optical and Digital Image Processing, volume 7000.
[3] D. Bakker, M. Stoufs, J. Barbarien, M. Jacobs, P. Schelkens, and
A. Munteanu. Optimized erasure-resilient streaming of SVC using unequal error protection. In Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on, pages 413-416. IEEE, 2008.
[4] B. Barmada, MM Ghandi, EV Jones, and M. Ghanbari. Prioritized transmission of data partitioned H. 264 video with hierarchical QAM. Signal Processing Letters, IEEE, 12(8):577- 580, 2005.
[5] E.R. Berlekamp. Algebraic coding theory, volume 111. McGraw-Hill New York, 1968.
[6] 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. IEEE, 2009.
[7] R. D'Haenens, J. Doggen, D. Bakker, and T. Dams. Transmitting Scalable Video with Unequal Error Protection over 802.11 b/g. In Networking and Communications, 2008. WIMOB'08.IEEE International Conference on Wireless and Mobile Computing, pages 638-643, 2008.
[8] S. Lin and D.J. Costello. Error control coding: fundamentals and applications. Prentice-hall, Englewood Cliffs, NJ, 1983.
[9] A. Naghdinezhad, MR Hashemi, and O. Fatemi. A novel adaptive unequal error protection method for scalable video over wireless networks. In IEEE International Symposium on Consumer Electronics, 2007. ISCE 2007, pages 1-6, 2007.
[10] A. Pulipaka, P. Seeling, M. Reisslein, and L.J. Karam. Overview and traffic characterization of coarse-grain quality scalable (CGS) H.264 SVC encoded video. In Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, pages 1-5. IEEE, 2010.
[11] J. Reichel, H. Schwarz, and M. Wien. Joint Scalable Video Model JSVM-3. Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, Doc. JVT- P, 2005.
[12] H. Schwarz and M. Wien. The Scalable Video Coding Extension of the H. 264/AVC Standard. IEEE Signal Processing Magazine, 25(2):135, 2008.
[13] P. Seeling, M. Reisslein, and F.H.P. Fitzek. Offset distortion traces for trace-based evaluation of video quality after network transport. In Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on, pages 375-380. IEEE, 2005.
[14] Y.K. Wang, M.M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger. System and transport interface of SVC. IEEE transactions on circuits and systems for video technology, 17(9):1149-1163, 2007.

[15] S. Wenger, Y. 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.
[16] Fiandrotti, A.; Gallucci, D.; Masala, E.; Magli, E.; ,Traffic Prioritization of H.264/SVC Video over 802.11e Ad Hoc Wireless Networks, Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on , vol., no., pp.1-5, 3-7 Aug. 2008
[17] Schierl, T.; Hellge, C.; Mirta, S.; Gruneberg, K.; Wiegand, T.; ,Using H.264/AVC-based Scalable Video Coding (SVC) for Real Time Streaming in Wireless IP Networks, Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on , vol., no., pp.3455-3458, 27-30 May 2007
[18] Monteiro, J.M.; Calafate, C.T.; Nunes, M.S.;Evaluation of the H.264 Scalable Video Coding in Error Prone IP Networks, Broadcasting, IEEE Transactions on , vol.54, no.3, pp.652-659, Sept. 2008
[19] D. Marpe, T. Wiegand, and G. Sullivan, The H.264/MPEG-4 advanced video coding standard and its applications, IEEE Communications Magazine, vol. 44, no. 8, pp.,134-143, Aug. 2006.
[20] M. Ghanbari, Standard Codecs: Image Compression to Advanced Video Coding, Institution of Electrical Engineers, London, UK, 2003
[21] T. Wiegand, X. Zhang, and B. Girod, Long-term memory motion-compensated prediction, IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 1, pp. 70-84, Feb 1999
[22] H. Schwarz, D. Marpe, and T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard, IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103-1120, Sept. 2007
[23] Wang Ye-Kui; Hannuksela, M.M.; Pateux, S.; Eleftheriadis, A.; Wenger, S , System and Transport Interface of SVC, Circuits and Systems for Video Technology, IEEE Transactions on , vol.17, no.9, pp.1149-1163, Sept. 2007
[24] T Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits and Systems for Video Technology, vol. 13, pp. 560-576, Jul 2003.
[25] J Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, Video coding with H.264/AVC: tools, performance and complexity, IEEE Circuits and Systems Magazine, vol. 4, no. 1, pp. 7-28, 2004
[26] T. Wiegand, G.J. Sullivan, J.R. Ohm, and A. Luthra, Introduction to the Special Issue on Scalable Video Coding: Standardization and Beyond, IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1099-1102, Sep 2007.
[27] H. Schwarz and M. Wien, The scalable video coding extension of the H.264/AVC standard, IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 135-141, Mar. 2008.

[28] H. Schwarz and T. Wiegand, Implementation and performance of FGS, MGS and CGS, Joint Video Team JVT of ISO/IEC MPEG and ITU-T VGEC, Doc. JVTV126, Jan. 2007
[29] Van der Auwera, G.; David, P.T.; Reisslein, M., Traffic and Quality Characterization of Single-Layer Video Streams Encoded with the H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding Extension," Broadcasting, IEEE Transactions on , vol.54, no.3, pp.698-718, Sept. 2008
[30] Amon, P.; Rathgen, T.; Singer, D , File Format for Scalable Video Coding, Circuits and Systems for Video Technology, IEEE Transactions on , vol.17, no.9, pp.1174-1185,Sept.2007
[31] Reimers, U.H., DVB-The Family of International Standards for Digital Video Broadcasting," Proceedings of the IEEE , vol.94, no.1, pp.173-182, Jan. 2006
[32] Y. Chen, K. Xie, F. Zhang, P. Pandit, and J. Boyce, Frame loss error concealment for SVC, Journal of Zhejiang University SCIENCE A, also in Proc. Packet Video Workshop'06, pp. 677–683, Hangzhou, China, Apr. 2006
[33] I.Guo; Ying Chen; Ye-Kui Wang; Houqiang Li; Hannuksela, M.M.; Gabbouj, M.; , "Error Resilient Coding and Error Concealment in Scalable Video Coding," Circuits and Systems for Video Technology, IEEE Transactions on , vol.19, no.6, pp.781-795, June 2009
[34] P. Seeling and M. Reisslein.Evaluating Multimedia Networking Mechanisms Using Video Traces. IEEE Potentials, 24(4):21-25, October/November 2005.
[35] P. Seeling, M. Reisslein, and F.H.P. Fitzek. Offset Trace-Based Video Quality Evaluation after Network Transport. Journal of Multimedia (Academy Publisher), 1(2):1-13, May 2006.