MULTIPLEXING/DE-MULTIPLEXING DIRAC VIDEO WITH AAC AUDIO BIT STREAM

by

ASHWINI URS

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2011

ACKNOWLEDGEMENTS

Firstly, I express my sincere gratitude to Dr. K. R. Rao for being my advisor and his constant encouragement and motivation ever since I joined UTA. He has always been a source of inspiration and his invaluable advice has helped me to successfully complete my research. I have always admired him for his immense knowledge and contribution to the area of multimedia processing. I am thankful and honored for having the opportunity to be a part of his research group.

I am thankful to Dr. Jonathan Bredow and Dr. Alan Davis for being a part of my thesis committee during their busy schedule and providing insightful comments.

I am thankful to Dr. Tim Borer for his suggestions, comments and for sharing Dirac video codec specification documents which guided me in the research.

I am thankful to Mrs. Anuradha Suraparaju of BBC research team for suggestions and comments on mkvmerge which aided me in successfully completing my work.

I am grateful to my husband, Nagesh Urs, who has inspired, encouraged and motivated me to pursue the MSEE degree. This would not have been possible without his guidance and support. I am thankful to my father, mother and brother who have motivated and supported me in all my endeavors. I would also like to thank my in-laws and all the family members who have encouraged me to pursue the master's degree.

I would also like to thank the EE department advising staff who have always helped me with my questions and made my stay a memorable one.

Finally, I would like to thank Naveen, Pragnesh and all my friends who have directly or indirectly helped me in my research.

April 8, 2011

ABSTRACT

MULTIPLEXING/DE-MULTIPLEXING DIRAC VIDEO WITH AAC AUDIO BIT STREAM

Ashwini Urs, M.S

The University of Texas at Arlington, 2011

Supervising Professor: K.R.Rao

With the inception of High Definition Television (HDTV) for broadcasting digital multimedia, enormous demand for video streaming over internet and Internet Protocol Television (IPTV) applications, the choice of a good compression scheme is vital. A good compression scheme assists in exploiting the limited storage capacity and efficient use of bandwidth required for broadcasting.

Dirac [31] is a state-of-the-art video codec aimed at applications from HDTV to web streaming [1]. Dirac was developed by the British Broadcasting Corporation (BBC) and is an open technology which does not involve any licensing fees. Studies have shown that the performance of Dirac compares well to the H.264 video codec [3]. At low bitrates, the quality of video deteriorates due to distortion for the Dirac video codec, while H.264 outperforms [2]. Performance of Dirac for HD media is similar to H.264, due to absence of large and intolerable variations between the two codecs [2]. Hence, Dirac is chosen as the video codec in this thesis. The right choice of audio codec is also necessary. Advanced Audio Coding (AAC) [4] is one of the audio digital codec standards defined in Moving Picture Experts Group (MPEG-2) and MPEG-4 with a few modifications [4]. The audio sampling frequency ranges from 8 kHz – 96 kHz [5]. The performance of AAC is superior at bitrates greater than 64 Kbps and also at lower bitrates (16 Kbps), and hence it is adopted in this thesis [22]. The raw video and audio data is encoded using the Dirac video and the AAC audio codec respectively. The video and audio bit-streams obtained need to be multiplexed as a single stream in order to be transmitted over the network. The objective of this thesis is to multiplex

the video and audio bit-streams for transmission, de-multiplex audio and video bit-streams at the receiver's end while maintaining lip synchronization during the playback. The MPEG-2 [19] system is adopted in this thesis to achieve the multiplexing process. The bit-streams of audio and video correspond to the respective frame data. This data is packetized as Packetized Elementary Streams (PES) which is of variable lengths. This is further packetized as Transport Stream (TS) packets of fixed length and 188 bytes long [9]. The fixed size packet length facilitates the transmission process. The timestamp information is encapsulated into the PES header in the form of frame numbers which help in achieving lip synchronization during playback. The presentation time of video and audio is used as a reference in multiplexing the audio and the video TS packets which aid in ensuring the buffer fullness (i.e. prevents buffer overflow or underflow) at the de-multiplexer end. Sequence Parameter Sets (SPS) and Picture Parameter Sets (PPS) present in the video bit-stream are also transmitted in the form of packets to assist the decoder in decoding the video data. The header information included helps in a faster and efficient demultiplexing process. The algorithm for multiplexing and de-multiplexing was implemented while maintaining lip sync during playback. Advanced Television Systems Committee – Mobile/ Handheld (ATSC - M/H) has an allocated bandwidth requirement of 19.6 Mbps [13], whereas the transport stream bitrates obtained using the multiplexing algorithm implemented for the inter coding of sequences used are 102.13 kbps and 96.72 kbps which can be easily and efficiently accommodated. Encoding video using Dirac and audio based on AAC, multiplexing the two coded bit-streams, packetization, de-multiplexing the two coded bit-streams, decoding the video (Dirac) and audio (AAC) while maintaining the lip sync are the highlights of this thesis. Advantages and limitations of the method proposed are discussed in detail.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AAC – Advanced Audio Coding.

AC – Arithmetic Coding.

ADIF – Audio Data Interchange Format.

ADTS – Audio Data Transport Format.

AFC – Adaptation Field Control.

ATSC – Advanced Television Systems committee.

ATSC - M/H – Advanced Television Systems committee – Mobile / Handheld.

ATSC - H – Advanced Television Systems committee – Handheld.

AVI – Audio Video Interleave.

B – Bi-directional.

BBC – British Broadcasting Corporation.

CC – Continuity Counter.

DCT – Discrete Cosine Transform.

DVB – Digital Video Broadcasting.

DVB – H – Digital Video Broadcasting - Handheld.

EBU – European Broadcasting Union.

FAAC – Free Advanced Audio Coder.

GOP – Group of Pictures.

HDTV– High Definition Television.

HH – High - High.

HL – High - Low.

I – Intra.

IDR – Instantaneous Decoder Refresh.

IPTV– Internet Protocol Television.

ITU – R – International Telecommunication Union – Radio communication sector.

JPEG – Joint Photographic Experts Group.

LH – Low - High.

LL – Low - Low.

KBD – Kaiser Bessel Derived.

MDCT – Modified Discrete Cosine Transform.

MPEG – Moving Picture Experts Group.

M/S – Mid- Side.

NAL – Network Abstraction Layer.

OBMC – Over-lapped Block based Motion Compensation.

P – Predictive.

PCR – Program Clock Reference.

PID – Packet Identifiers.

PES – Packetized Elementary Stream.

PPS – Picture Parameter Sets.

PUSI – Payload Unit Start Indicator.

QF – Quality Factor.

SPS – Sequence Parameter Sets.

SMPTE – Society of Motion Pictures and Television Engineers.

TNS – Temporal Noise Shaping.

TS – Transport Stream.

YUV – Luminance and Chrominance components.

CHAPTER 1

INTRODUCTION

1.1 Introduction

In today's digital world, good quality, high resolution and efficient use of bandwidth is imperative. Digital Video Broadcast - Handheld (DVB-H) [16] and Advanced Television Systems Committee – Mobile / Handheld (ATSC – M/H) [17] [18] are amongst major digital TV broadcasters and the allocation of bandwidth is relatively ~ 14Mbps and ~19.6 Mbps for the former and latter respectively. To efficiently use the bandwidth for video and audio data, compression is necessary.

In order to meet these requirements, a good compression scheme has to be chosen. Hence, a choice of good video codec along with audio codec in order to have a complete and meaningful delivery of data becomes necessary. This can be achieved by choosing Dirac [31] as video codec and AAC [4] as audio codec. Dirac is an open technology meaning that it does not involve any licensing fees and is an open source available to all [1]. Dirac uses wavelet transform and is applied on the entire image at once and hence no blocky artifacts are observed unlike other conventional video codecs. For entropy coding, Dirac uses arithmetic coding which packs bits into a bit stream efficiently [1]. Dirac incorporates long Group of Pictures (GOP) structure format. The performance of Dirac is comparable with that of H.264 video codec in terms of compression ratio and quality while H.264 outperforms at lower bitrates [3]. Whilst, for HD media the performance of Dirac is as good as H.264 [2]. Hence, the choice of Dirac video codec in this thesis. AAC is a second generation audio coding scheme for generic coding of stereo and multichannel signals [11]. The wide range of audio sampling frequency support ranges from 8 kHz – 96 kHz [5] with 13 pre-defined frequencies [15]. The performance of AAC audio codec is superior at bitrates greater than 64 kbps and also at lower bitrates (16 kbps) and hence it is adopted in this thesis [22]. AAC is one of the popularly used audio codec for most applications like broadcasting, streaming. Once the

right choice for the video and audio codec has been made, the video and audio data needs to multiplexed and transmitted enabling reliable and timely delivery of data to the consumer and hence the need for multiplexing. MPEG-2 [19] TS specification is used to achieve the same. Firstly, the encoded video and audio data undergoes first layer of packetization, PES and then PES elementary stream is packetized to 188 bytes long TS packets, which are apt and required for transmission of data in a channel. The data is received at the de-multiplexer and audio-video data is sent to their respective buffers using the knowledge available from TS header. This is then put to a container format using mkvmerge [32] and the metadata is decoded and played back using VLC media player [40]. The insight to the process of multiplexing, de-multiplexing and achieving lip synchronization during playback is discussed later in the following chapters.

1.2 Thesis outline

Chapters 2 and 3 introduce the Dirac video and AAC audio codec used in this thesis. The reason for choosing Dirac as video codec and AAC as audio codec and the bit-stream format of Dirac and AAC codecs are also addressed.

Chapter 4 provides a detailed description of multiplexing process adopted in this thesis. The process of packetizing the elementary streams into PES and further to TS as per MPEG-2 specifications is discussed along with the time stamp information in the form of frame numbers.

Chapter 5 provides a detailed description of de-multiplexing process adopted in this thesis. The information embedded in the header of the TS packets aids to demultiplex the transmitted program into video and audio buffers. The time stamp information embedded in the PES header helps to achieve synchronization between video and audio during playback.

Chapter 6 tabulates results of the thesis showing that delay between video and audio is not perceptible and synchronization is achieved during playback. It also discusses conclusions and future work of the algorithms implemented.

CHAPTER 2

OVERVIEW OF DIRAC VIDEO CODEC

Dirac video codec [31] achieves state of the art performance - good quality at low bitrates, leading to lower costs [26]. Dirac is a hybrid video codec developed by the BBC research team and named after Paul Dirac, a British physicist [28]. Dirac 1.0.2 version of C++ implementation is used in this thesis and was released in February 2009 [29]. It is an open technology involving no licensing fees. Dirac video codec has two software implementations, namely, Dirac research and Schrodinger [28]. Dirac has a versatile range of applications including digital video broadcasting, internet streaming and IPTV to name a few. With industry plans for “On demand” TV and enormous demand for streaming over internet, open platform technologies like Dirac have become highly significant [26]. Dirac is apt for applications from low resolution to HDTV. Compression achieved using Dirac can be lossless or lossy when long GOP format is used. Dirac uses wavelet technology. Dirac is flexible due to its reduced encoder complexity and the compression efficiency of Dirac is similar to that of H.264 [1] [26]. The performance of Dirac video codec was analyzed and the video obtained was noticed to be of good quality at a considerably low bit rate. Taking advantage of its good quality, reduced complexity and open platform technology, Dirac is chosen to be the video codec in this thesis. [1] [6]

2.1 Dirac encoder

Dirac video codec is a conventional hybrid motion compensated video codec similar to MPEG standards. It is hybrid because it incorporates transform and motion compensation. Dirac uses wavelet technology for transforms. Image motion is tracked and this information is best used to make prediction of the later frame. Transform is applied to the residual and the transform coefficients are scaled, quantized and later entropy coded into fewer bits. The transform used removes any spatial redundancy in the data and motion compensation used removes temporal redundancy. Dirac uses arithmetic coding [33]

for entropy coding to pack the bits efficiently to a bit stream format. Figure 2.1 shows the block diagram of Dirac encoder. The decoder does inverse operations. [1]

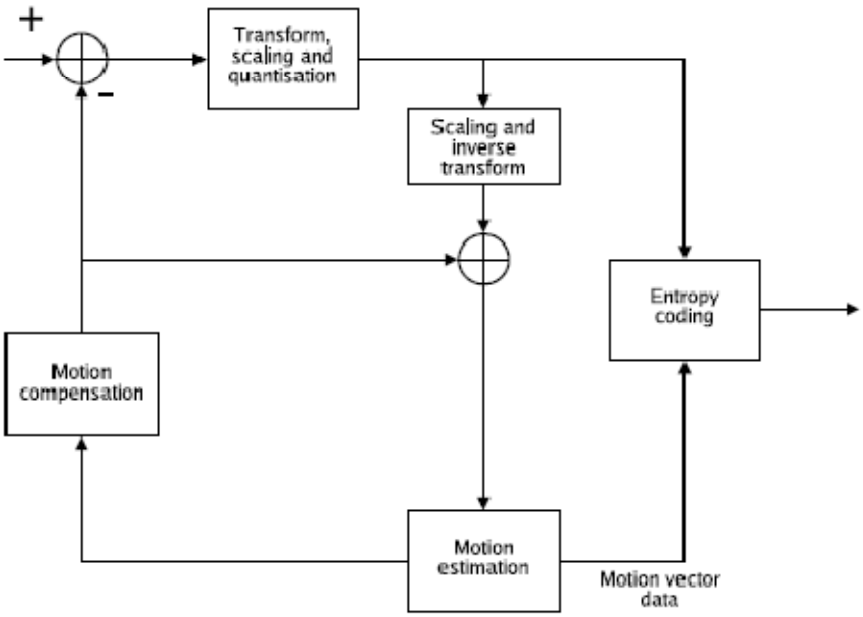

Figure 2.1 Block diagram of Dirac encoder [1] [6]

Some of the features Dirac video codec supports are:

- Direct support of multiple picture formats.
- From QCIF through ultra high resolution digital cinema.
- Direct support of 4:2:0/4:2:2/4:4:4 chroma sub-sampling format.
- Direct support of multiple bit depths, for e.g: 8 bit to 16 bit.
- Direct support of frame rates from 23.97 fps – 60 fps.
- Definable pixel aspect ratios.
- Definable ‘clean input’ area for inputs within larger containers.
- Definable signal ranges and offset.
- Definable wavelet depth.
- A choice from multiple wavelet filters (including filters optimized for down conversions). [27]

A brief description of each of the blocks used in the Dirac encoder is as follows:

2.2 Wavelet transforms

In contrast to other conventional MPEG standard video codecs using discrete cosine transforms (DCT) as block based transforms, Dirac uses wavelets. Wavelets have already proven superior to block based transforms for still image compression and are adopted in the JPEG2000 still image compression standard. Wavelets operate on the entire image at once and hence no blocky artefacts will be observed in case of Dirac. Figure 2.2 shows the stages of a wavelet transform. [1]

Figure 2.2 Stages of a wavelet transform [1]

The wavelet transform is constructed by repeated filtering of signals to low frequency and high frequency sub bands. For a 2-D signal, the filter is applied vertically and horizontally. Hence, the signal decomposes to low- low (LL), low-high (LH), high-low (HL) and high-high (HH) sub- bands. Since the low frequency component is significant, this undergoes multiple levels of decomposition based on the wavelet depth. By default, the wavelet depth is 4. The left most image is the original Lena image. The middle and right most images are obtained from level-1 and level 2 decomposition of wavelet transform. [1]

Figure 2.3 shows the original image; Dirac compressed image and MPEG-2 [19] compressed image with a compression ratio of 160:1. The presence of blocky artefacts in case of MPEG compressed image is observed while this is absent in case of a wavelet compressed image.

Figure 2.3 Comparison of Dirac with MPEG-2 in terms of compression. [1]

Original image (left), Dirac compression (middle) and MPEG-2 compression (right)
(Compression ratio: 160:1)

Figure 2.4 shows the frequency decomposition of wavelet transform to four sub-bands. Figure 2.5 shows the wavelet transform architecture. The only limitation for wavelet decomposition is that there should be exact number of macro blocks horizontally and vertically in which the dimensions are divisible by 16 ; typically of 4 × 4 block size. In order to overcome this problem, the data should be padded with zeros prior to encoding. But this is not yet implemented in Dirac. [2]

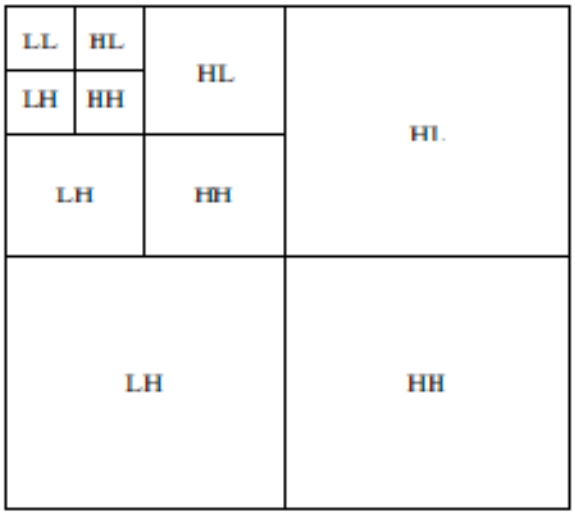

Figure 2.4 Wavelet transform frequency decomposition. [2]

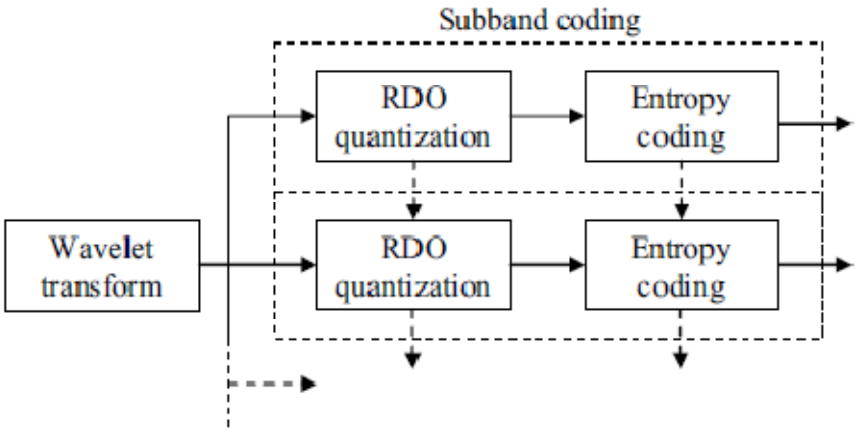

Figure 2.5 Dirac wavelet transform architecture [2]

2.3 Scaling and Quantization

The transformed co-efficients are then quantized, scaled and entropy coded. Each sub band co-efficients are quantized using a dead zone quantizer. Figure 2.6 shows the uniform and dead zone quantizers.

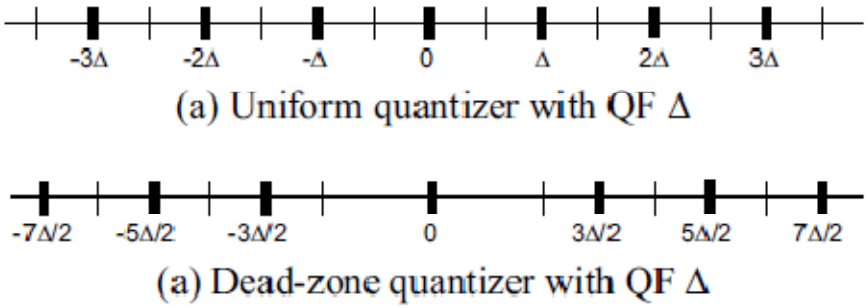

Figure 2.6 Uniform and dead zone quantizers. [2]

The quantization step size in case of Dirac is twice as that of regular uniform quantization process and this step size allows it to perform coarse quantization on smaller values. With LL sub band being significant, the prediction is done on a pixel basis. Current pixel (pel) under prediction is predicted by calculating the mean of the neighboring pels. Coefficient prediction is carried out on the transformed data

to remove residual dependencies if any by making entropy coding of data very effective. The difference is quantized and added at a later stage during reconstruction. [2]

The quantization process is separated into three stages:

- A first quarter of co-efficients are used to obtain bit- accuracy.
- A second quarter are used to estimate half bit accuracy and
- The resulting half pixels are used to perform quarter bit accuracy. [2]

2.4 Entropy coding

Figure 2.7 shows the block diagram of entropy coding. Entropy coding is carried out in three stages namely,

- Binarization.
- Context modeling and
- Arithmetic coding (AC). [2]

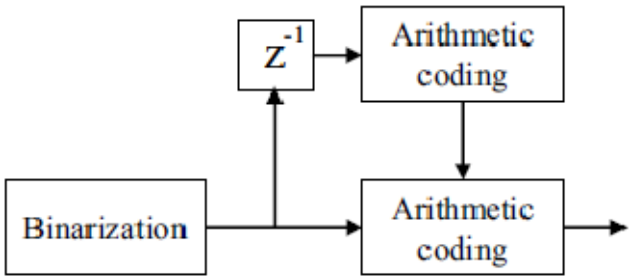

Figure 2.7 Block diagram of entropy coding [6]

2.5 Motion estimation

Motion estimation exploits temporal redundancy in video streams by looking for similarities between adjacent frames [6]. Dirac implements hierarchical motion estimation and consists of Intra (I) frame, L1 (Predictive - P) and L2 (Bi-directional - B) frames. I frames are independent and are always used as reference frames for L1 and L2. L1 and L2 frames are inter frames and depend on previously

coded frames for reference. L1 frames are temporarily used as reference while L2 frames are like B frames and never used as reference for coding any frames. Each Dirac frame can use up to a maximum of two frames for reference [6].

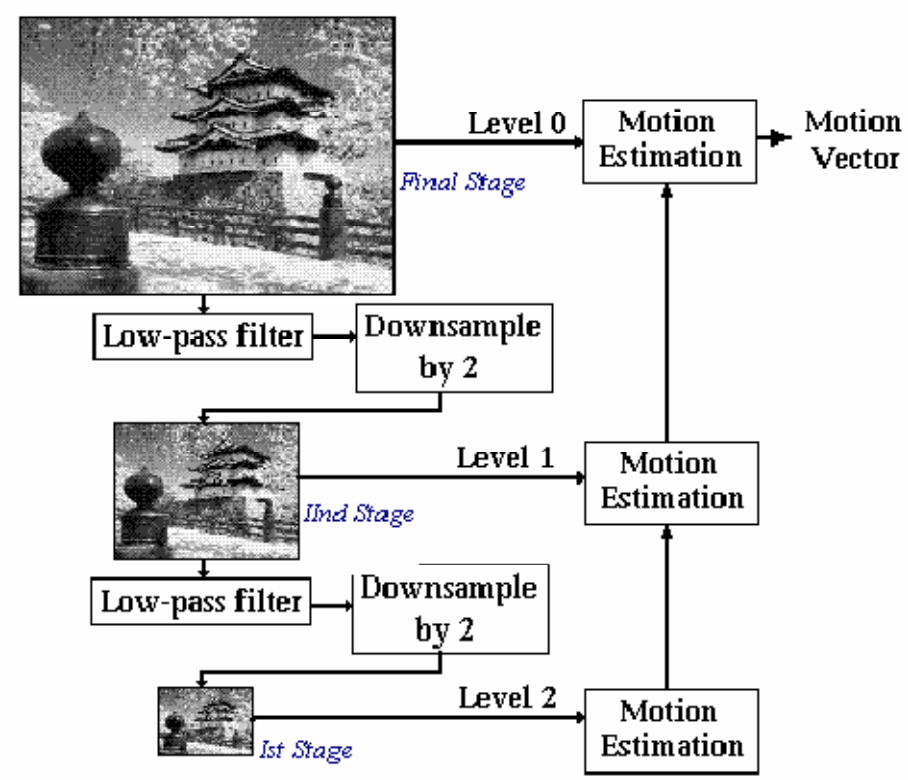

Figure 2.8 Hierarchical motion estimation [30].

2.6 Motion compensation

Motion compensation is followed after motion estimation. Here, the motion vectors are used predict the current frame. Dirac uses overlapped block- based motion compensation (OBMC) to avoid block- edge artifacts which would be too costly to code using wavelets. Sub-pixel motion compensation with motion vector precision up to $1/8^{th}$ pel accuracy can be obtained using Dirac. [6]

2.7 Dirac bit-streams

The encoded Dirac bit stream begins with a sequence header followed by the byte stream data in the form of frame data and ends with end of a sequence. Dirac bit-stream is concatenation of many such sequences. The data in the Dirac corresponds to a single video stream with constant parameter sets required for decoding. [23]

The start of the frame data is signaled by the parse info header. Parse info headers shall contain 4 byte code so that the decoder can be synchronized with the stream. A Dirac video sequence consists of an alternating sequence of parse info headers and data units. Data unit can be sequence header, a picture, auxiliary data or padding data. Each Dirac sequence consists of at least one sequence header and one picture data. The first picture data in a Dirac sequence is always an I picture. Dirac supports Inter and intra coding of pictures. Intra predicted macro blocks can be present even in inter predicted frames. If the sequence has more than one sequence header, the data will remain the same amongst the all the available sequence header (byte-byte identical). Since, auxiliary and padding data do not contribute to the decoding, they can be discarded. [23]

Parse info header occurs at the beginning of the sequence, before each picture data indicating the start of any Dirac picture and before each data unit. The parse info header consists of 13 bytes. The first 4 bytes namely parse info prefix i.e. 0x42, 0x42,0x43 and 0x44 signals the start of the picture data, the fifth byte i.e. the parse code signals type of picture i.e. whether I, P or B pictures. Next parse offset consists of 4 bytes signaling the data from first byte of current parse info header to the first byte of the next parse info header, if present. It shall be zero, if there is no further parse info header. Previous parse offset consists of 4 bytes signaling the data from first byte of current parse info header to the first byte of the previous parse info header, if present. It shall be zero, if there is no further parse info header. The parse info prefix, next parse offset and previous parse offset values support navigation to the decoder and are not required to decode the sequence. The parse offset values are generally non- zero and will be zero at the beginning and end of the elementary stream. [23]

Table 2.1 Parse codes. [23]

state[PARSE_CODE]	Bits	Description	Number of Reference Pictures
Generic			
0x00	0000 0000	Sequence header	–
0x10	0001 0000	End of Sequence	–
0x20	0010 0000	Auxiliary data	–
0x30	0011 0000	Padding data	–
Core syntax			
0x0C	0000 1100	Intra Reference Picture (arithmetic coding)	0
0x08	0000 1000	Intra Non Reference Picture (arithmetic coding)	0
0x4C	0100 1100	Intra Reference Picture (no arithmetic coding)	0
0x48	0100 1000	Intra Non Reference Picture (no arithmetic coding)	0
0x0D	0000 1101	Inter Reference Picture (arithmetic coding)	1
0x0E	0000 1110	Inter Reference Picture (arithmetic coding)	2
0x09	0000 1001	Inter Non Reference Picture (arithmetic coding)	1
0x0A	0000 1010	Inter Non Reference Picture (arithmetic coding)	2
Low-delay syntax			
0xCC	1100 1100	Intra Reference Picture	0
0xC8	1100 1000	Intra Non Reference Picture	0

Table 2.1 shows the possible parse code values for a Dirac sequence. In this thesis, the generic and core syntax fields are used and are relevant to Dirac. The low- delay syntax is required for intra coding of pictures using VC-2 standardized by Society of Motion Pictures and Television Engineers (SMPTE) and not used in this thesis. Auxiliary and padding data is not required and hence discarded. The fields used are described below:

Sequence header – Start of a Dirac sequence. [23]

End of a sequence – End of a Dirac sequence. [23]

Intra reference and non-reference pictures correspond to I frames with arithmetic coding (AC) and without AC. Since, Dirac uses AC for entropy coding the fields without AC for Intra coding are not required in this thesis. Inter reference picture with AC corresponds to L1/P pictures while Inter non-reference picture with AC corresponds to L2/B pictures.

2.8 Dirac decoder

Decoder performs the reverse operation of encoder. The encoded data is entropy decoded, scaled, inverse quantized and inverse transformed to obtain the reconstructed data. Figure 2.9 is an approximated block diagram of the decoding process understood in Dirac.

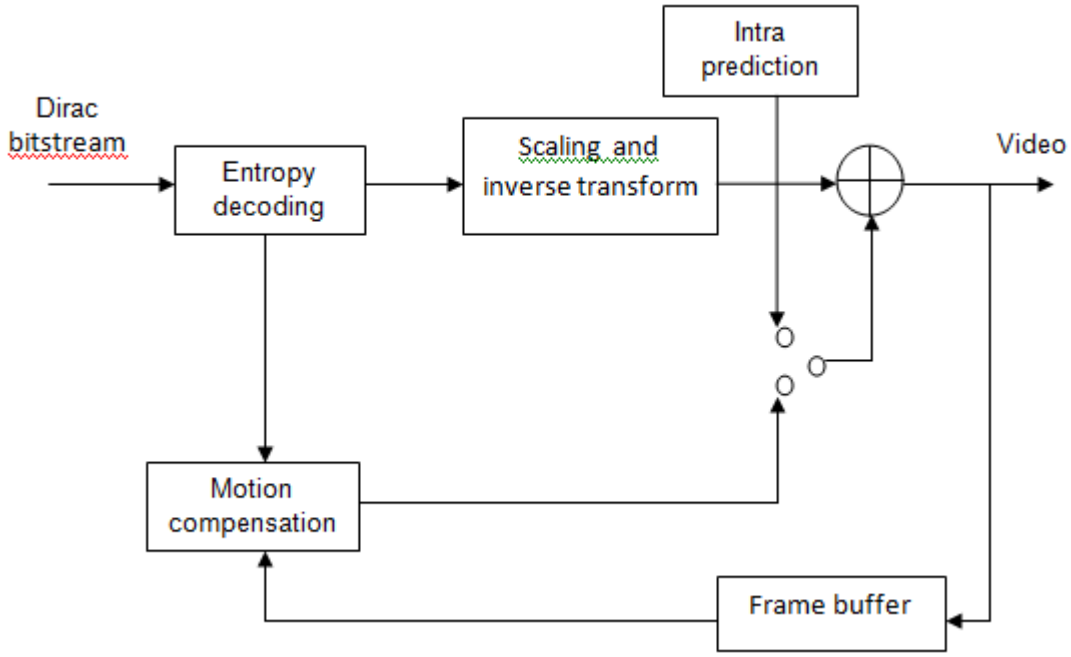

Figure 2.9 Block diagram of Dirac decoder

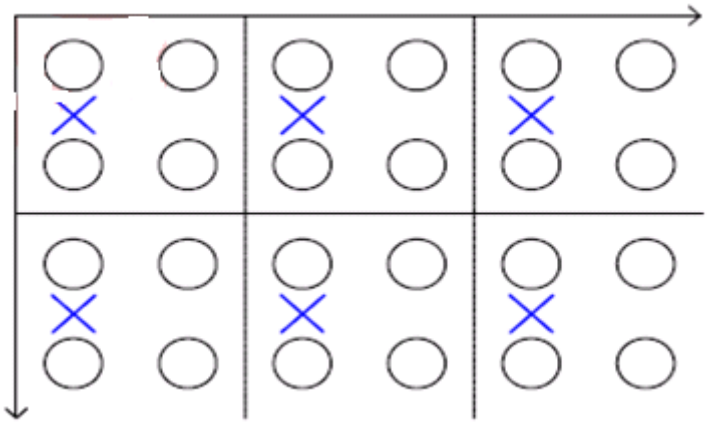

Figure 2.10 4:2:0 chroma sub sampling [51]

2.9 Summary

In this chapter, the need for Dirac video codec, the blocks of encoder, the nature of Dirac video bit stream describing frames as video elementary data and the decoder are explained. Chapter 3 discusses about the need for AAC audio codec, the encoder architecture, profiles and the nature of bit-stream syntax required for multiplexing.

CHAPTER 3

OVERVIEW OF ADVANCED AUDIO CODING

3.1 Advanced audio coding

Advanced audio coding [4], is a MPEG audio coding standard employing the perceptual audio coding technique. It is one of the popularly used audio codecs. AAC emerged as a standard for a high quality multichannel audio jointly from AT&T Corporation, Dolby laboratories, Fraunhofer institute for integrated circuits (Fraunhofer IIS) and Sony [22]. The performance of AAC audio codec is superior at bitrates greater than 64Kbps and also at lower bitrates (16 Kbps) and hence adopted in this thesis. This supports a wide range of sampling frequency ranging from 8 kHz – 96 kHz and supports a wide range of bit rate 16 – 576 kbps [5]. Additional tools like, temporal noise shaping, backward adaptive linear prediction and enhanced joint stereo coding techniques adopted in AAC helps in achieving good quality at low bitrates [11]. AAC fulfills the ITU-R/EBU requirements for indistinguishable quality of audio at 128 kbps/stereo [34]. Approximately, 30% more coding power than former MPEG, layer III [34].

In this thesis, the low complexity profile of AAC is used. AAC consists of two bit stream formats namely; Audio Data Interchange Format (ADIF) and Audio Data Transport Stream (ADTS). In this thesis the ADTS format is used.

3.2 AAC Profiles

A modular approach is used by the AAC standard. Three profiles are defined in AAC and the implementer selects the type of profile based on type of the application and produces relevant performance as per the application. They are main, low-complexity and scalable sampling rate profile. [4]

- Main profile: This is the most complex profile amongst all profiles, as it consists of all the encoding and decoding tools except for the gain control module. Thus, provides superior audio quality. This profile is apt only when there are no memory constraints to be met.

- Low complexity profile: In this profile, there can be considerable amount of savings in terms of physical memory and processing requirements. Also, the quality of audio obtained is nearly the same as that of main profile. Prediction tool is deleted in this profile. Generally, for practical applications this is most suitable as memory constraint is to be met. Also, complexity of temporal noise shaping tool is reduced.
- Scalable sampling rate profile: Flexibility for low complexity and scalable applications are provided in this profile. If bandwidth requirement is less and has to be met, this profile is apt for such applications. This profile adds the gain control module to the low complexity profile.

3.3 AAC encoder

Figures 3.1 and 3.2 show the block diagram of AAC encoder and AAC decoder respectively. The encoder comprises of a wide range of blocks and functionality of each block is described below [22]. The decoder does the inverse operation of encoder. AAC includes some of the powerful tools which help in achieving high audio quality at a lower bit rate [11]. The control flow path and data flow path is clearly differentiated in the figures 3.1 and 3.2.

Pre-processing: Initially, the input signal is pre-processed, undergoing multiple layers of filtering. The audio sample data is split to several blocks. Then the data needs to be smoothly traversed throughout the path in this module for further processing. In order to have smooth transition block by block, and to modify the data in each one of these blocks, a time domain filter, called windows provides navigation. Modified discrete cosine transform (MDCT) is applied in order to modify the data [4]. AAC effectively handles the situation in case of transients. AAC switches dynamically between two block lengths of 2048 and 256 samples referred as long and short blocks, respectively when the audio oscillates between steady state and transient signals. Depending on the signal complexity, AAC also switches between two different long blocks sine function and Kaiser-Bessel derived (KBD).

Temporal Noise shaping (TNS): TNS plays an important role in effectively handling the input audio signal in case of transient signals and signals intermediate to steady and transient state. The presence of transients in the audio data generates noise in the signal. The severity of the generated noise

depends on location of the transient present. Hence, TNS provides enhanced control of the location of theses transients in the filter window. Hence, TNS masks the transient like signals when present aiding in preventing inaudibility of noise and reduction in quantization noise in the steady state region and allows more bit allocation for describing non-transient signals in the block. TNS can be applied to part of a spectrum or fully, such that time domain quantization can be controlled in frequency domain.

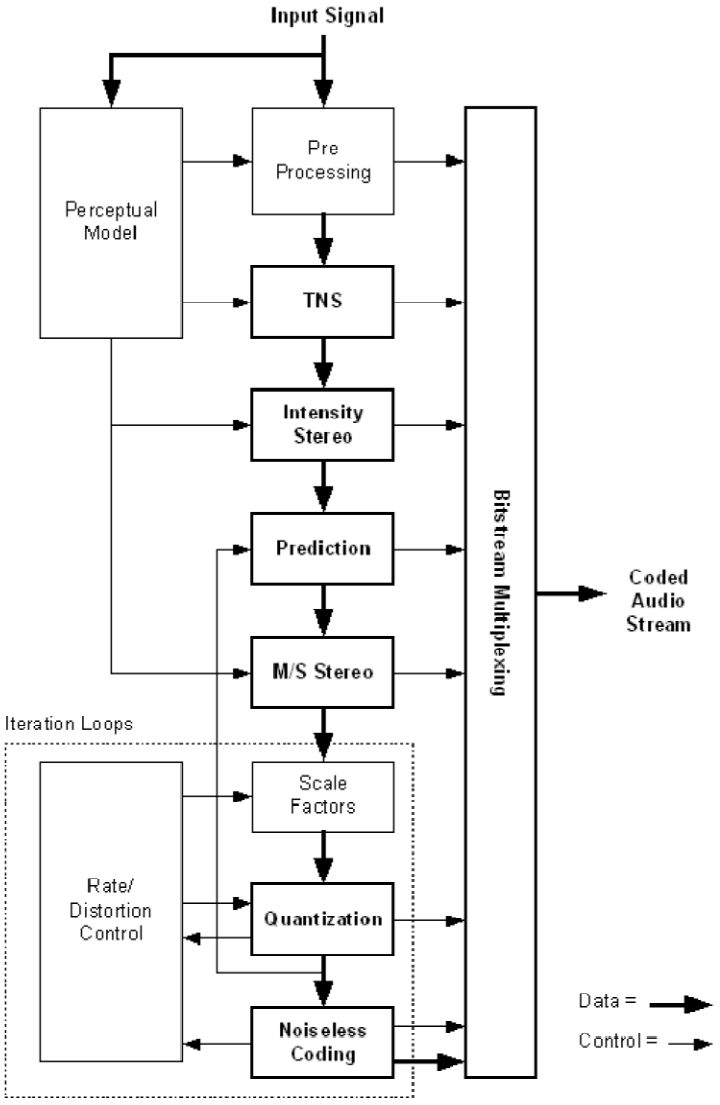

Figure 3.1 Block diagram of AAC encoder. [4] [37]

Perceptual coding: An audio signal consists of low frequency and high frequency components. If the frequency of the spectrum is too high or too low below the range of frequency perceptible to human ear, fewer number of bits can be used to encode such frequency component. Perceptual coding module intends to perform this task.

Intensity stereo coding: Sharing a single set of spectral values for high frequency components in case of a stereo channel pair is allowed in this module. This is achieved with no compromise on the audio quality.

Prediction: Prediction is carried out on a block by block basis. The block under prediction uses spectral component of preceding blocks for prediction. It is based on second - order backward adaptive process. Stationary and semi-stationary signals of an audio are represented by a prediction module. When stationary or a semi-stationary signal is encountered, the number of bits is reduced by passing a repeat instruction instead of repeating the same for subsequent windows.

Mid/Side (M/S) stereo coding: This is based on channel pair coding. It also helps in reducing the amount of data necessary for coding. The channel pair elements are analyzed on a block by block basis as right/left and sum/difference.

Quantization and coding: It is during this process the majority of bits can be reduced although the earlier modules provide some compression. In AAC, quantization is achieved by control of psychoacoustic model. The bits used in the process should be below the value determined by this model. Huffman coding is used for entropy coding.

Noiseless coding: This module is embedded in quantization and coding module. Prior to entropy coding, the noiseless dynamic range can be applied.

3.4 AAC decoder

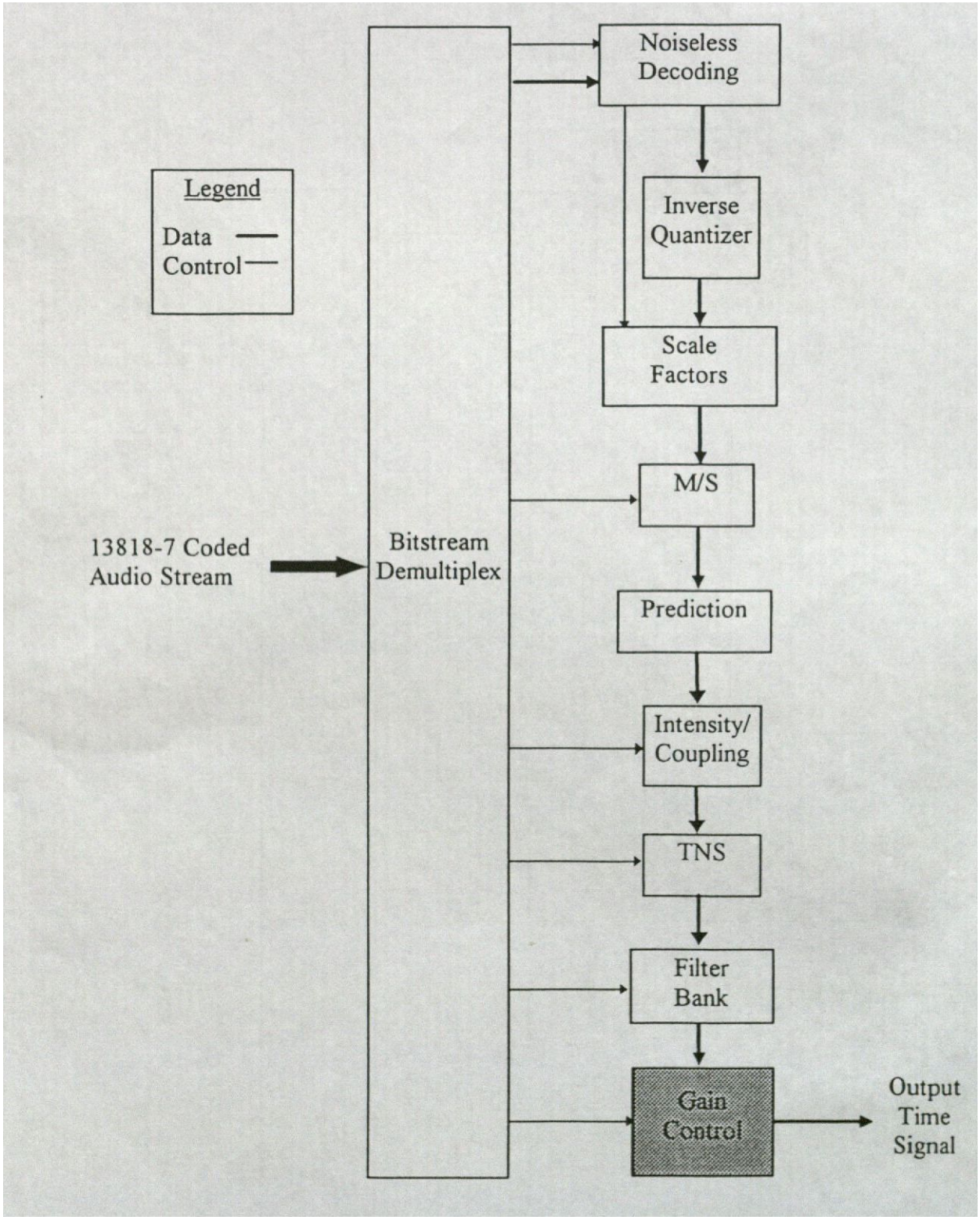

Figure 3.2 Block diagram of AAC decoder. [4] [37]

3.5 AAC audio bit stream

AAC has two bit stream formats ADIF and ADTS. ADTS header is used in this thesis, since it comprises of a header for each frame data and is error robust for streaming environment. Since, ADIF consists of a single header followed by entire frame data and due to its nature of being a single stream it is not adopted. Table3.1 tabulates the header syntax of ADTS bit stream format. Table 3.2 tabulates the profile bits expansion of AAC. [4] [37]

Table.3.1 ADTS bit stream header format [36] [37]

Field	Number of bits	Description	
syncword	12	Always '111111111111'	ADTS fixed header
ID	1	0:MPEG-4 1: MPEG-2	
layer	2	Always"00"	
protection_absent	1		
profile	2	Explained below	
sampling_frequency_index	4		
private_bit	1		
channel_configuration	3		
original/copy	1		
home	1		
copyright_identification_bit	1		ADTS variable header
copyright_identification_start	1		
aac_frame_length	13	Length of frame including header (in bytes)	
adts_buffer_fullness	11	0x7FF indicates VBR	
no_raw_data_blocks_in_frame	2		
crc_check	16	Only if protection_absent ==0	
raw_data_blocks		Variable size	

The ADTS bit stream always begins a sync word. Table.3.1 indicates the field of the fixed and variable header of ADTS bit stream format. ADTS fixed header remains same for each of the audio frames while ADTS variable header varies for each audio frame.

Table.3.2 AAC profile bits expansion. [36] [37]

Profile bits	Bits ID == 1 (MPEG -2 profile) ID ==0 (MPEG-4 object type)
00(0)	Main profile
01(1)	Low complexity profile
10(2)	Scalable sampling rate
11(3)	reserved

3.6 Summary

In this chapter, the need for AAC codec, the encoder, decoder, profiles and type of bit stream format selected are explained. Chapter 4 describes in detail the multiplexing process.

CHAPTER 4

MULTIPLEXING

4.1 Need for multiplexing

A program contains one video elementary stream and one audio elementary stream. The use of data in the form of subtitles is optional along with a multimedia program. When data is broadcast over a medium using standards like ATSC-H, ATSC-M/H [17] [18], DVB [21] or IPTV, it cannot be sent separately and needs to be transmitted as a single stream to meet the requirements necessary for transmission. Hence, the audio-video data needs to be multiplexed as a single transport stream and then transmitted. Thus, the need for multiplexing the video-audio data is substantiated. During transmission of the data, superior quality, high resolution and efficient use of bandwidth need to be achieved at lower bitrates. So, the use of the Dirac video codec and the AAC audio codec for encoding video and audio bit-streams respectively is necessary. Figure 4.1 demonstrates the ATSC standard for transmission and reception of data using MPEG-2 TS multiplexer [20].

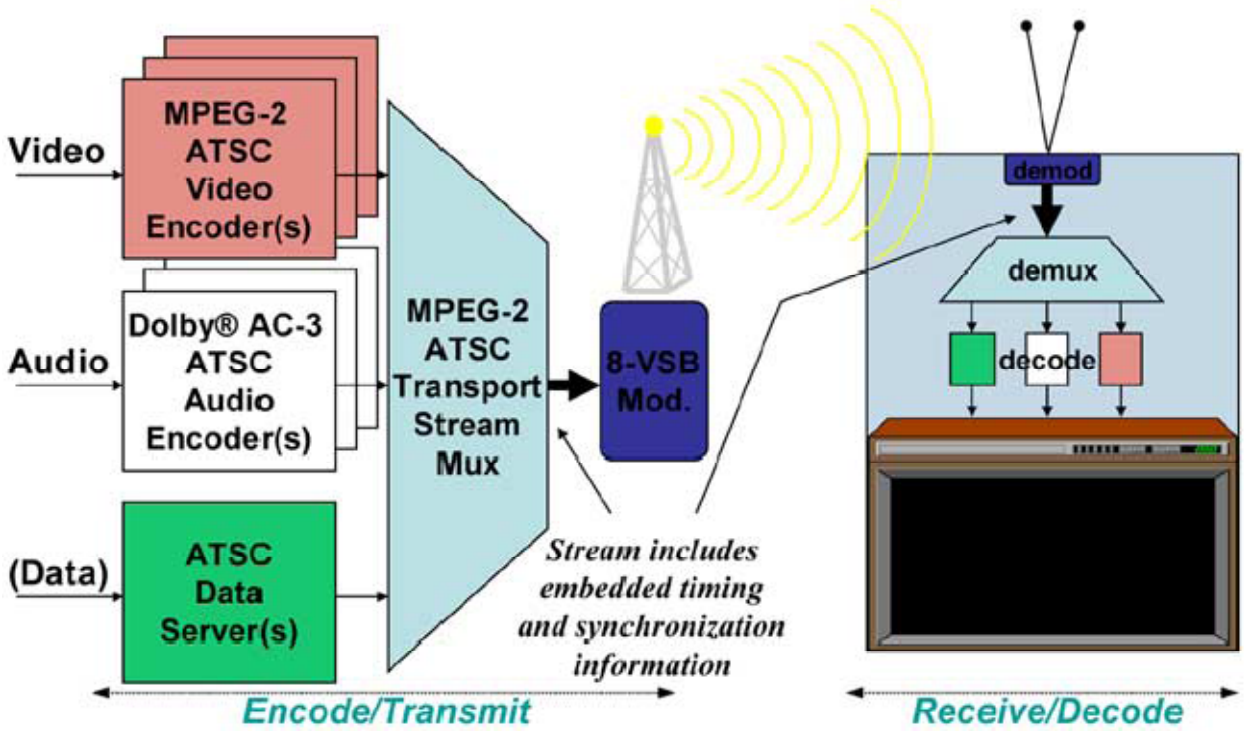

Figure 4.1 Block diagram of ATSC transreception showing MPEG-2 TS multiplexer [20]

The MPEG-2 standard transport stream specification is used to multiplex the data in bit-stream format [19]. The factors that need to be considered during multiplexing of data as a single stream are [22]:

- Each one of video/audio elementary stream data needs to be prioritized equally. This is essential to maintain buffer fullness at the receiver's end.
- In addition to carrying payload information consisting of audio/video frame data, the information in the form of time stamps is embedded to the PES header which will assist in maintaining lip sync and also the information to help playback of the data in a sequential manner is included.
- Finally, since the multiplexed data will be transmitted in an error prone network, a provision for error detection and correction methods is to be made.

4.2 Packetization

Packetization is the first step in the process of multiplexing. The video and audio elementary stream data needs to be packetized for reliable and efficient transmission. Several multimedia programs need to be handled i.e. many video and audio frame data. In order to meet the transmission channel requirements the audio/video data undergoes two layers of packetization. First layer of packetization is called packetized elementary streams (variable length) and this further undergoes another layer of packetization namely, transport stream (fixed length - 188 bytes). The packet consists of a header and a payload. The payload consists of the frame data and header consists of additional information required to determine the type of packet (video or audio) and also assist in achieving synchronization. Multiplexing is carried out after TS packets are obtained. [19]

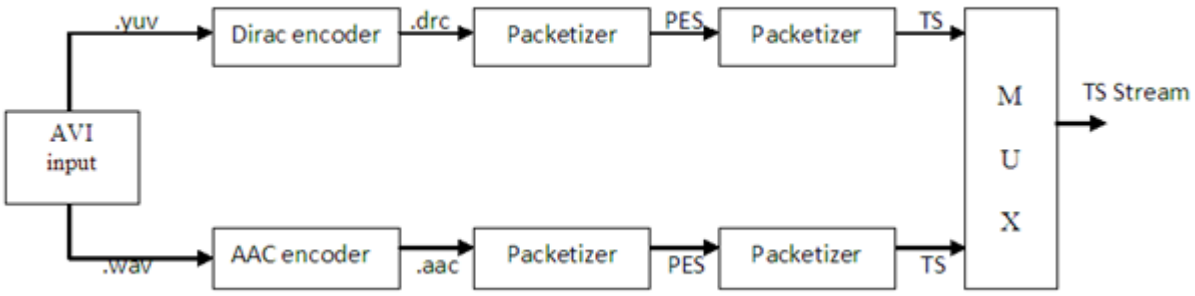

Figure 4.2 Two layers of packetization.

4.3 Packetized elementary streams

The packetized elementary streams consist of a header and a payload. The encapsulation of audio/video data is performed by separating elementary streams into access units. Access units are nothing but the encoded audio/video frame data as elementary streams. This forms variable length packets each consisting of I, P or B pictures in the case of a video and a block of frame data in the case of audio. The PES header consists of 8 bytes carrying time stamp information in the form of frame numbers and additional information to facilitate in multiplexing the data [19].

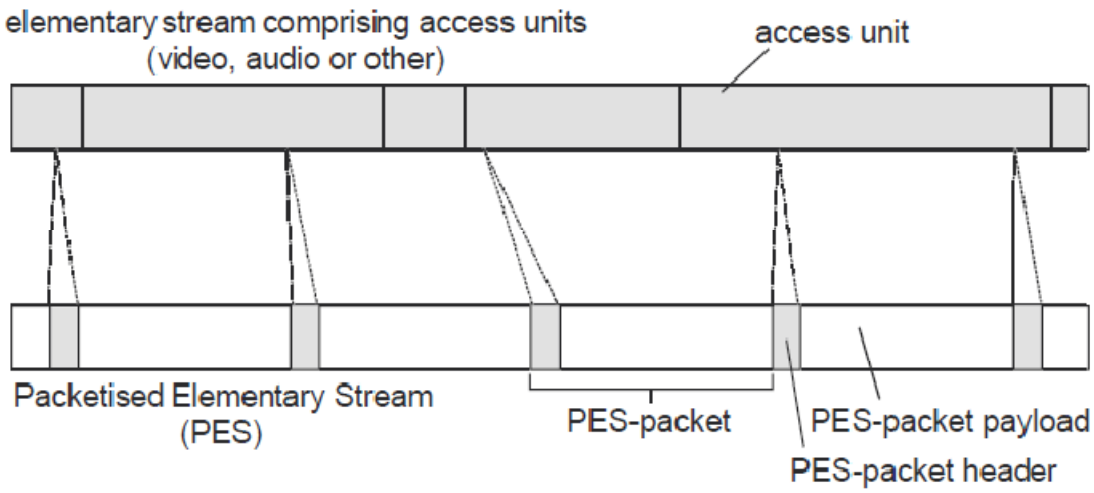

Figure 4.3 Encapsulation of PES from elementary streams [9]

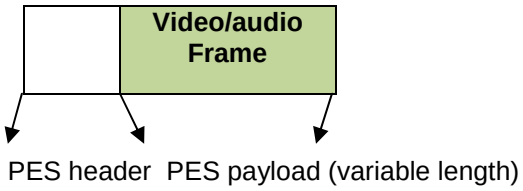

Figure 4.4 Structure of PES packet

Table 4.1 describes the PES header fields. The first 3 bytes (0x000001) represents the start of a PES packet. The byte following this distinguishes between an audio and a video stream ID. 0xD1 is set for a video stream ID [7] and 0xC1 is set for an audio stream ID chosen from a range of 0xC0 – 0xDF [19]. The first 4 bytes together is known as the start code, which determines that the PES packet is valid. The next 2 bytes of data hold the PES packet length information which is signaled from the Dirac byte stream parse info header [23]. If the PES packet length is neither specified nor if unbounded i.e. (exceeds 65536 bytes), value of zero is indicated is this field [8]. Assuming, that the PES packet length does not exceed the allocated space, the reset option is not included in the algorithm implemented. Lastly, the final 2 bytes of the PES header points to the time stamp information in the form of frame numbers which helps in achieving sync during playback. This is discussed later in section 4.5 of this chapter.

Table 4.1 PES header description [19]

Header fields	Size (Bytes)	Description
PES packet start prefix	3	0x000001
Video/audio stream ID	1	Unique for Video/audio
PES packet length	2	Unbounded when exceeds 65536 bytes and value of 0 is indicated under such circumstances.
Time stamps	2	Frame numbers used a s time stamps

The 2 byte frame number is calculated when a video/audio elementary stream begins. The payload information is then encapsulated into the PES header.

In the case of PES video encapsulation, the video elementary stream is searched for a 4 byte parse info prefix 0x42, 0x42, 0x43 and 0x44 followed by a byte of start code which indicates the type of picture. The first 4 bytes indicate the start of a PES payload. The picture type can be a frame or field based on the format used (progressive or interlaced). In this thesis, the progressive format is used. Table.2.1 of chapter 2 was used to determine the type of start code, i.e. an I, P or a B picture. The payload information is encapsulated in the header to form a video PES packet. [23]

In the case of PES audio encapsulation, the audio elementary stream is searched for a 12 bit header '111111111111', which indicates the start of a frame and PES payload data. This is encapsulated with PES header to form an audio PES packet. Each PES packet consists of one frame (video/audio) data only.

Both audio and video PES packets are obtained for the entire encoded video /audio bit-stream. Thus, first layer of packetization of variable length is obtained and the video/audio PES packets are ready to undergo a second layer of packetization, namely transport streams.

4.4 Transport streams

MPEG-2 [19] systems specification describes how the encoded video, audio and data streams (optional) if present can be contained in a single stream suitable for digital transmission or storage. MPEG-2 systems define two alternative multiplexes, namely, program streams and transport streams. Program streams are suitable for a error-free medium due to its susceptible nature to errors and hence transport streams are adopted in this thesis as it is designed for multi-program applications like broadcasting, so that single transport stream can hold multiple programs. [9]

TS consists of fixed length packets each 188 bytes long. The audio/video PES packets is encapsulated as TS payload while TS header consists of 3bytes/4 bytes header based on the status of a flag bits present in the TS header. Each TS packet payload consists of information of only one audio or video PES packet at a time.

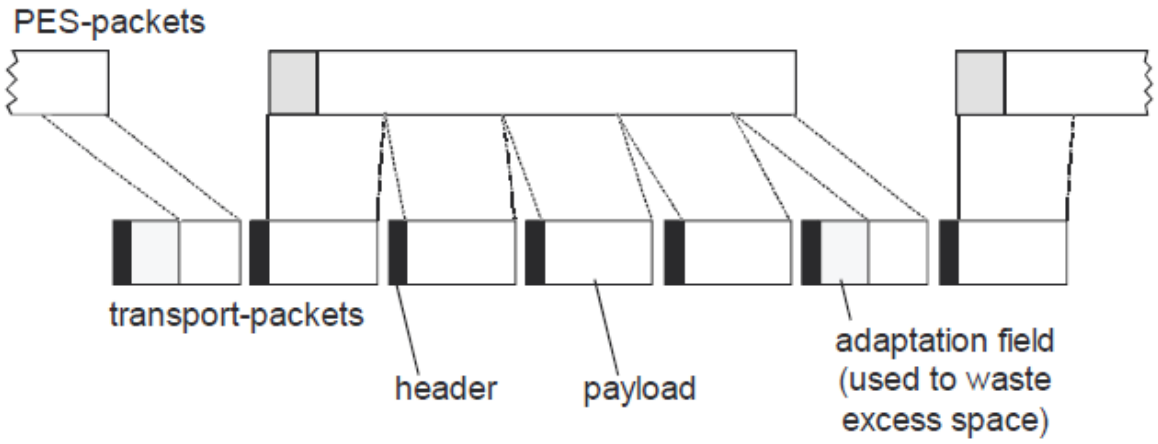

Figure 4.5 Splitting of PES packet to several TS packets [9]

Figure 4.5 shows the division of PES packet to several TS packets. Figure 4.6 shows the MPEG transport stream and structure of MPEG- TS packet.

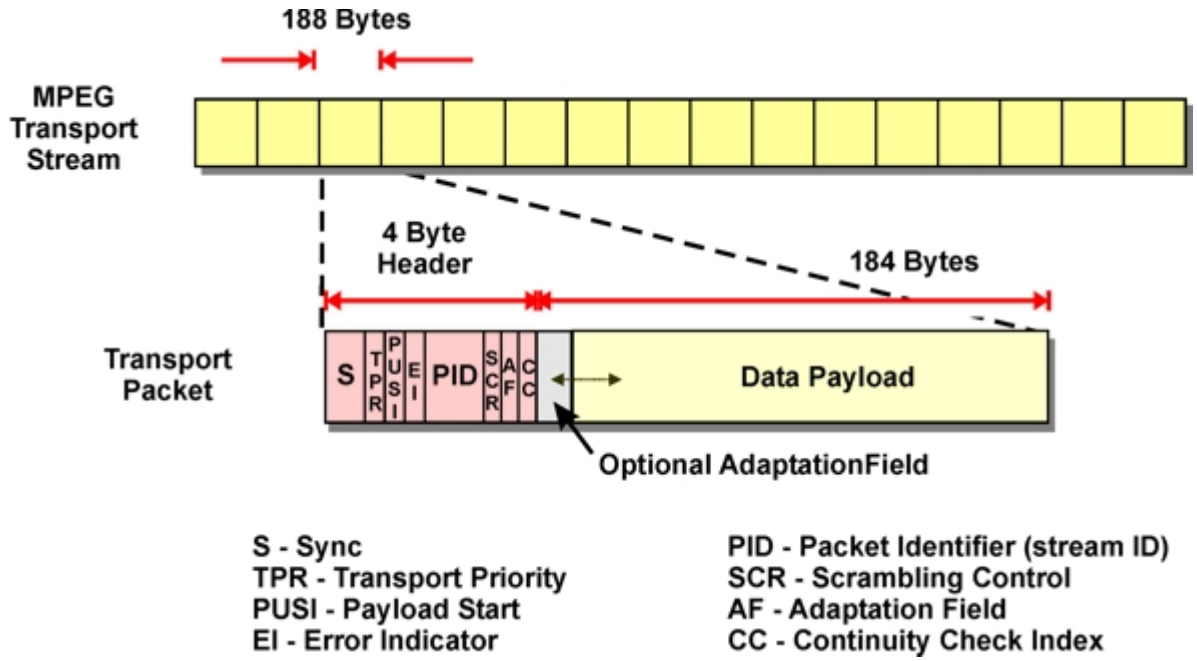

Figure 4.6 MPEG transport stream and standard structure of MPEG-TS packet [12] [24].

Figure 4.7 shows the structure of TS packet consisting of TS header and payload. This clearly indicates that the TS packet size is fixed and is 188 bytes long.

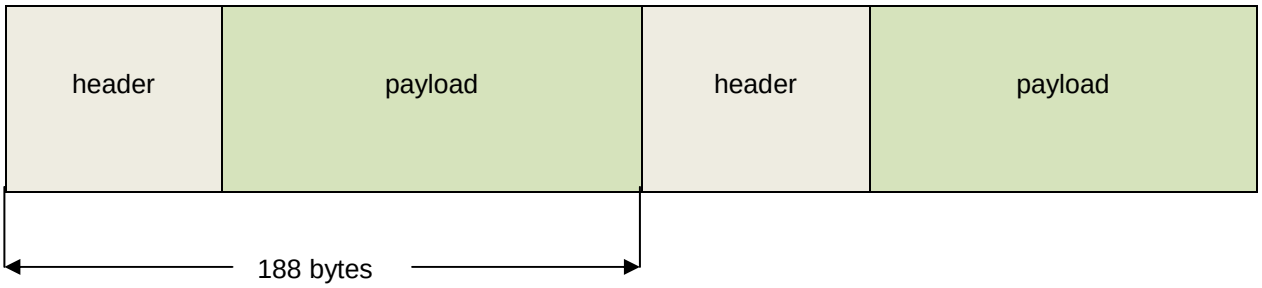

Figure 4.7 Structure of TS packet.

Figure 4.8 describes the structure of TS packet header. TS packet header consists of multiple fields and the number of bits representing each header fields is indicated clearly.

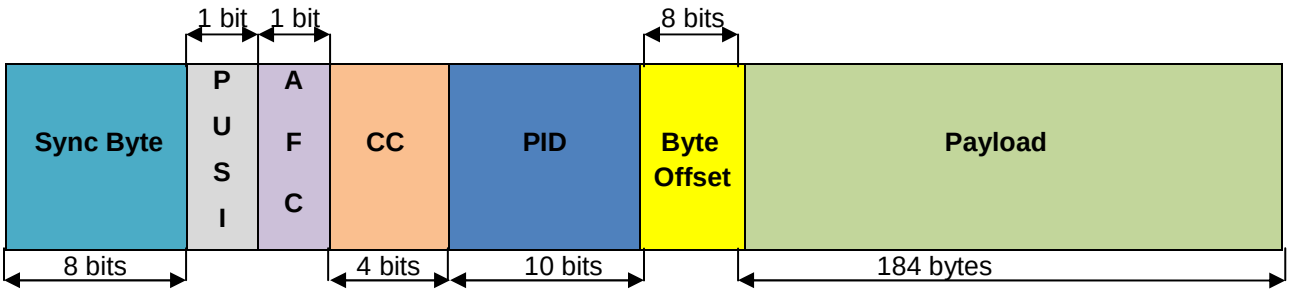

Figure 4.8 Structure of TS packet header used in this thesis.

The TS header consists of sync byte, payload unit start indicator (PUSI), adaptation field control (AFC), continuity counter (CC) and byte offset field (optional). The payload consists of PES packet data. The table.4.2 tabulates the bits allocated for a TS packet header. [19]

Table.4.2 TS packet header

Header fields	Size
Sync byte	8 bits
PUSI	1bit
AFC	1bit
CC	4 bits
PID (Unique for video/audio)	10 bits
Byte offset (optional)	8 bits (If AFC ==1 && LPES < 185 bytes)
Stuffing bytes	Added in case of byte offset

Note: LPES - Length of PES packet.

Each one of the TS header fields used is explained below:

- Sync byte: Sync byte indicates the start of any TS packet. The value of sync byte is set to 0x47 to indicate the start of TS packet. [9]
- Payload unit start indicator: PUSI is a 1 bit flag field which assists in determining the start of PES packet data in the payload of a TS packet. If the flag bit is set to '1', it means that the TS payload consists of a start of PES packet else if the bit is set to '0' it means that TS packet consists of the PES data corresponding to the same PES.[9]
- Continuity counter: CC is a 4-bit field which increments from '0000' to '1111' every time it encounters a TS packet. If the value reaches '1111', the counter is reset back to zero and repeats itself in case of a longer PES packet aiding in determining packet loss if any at the receiver's end. The CC value is also reset back to '0000' when it encounters a TS packet of a new PES. [9]
- Packet identifiers: PID is unique for audio/ video PES and is 10-bits in length. It is assumed that decoder has the knowledge of PID. [9]

  PID for video: '0000001111'

  PID for audio: '0000001110'
- Adaptation filed control: AFC is also a 1 bit flag field which assists in determining the last TS packet for a given PES packet. If this flag is set to '1' it means that it is last TS packet of a given PES else if the flag bit is set '0' it means there are more TS packets following for a given PES. When AFC is set to '1' and if there are exactly 185 bytes of PES data then the byte offset field is absent. [9]
- Byte offset: This is an optional field and is used only when the remaining PES packet length is less than 185 bytes and AFC is set to '1'. This is done in order to maintain TS packet length to 188 bytes. Byte offset is calculated in this case as:

  Byte offset = 184 – remaining length of PES. Once, this is calculated the TS payload is stuffed with zeroes of Byte offset followed by the remaining PES data. [9]

Thus, the second layer of packetization is performed and the data is ready to be multiplexed.

4.5 Time stamps

MPEG-2 standard specifies methods to help in achieving synchronization namely, program clock reference (PCR) or frame numbers as time stamps. In this thesis, frame numbers as time stamps is adopted. [19]

It is known that frame rate remains constant for a video bit-stream and audio sampling frequency remains constant for an audio bit-stream. With this knowledge, the duration of a video frame and audio can be calculated if frame numbers are known. Hence, the concept of duration of a video/audio can be exploited and best used when frame numbers are used as time stamps. AAC standard defines that an audio frame consists of 1024 samples and sampling frequency can range anywhere from 8 kHz – 96 kHz. [5]

The playback time of video and audio can hence be calculated as:

$$\text{Video playback time} = \text{frame number}/\text{frame rate} \quad (4.1)$$

$$\text{Audio playback time} = (1024 \times \text{frame number}) / \text{sampling frequency} \quad (4.2)$$

Using (4.1) and (4.2), if frame number of video or audio is known, the audio or video frame corresponding to that frame number can be determined. The time of occurrence of both video and its corresponding audio can be easily computed. This is helpful in achieving lip synchronization during playback of audio and video which is crucial. Two bytes of header fields is allocated for time stamps. Hence, the frame numbers can go from 0 to 65535. If, the number exceeds the allocated space, it needs to be reset and started over again. When the frame number of one stream is reset the frame number of other stream also needs to be simultaneously reset in order to avoid ambiguity at the de-multiplexer. At no point there will be similar frame numbers be it in video or audio buffer. Since, the chance of frame numbers to exceed the allocated space is extremely slim to occur either in case of video or audio, this condition is not checked in the algorithm implemented.

The advantages of using frame numbers as time stamps over Program Clock Reference (PCR) are: [22]

- Less complex.
- Can be easily incorporated in software and is well suited.
- Savings in PES header bytes. (In case of PCR, time information needs to be sent periodically)
- Synchronization can be effectively achieved due to absence of clock jitters and inaccurate clock samples.

4.6 Proposed multiplexing method

With the availability of video and audio TS packets, the process of multiplexing can be carried out. In order to efficiently use the system memory, the audio/video TS packets are generated once the available TS packets are already transmitted. The criterion used to multiplex the data is the presentation time of audio and video TS packets. First, the number of TS packets generated for a given PES is calculated. With the knowledge of video and audio frame duration, the duration for each TS packet of a PES is calculated in both video and audio case. Counters for video and audio TS are maintained at the multiplexer. Based on the presentation times obtained, the TS packet with lesser presentation time is transmitted and the corresponding counter is incremented. This process continues until all the TS packets are transmitted at the multiplexer. By default, video TS is transmitted for the first time.

To simplify the understanding of the criterion used, an example has been used to explain the same.

For example:

Consider, we have a video PES packet of 500 bytes @ frame rate = 25 fps.

Number of TS packets for PES = 500/185 = 2.7 ~= 3 TS packets.

Video PES duration = 1/25 = 0.04 = 40ms.

Duration of each TS packet = 40/3 = 13.33ms.

Consider, we have an audio PES packet of 600 bytes @ sampling frequency = 44100 Hz.

Number of TS packets for PES = 600/185 = 3.2 ~ =4 TS packets.

Audio PES duration = 1024/44100 = 0.04 = 23.2ms.

Duration of each TS packet = 23.2/4 = 5.8ms.

The video and audio timing counters are initialized to zero. Hence, video TS is transmitted at first and the video timing counter is incremented to 13.33ms, while audio counter is zero. Then, the presentation time is again compared and now the audio duration is less and audio TS is transmitted. Then, the audio timing counter is incremented by 5.8ms. This process continues until all the video and audio TS packets are transmitted at the multiplexer's end.

The approach of using presentation time as criterion to multiplex a program to a single elementary stream is effective due to the fact that the video and its corresponding audio will be transmitted one after the other. This method will aid to handle buffer fullness at the de-multiplexer's end without buffer overflow or underflow at the receiver's end.

4.7 Summary

In this chapter, the process of multiplexing Dirac video and AAC audio bit streams are explained in detail along with the two layers of packetization. The timestamp information aiding in achieving lip sync at the decoder's side and its advantages are also discussed. Thus, the transmitted data needs to be de-multiplexed and decoded at the receiver's end to ensure lip sync. The method of de-multiplexing and achieving lip synchronization is discussed in detail in chapter 5.

CHAPTER 5

DE-MULTIPLEXING

5.1 De-multiplexing

De-multiplexing is the inverse process of multiplexing which means that the video and the audio elementary streams are retrieved from the multiplexed data. The payloads of video and audio TS packets are extracted to the video and audio buffers respectively. While doing so, the video and audio buffers need to be constantly monitored to ensure no buffer overflow or underflow of data takes place. The flowchart of the de-multiplexer algorithm is shown in figure 5.1.

The de-multiplexing process can be explained as follows. First, buffer for video and audio are maintained separately. Each incoming TS packet of 188 bytes long is read to check if it is a valid TS packet or not. If the first byte of TS is 0x47, the packet is valid and data needs to be read to buffer, else if invalid, the data is discarded and the next packet is read. The third byte (PID) is checked to determine whether it is a video or audio TS packet. If it is '15' it is a video packet else if '14' it is audio packet. The first two flag bits of second byte correspond to PUSI and AFC respectively and help in determining the start and end of a TS packet for the same PES. If the bits are set to '1' they indicate the start and end of TS packet corresponding to the same PES. If, PUSI is set to '1' and AFC is set to zero then TS packet corresponds to first packet of a PES and the 8 byte PES header is discarded and the payload is extracted to their respective buffer depending on PID. If, PUSI and AFC bits of a TS packet are both zero which means that there are more TS packets of the same PES, then the 185 bytes payload is accrued to the respective buffer. If PUSI is zero and AFC is '1, it means that it is the last TS of one PES then the fourth byte of TS packet which corresponds to offset byte is checked and the payload is extracted to the respective buffer discarding the offset zero bytes. When the last TS packet is encountered i.e. when AFC is '1', the frame numbers are also stored. When PUSI is '1' and AFC is also '1', it means that it is the only packet of a PES and the payload is extracted. This process of extraction of video and audio elementary streams is carried for the entire multiplexed stream. While extracting the data, the buffer fullness is

monitored at the de-multiplexing end continuously. If there is underflow or overflow of data, when the sequence is played back the audio-video will be out of sync. Absence of underflow or overflow of data ensures that the multiplexing algorithm was efficiently implemented and also the time stamp information embedded will help in achieving lip sync.

In this thesis, the buffer fullness is ensured while verifying the number of video and audio frames in the buffer. The frame numbers of the de-multiplexed data are stored in an array and the corresponding indices are also stored to keep track of the number of video/audio frames into their respective buffers. Thus, at various stages of data extraction into the buffer, the number of video frames to be present is chosen and the corresponding synchronized number of audio frames is also stored. The chosen video and audio content playback time are calculated using (5.1) and (5.2) respectively. Difference in the content playback time up to one frame duration is tolerable without causing out of lip sync problems. Using the proposed method, the buffer fullness is effectively handled preventing buffer underflow or overflow and the elementary stream data was extracted to the respective buffer. Results to ensure for buffer fullness, analysis of the results obtained and the output of the de-multiplexer are explained in detail in Chapter 6. In the de-multiplexing algorithm implemented, the video and audio data can be played from beginning to the end of the sequence as well as from any transport stream packet.

Video content playback time = Number of video frames × (1/fps) (5.1)

Audio content playback time = Number of audio frames ×(1024/sampling frequency) (5.2)

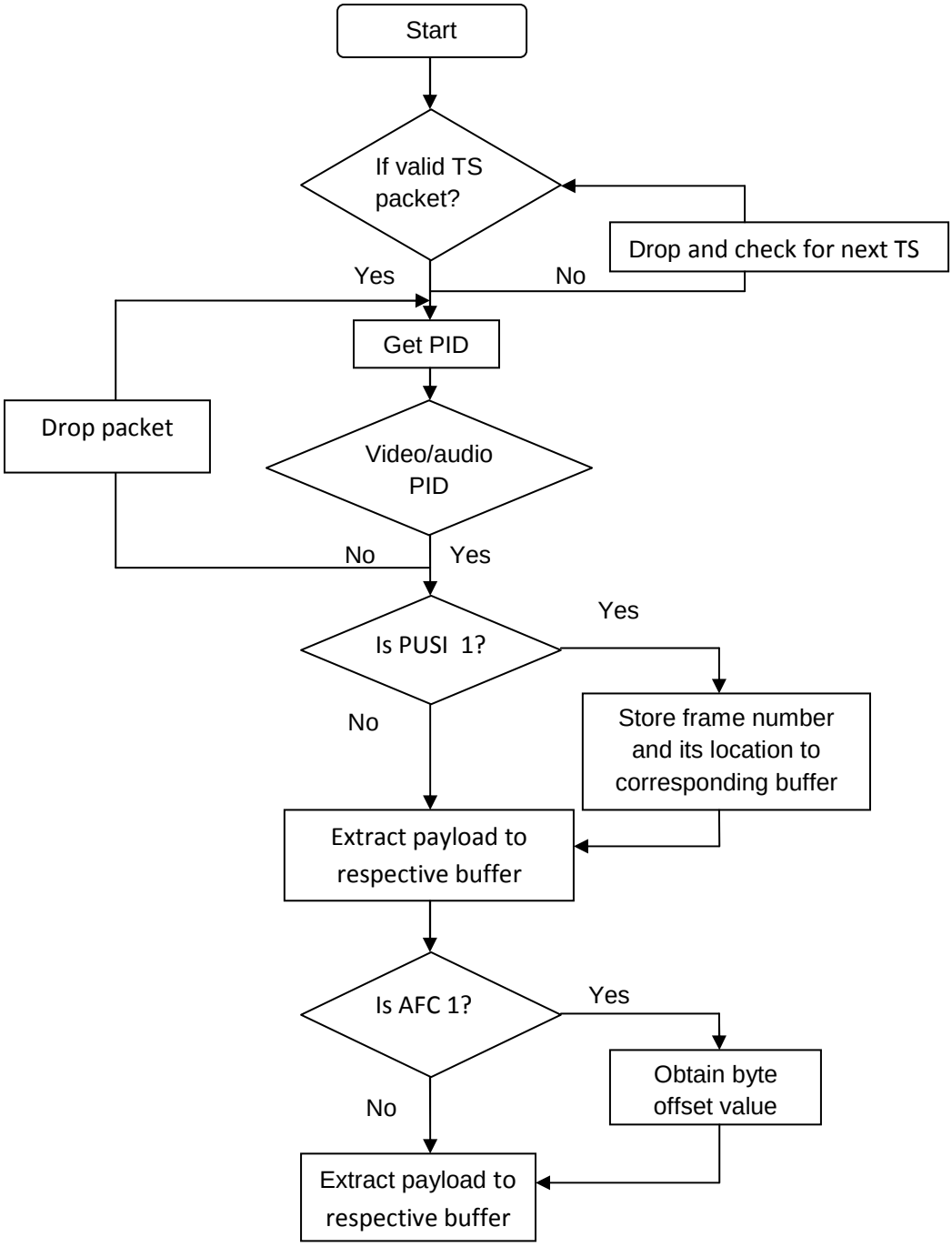

Figure 5.1 Flowchart of the de-multiplexer

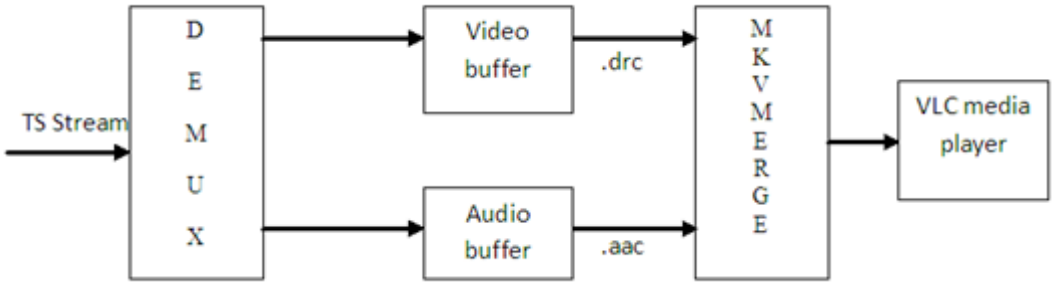

Figure 5.2 Block diagram of de-multiplexing process.

5.2 Lip synchronization and playback

Once, the data in the buffer is ready to be played back, the start of video and audio frame data needs to be determined. Since Dirac is used for video, in order to start decoding other than at the start of the sequence the decoder must first synchronize to the stream. The parse info prefix present in the parse info header supports the synchronization. So, the video buffer needs to be a searched for the parse info prefix which indicates the start of the parse info header. Once decoder is synchronized, the sequence header needs to be searched forwards or backwards in order to obtain the parameters required for decoding. Then, the Dirac video is ready for play back. In case of Dirac, if the picture data to be played back is dependent on the previous picture data some pictures may not be (completely) decodable. [23]

Since, AAC ADTS header format is used in this thesis, the audio data can be decoded from any audio frame. [25]

In this thesis, to check for visual delay between video and audio, the video buffer is searched for first Instantaneous Decoder Refresh (IDR frame) and the corresponding synchronized audio frame number is calculated using:

$$\boldsymbol{Audio\ frame\ number} = \frac{\boldsymbol{video\ frame\ number}}{\boldsymbol{fps}} \times \frac{\boldsymbol{audio\ sampling\ frequency}}{\boldsymbol{1024}}$$

$$\boldsymbol{Video\ presentation\ time} = \frac{\boldsymbol{Video\ frame\ number}}{\boldsymbol{fps}}$$

$$\boldsymbol{Audio\ presentation\ time} = \frac{\boldsymbol{1024 \times audio\ frame\ number}}{\boldsymbol{audio\ sampling\ frequency}}$$

If the audio frame number calculated is not an integer then the frame number calculated is rounded off to the nearest integer. The maximum round off error will be 0.5 times the duration of an audio frame [22]. The maximum delay that is allowed by MPEG-2 standard is ± 40ms [25]. If the difference between audio and video results in a negative threshold, audio lags video else, in case of positive threshold, audio leads video [52]. This delay that can be obtained due to round off error is the only limitation to this thesis. Once the audio frame number is obtained, the data corresponding to it is searched. If not found, then next IDR frame is searched and the corresponding audio frame number is calculated. The indices corresponding to the frame numbers are stored and both the video and the audio data corresponding to them is obtained and put in to container format using mkvmerge [32] and decoded from that point and played back using the VLC media player [40]. The results tabulated in Chapter 6 shows that visual delay is not perceptible and is almost consistent once synchronized. Thus, the de-multiplexing process is successfully achieved along with lip sync.

5.3 Summary

In this chapter, the process of de-multiplexing is explained in detail along with lip synchronization during playback. Chapter 6 provides details of the results obtained, conclusions and future work.

CHAPTER 6

RESULTS AND CONCLUSIONS

6.1 Implementation and results

The proposed algorithm for multiplexing and de-multiplexing is implemented in Matlab. There is no standard test sequence available and hence Audio Video Interleave (AVI) sequences were downloaded from YouTube and are used in this thesis. Dirac video codec has two software implementations, namely, Dirac research and Schrodinger implemented in C++ and ANSI - C respectively [28]. In this thesis, the Dirac research version is used for video.

The AVI sequence is split into YUV video format and WAV audio format using ffmpeg [38] freeware. YUV video format is encoded using Dirac (1.0.2 version) encoder and WAV audio format is encoded using Free Advanced Audio Coder (FAAC) [39]. Long GOP profile for Dirac video and low complexity profile for AAC audio is used. The video and audio bit streams obtained from the encoders are analyzed and the data is multiplexed based on the presentation time of video and audio respectively. Proposed algorithm incorporates both intra coding (I only) and inter (P and B pictures) coding of video. Sequences of length 10 seconds, 36 seconds and 50 seconds are used. A single program stream comprising of an audio and video elementary stream is implemented in this thesis. The Tables 6.1 and 6.2 provide the information of video frame rate, audio sampling frequency, compression ratio, bit rates and file size of the test sequences used. Clips1 and 2 of Table 6.1 are encoded for inter coding and at quality factor i.e. Q.F- 8 and Q.F- 7 respectively. Clips1 and 2 of Table 6.2 are encoded for intra coding and at Q.F- 8. ATSC - M/H [17][18] has an allocated bandwidth requirement of 19.6 Mbps [13], whereas the transport stream bitrates for the sequences obtained are 102.13 kbps and 96.72 kbps which can be easily accommodated. The TS bitrates for intra coded sequence are 482.65 kbps and 576.57 kbps for clip1 and clip2 respectively. Since, I only coding requires more number of bits when compared to inter coding, inter coding of pictures is used in most practical applications.

Table 6.1 Inter coding: test clips characteristics

Details of test clips	Clip 1 [49] (Shrek)	Clip 2 [49] (Despicable me)
Clip Duration (seconds)	50	36
Resolution	480 × 204	480 × 270
Chroma sub-sampling format	YUV 420P	YUV 420P
No: of video frames	1206	885
No: of audio frames	2166	1589
Video frame rate (fps)	24	24000/1001
Audio sampling frequency(kHz)	44.1	44.1
YUV file size (kB)	172985.63	168011.72
WAV file size (kB)	8658.04	6349.54
.DRC file size(kB)	4041.28	2629.67
.AAC file size (kB)	676.32	563.95
Video compression ratio	43:1	64:1
Audio compression ratio	13:1	12:1
DRC video bitrate (kbps)	80.83	73.05
AAC audio bitrate (kbps)	13.53	15.67
Video TS packets	23042	15039
Audio TS packets	4772	3927
TS packets	27814	18966
TS file size(kB)	5106.48	3482.039
TS bitrate(kbps)	102.13	96.72
Original AVI file size (kB)	4227	3199

Table 6.2 Intra coding: test clips characteristics

Details of test clips	Clip 1 [49] (Despicable me)	Clip 2 [49] (Despicable me)
Clip Duration (seconds)	36	10
Resolution	480 × 270	480 × 360
Chroma sub-sampling format	YUV 420P	YUV 420P
No: of video frames	885	224
No: of audio frames	1589	385
Video frame rate (fps)	24000/1001	25
Audio sampling frequency(kHz)	44.1	44.1
YUV file size (kB)	168011.72	56700
WAV file size (kB)	6349.54	1534.54
.DRC file size(kB)	16290.97	5496.87
.AAC file size (kB)	563.95	126.15
Video compression ratio	10:1	10:1
Audio compression ratio	12:1	12:1
DRC video bitrate (kbps)	452.53	549.69
AAC audio bitrate (kbps)	15.67	12.62
Video TS packets	90714	30563
Audio TS packets	3927	842
TS packets	94641	31405
TS file size (kB)	17375.5	5765.76
TS bitrate (kbps)	482.65	576.57
Original AVI file size (kB)	3199	1083

The buffer fullness of the data at the de-multiplexer needs to be continuously monitored to make sure that there is no overflow/underflow of data in video and audio buffers. If this is not handled effectively, there is a possibility of mute errors, freeze and out of sync problems during playback [36][37]. The Tables 6.3, 6.4 and 6.5 demonstrate that the data has been handled effectively ensuring buffer fullness. This also ensures that the data was being multiplexed effectively at the transmitter's end.

Table 6.3 Clip1: Check for buffer fullness and video/audio content playback time (Inter coding)

# of Video Frames in buffer	# of Audio frames in buffer	Size of Video buffer (kB)	Size of Audio buffer (kB)	Video content playback time* (in sec)	Audio content playback time (in sec)
100	181	323.4893	54.3311	4.2083	4.2028
200	365	577.1025	115.0576	8.4583	8.4753
400	731	1323.3018	235.0371	16.9583	16.9738
600	1097	1851.4629	352.2656	25.4583	25.4723
800	1461	2440.4971	466.5039	33.9167	33.9244
1000	1829	3456.3975	575.2070	42.4583	42.4693
1100	2012	5106.4766	629.8096	46.7083	46.7185

Table 6.4 Clip2: Check for buffer fullness and video/audio content playback time (Inter coding)

# of Video Frames in buffer	# of Audio frames in buffer	Size of Video buffer (kB)	Size of Audio buffer (kB)	Video content playback time* (in sec)	Audio content playback time (in sec)
100	185	277.9268	64.7295	4.2960	4.2957
200	367	631.0459	130.1924	8.5085	8.5217
400	732	1055.0078	257.5605	17.0170	16.9970
500	916	1425.7266	328.4414	21.2713	21.2695
600	1098	1670.6875	392.0518	25.4838	25.4955
700	1281	2068.3242	456.8135	29.7381	29.7448
800	1464	2382.5303	524.0430	33.9923	33.9940

From Tables 6.3 and 6.4, it is observed that the video and audio content playback time varies with a maximum of 17ms meaning that the video content and its corresponding audio content is entirely present. Since, SPS/PPS was also multiplexed as a TS packet at transmitter's end with frame number as zero, the time of this is accrued in the video content playback time. The SPS/PPS will be present prior to occurrence of each or few I frames in order to assist at the time of decoding. The corresponding time of number of SPS/PPS present, by the time the number of video frames was in the buffer is reflected in the video content playback time along with content playback time of the number of video frames. For instance, in Table 6.3 by the time 100 video frames were in, 1 SPS/PPS was present and hence the video content playback time was calculated as 101 × (1/fps) although 100 video frames only were present. Similarly, for each case, the video content playback time is calculated as:

Video content playback time = (# of video frames + # of SPS/PPS) **×** (1 /fps) (6.1)*

Table 6.5 Clip2: Check for buffer fullness and video/audio content playback time (Intra coding)

No: of Video Frames in buffer	No : of Audio frames in buffer	Size of Video buffer (kB)	Size of Audio buffer (kB)	Video content playback time* (in sec)	Audio content playback time (in sec)
30	56	711.6436	16.6016	1.2800	1.3003
60	112	1419.9668	35.0361	2.6000	2.6006
100	188	2369.0645	60.0449	4.3600	4.3654
150	283	3545.3672	92.5586	6.5600	6.5712
200	378	4873.7568	123.8281	8.7600	8.7771

* - Please refer to equation 6.1.

From Table 6.5, it is observed that the video and audio content playback time vary with a maximum delay of 20ms. A delay time of one frame is allowed between video and audio content playback, while maintaining sync during decoding and playback.

The multiplexed output is de-multiplexed into video and audio buffers using the de-multiplexing algorithm implemented. Tables 6.6 and 6.7 tabulate de-multiplexed results for inter coding and Table 6.8 tabulates de-multiplexed results for intra coding.

Table 6.6 Clip 1: Output of de-multiplexer (Inter coding)

TS packet number	Video IDR frame number chosen	Synchronized audio frame number	Chosen video frame presentation time (sec)	Chosen audio frame presentation time (sec)	Delay (ms)	Visual delay Perceptible?
1	1	2	0.0417	0.0464	4.7	No
100	59	106	2.4583	2.4613	3.0	No
2000	118	212	4.9167	4.9226	6.0	No
5000	287	515	11.9583	11.9582	0.1	No
10000	467	838	19.4583	19.4582	0.1	No

Table 6.7 Clip 2: Output of de-multiplexer (Inter coding)

TS packet number	Video IDR frame number chosen	Synchronized audio frame number	Chosen video frame presentation time (sec)	Chosen audio frame presentation time (sec)	Delay (ms)	Visual delay Perceptible?
1	1	2	0.0417	0.0464	4.7	No
100	65	117	2.7110	2.7167	5.7	No
2000	146	262	6.0894	6.0836	5.8	No
10000	491	882	20.4788	20.4800	1.2	No
12000	611	1097	25.4838	25.4723	11.5	No

Table 6.8 Clip 2: Output of de-multiplexer (Intra coding)

TS packet number	Video IDR frame number chosen	Synchronized audio frame number	Chosen video frame presentation time (sec)	Chosen audio frame presentation time (sec)	Delay (ms)	Visual delay Perceptible?
1	1	2	0.0400	0.0464	6.4	No
5000	38	65	1.5200	1.5093	10.7	No
10000	75	129	3.0000	2.9954	4.6	No
20000	149	257	5.9600	5.9675	7.5	No
25000	182	314	7.2800	7.2911	11.1	No

The data can be de-multiplexed from any TS packet number irrespective of any sequence. From the Tables 6.6, 6.7 and 6.8, it is observed that the maximum delay between audio and video presentation time is about 12ms, while the delay is almost consistent otherwise. MPEG-2 threshold to ensure the audio-video sync is about 40ms [25]. The video and audio buffer data obtained then needs to be put into a container format and the data can then be played back using a media player. Mkvmerge [32], a freeware is used to put the data to matroska multimedia container format (.mkv). VLC media player [40] decodes and then the data is played back. Since, mkvmerge tool ignores the fact that Dirac has bi-directional predicted (B) frames some jitters will be observed in the reconstructed video. The tool works fine for intra only and I-P coding of Dirac and is justified in this thesis. Also, due to non-availability of other freeware that supports to check for inter sequence muxing, mkvmerge has only been used in this thesis.

6.2 Conclusions

This thesis has focused on implementing an effective scheme to transmit and receive the data ensuring audio-video synchronization during playback. This was achieved using MPEG-2 TS [19]. While data is to be broadcasted in a wireless medium (an error prone network), the need for error detection and correction becomes necessary. Packet identifiers are incorporated in the TS header to detect packet loss if any. Error correction codes can be accommodated in the implemented algorithm to ensure no loss of data at the receiver's end. Time stamps in the form of frame numbers aid in achieving synchronization. The use of Dirac video bit-stream and AAC audio bit-stream helps to deliver high quality video and audio at a reasonably low bit rate. Dirac was used to broadcast the Beijing Olympics in 2008 [14]. Since, the performance of Dirac is comparable with H.264 and it is an open source, Dirac can be suggested as a choice of consideration in broadcasting. Dirac does not have any NAL bit-stream format. It is shown that the nature of Dirac bit stream in non- NAL syntax is used without any conversion to NAL to perform multiplexing. From the results tabulated, it can be clearly concluded that synchronization between audio and video is effectively achieved with visual delay not perceptible. The buffer fullness was handled with a maximum delay of about 20ms between audio and video. Also, during decoding the audio-video synchronization was achieved with a maximum delay of about 12ms. Thus, the implemented algorithm effectively multiplexes, de-multiplexes and achieves synchronization during the playback.

6.3 Future Work

This thesis has focused on implementing one program elementary stream i.e. one video and one audio only. With minor additions to this implementation, multiple program elementary streams can be accommodated to broadcast data simultaneously in multiple channels. Also, subtitles can be incorporated while multiplexing along with audio and video elementary streams when required.

Mkvmerge [32] supports Intra (I) and Predictive (P) coding of pictures and is ignoring the fact that Dirac video has bi-directional predicted (B) frames. Hence, mkvmerge can be modified to support B picture coding for Dirac.

In practice, the data is broadcasted in an error prone network. The implementation used in this thesis does not include error correction codes and hence rugged error correction codes can be incorporated to this implementation.

APPENDIX A

LUMA COMPONENT OF THE TEST SEQUENCES USED

Figure A.1 Frame 140: luma component of Clip2: 36 sec [50]

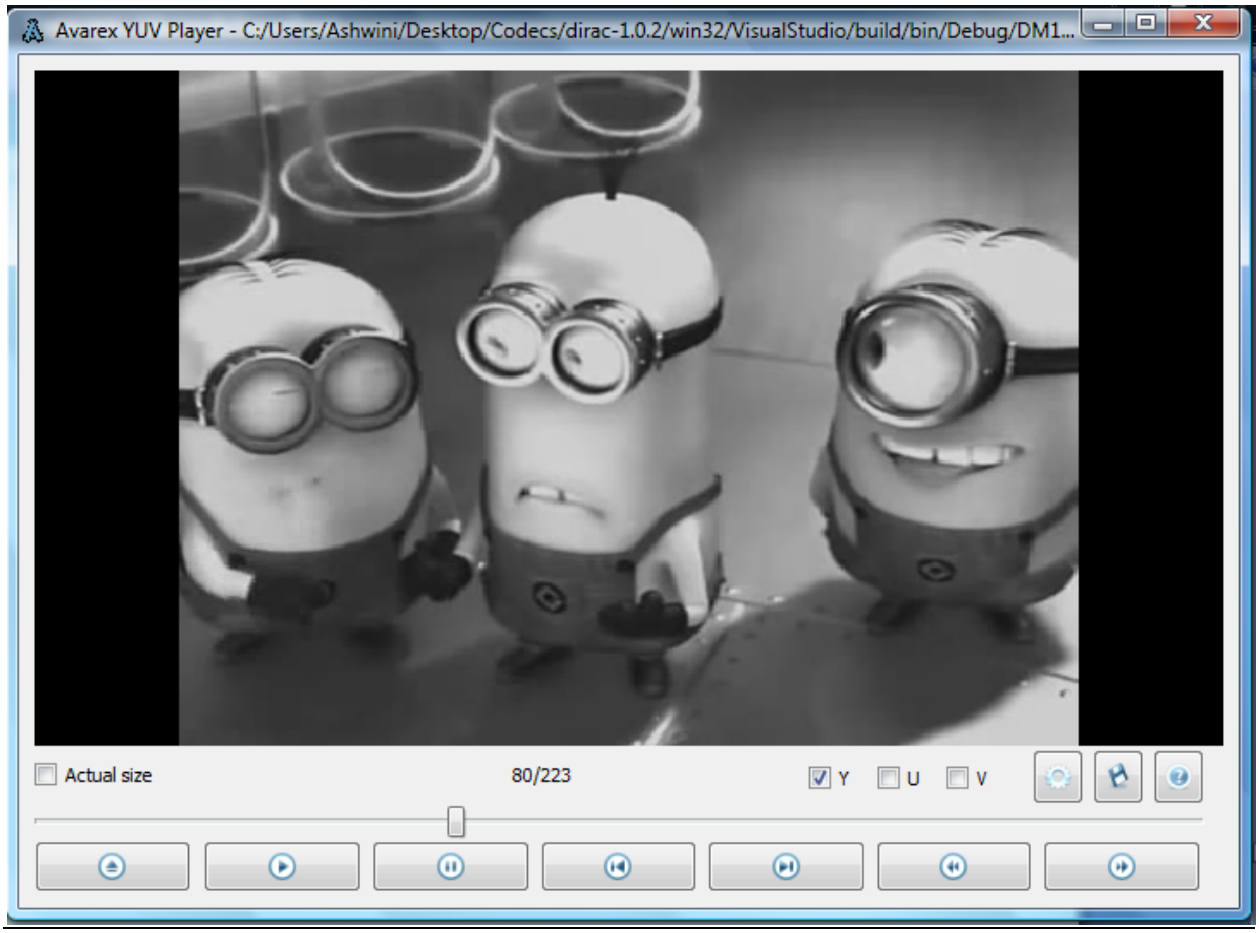

Figure A.2 Frame 80: luma component of Clip2: 10 sec [50]

Figure A.3 Frame 365: luma component of Clip1: 50 sec [50]

REFERENCES

[1] T. Borer and T. Davies, “Dirac video compression using open technology”, BBC EBU Technical Review, July 2005.

[2] K.Oriantar, K.K. Loo and Z.Xue, “Performance comparison of emerging Dirac video codec with H.264/AVC”, School of engineering and design, Brunel university, UK.

[3] A. Ravi, "Performance analysis and comparison of Dirac video codec with H.264/ MPEG-4 Part 10 AVC", M.S.E.E Thesis, University of Texas at Arlington, Arlington, TX, Aug. 2009.

[4] MPEG. Information technology - Generic coding of moving pictures and associated audio information, part 4: Conformance testing. International Standard IS 13818–4, ISO/IEC JTC1/SC29 WG11, 1998.

[5] M. Bosi and M. Goldberg, “Introduction to digital audio coding and standards”, Boston: Kluwer, 2003.

[6] H. Eeckhaut et al, “Speeding up Dirac’s entropy coder”, proceedings of the 5th WSEAS conference on multimedia , internet and video technologies, Corfu, Greece, pp: 120 -125, August 2005.

[7] “Encapsulation of Dirac in ISO/IEC 13818-1”, British Broadcasting Corporation.

[8] “Dirac ISO/IEC 13818-1 Transport stream mapping”, encapsulation of Dirac video content and time code markers in ISO/IEC 13818-1 transport streams, draft, Feb 2007.

[9] P.A. Sarginson, “MPEG-2: Overview of systems layer”, BBC RD 1996/2.

[10] J.M. Boyce, “The United States television broadcasting transition”, IEEE signal processing magazine, vol.26, pp. 110-112, May 2009.

[11] K. Brandenburg, “MP3 and AAC Explained”, AES 17th International conference, Florence, Italy, Sept. 1999.

[12] “Introduction to MPEG”, excerpted from IPTV basics, Lawrence Harte.

[13] Website: http://www.atsc.org/cms/

[14] Dirac information: http://en.wikipedia.org/wiki/Dirac_%28video_compression_format%29

[15] Audio sampling frequency table link:

http://wiki.multimedia.cx/index.php?title=MPEG-4_Audio#Sampling_Frequencies

[16] DVB-H link: http://dvb-h.org/

[17] ATSC-M/H link: http://atsc.org/cms/

[18] Mobile video DTV: http://www.openmobilevideo.com/about-mobile-dtv/standards/

[19] Information Technology – Generic coding of moving pictures and associated audio: Systems, International Standard 13818-1, ISO/IEC JTC1/SC29/WG11 N0801, 1994.

[20] B. J. Lechner, et al, "The ATSC transport layer, including program and system information (PSIP)," Proceedings of the IEEE , vol. 94, no. 1, pp.77–101, Jan. 2006.

[21] "Special issue on global digital television: technology and emerging services", Proceedings of the IEEE, vol. 94, pp. 5-332, Jan. 2006.

[22] H. Murugan, "Multiplexing H264 video bit-stream with AAC audio bit-stream, demultiplexing and achieving lip sync during playback", M.S.E.E Thesis, University of Texas at Arlington, Arlington, TX, May 2007.

[23] Dirac specification: Version 2.3, Issued: Sept.2008.

[24] MPEG transport stream:
http://www.iptvdictionary.com/iptv_dictionary_MPEG_Transport_Stream_TS_definition.html

[25] MPEG. Information technology - generic coding of moving pictures and associated audio information, part 4: Conformance testing .International Standard IS 13818–4, ISO/IEC JTC1/SC29 WG11, 1998.

[26] Dirac document: "Dirac compression family using open technology".
http://www.bbc.co.uk/rd/projects/dirac/index.shtml

[27] Digital Video Systems Characteristics Standard for Cable Television, ANSI/SCTE 432004.

[28] Dirac website: www.diracvideo.org

[29] Digital audio compression standard (AC-3, E-AC-3), revision B, ATSC Document A/52B, Advanced Television Systems Committee, Washington, D.C., Jun. 14, 2005. www.atsc.org/cms

[30] /CMPT 365 Course Slides/, School of Computing Science, Simon Fraser fig3:
http://www.cs.sfu.ca/CourseCentral/365/li/material/notes/Chap4/Chap4.3/chap 4.3.html

[31] Dirac software: http://sourceforge.net/projects/dirac/

[32] Mkvmerge software: http://www.bunkus.org/videotools/mkvtoolnix/downloads.html

[33] I.H.Witten, R.M. Neal and J.G.Cleary: “Arithmetic coding for data compression”, Communications of the ACM, Vol.30, No.6, pp. 520-540, June 1987.

[34] A. Puri, X. Chen and A. Luthra, “Video coding using the H.264/MPEG-4 AVC compression standard”, Signal processing: image communication, vol. 19, issue 9, pp. 793-849, Oct. 2004.

[35] G. A. Davidson, “Digital audio coding: Dolby AC-3,” in the Digital Signal Processing Handbook, V. K. Madisetti and D. B. Williams, Eds. Boca Raton, FL: CRC, pp. 41-1–41-21, 1998.

[36] MPEG-4: ISO/IEC JTC1/SC29 14496-3: Information technology – coding of audio-visual objects-part 3 : Audio, amendment 4:audio lossless coding(ALS), new audio profiles and ASAC extensions.

[37] MPEG-2: ISO/IEC JTC1/SC29 13818-7: Advanced audio coding, AAC. IS WG11,1997.

[38] ffmpeg software: http://www.ffmpeg.org/

[39] FAAC software: www.audiocoding.com

[40] VLC media player: www.videolan.org

[41] A.Ravi and K.R.Rao, “Performance analysis and comparison of the Dirac video codec with H.264/ MPEG-4, part 10”, for the book “Advances in reasoning-based image processing, analysis and intelligent systems: conventional and intelligent paradigms”, 2011.

[42] H.Murugan and K.R.Rao, “Multiplexing H.264 video with AAC audio bit streams, de-multiplexing and achieving lip sync”, ICEAST 2007, Bangkok, Thailand, 21 – 23 Nov.2007.

[43] H. Kalva, et al, “Implementing multiplexing, streaming and server interaction for MPEG-4”, IEEE Trans. on circuits and systems for video technology, vol. 9, No. 8, pp. 1299-1311, Dec. 1999.

[44] C. C. Todd, et al, “AC-3: perceptual coding for audio transmission and storage,” presented at the 96th Conv. Audio Engineering Society, 1994, preprint 3796.

[45] M. Uehara, “Application of MPEG-2 systems to terrestrial ISDB (ISDB-T)”, Proceedings of the IEEE, vol.94, pp. 261-268, Jan. 2006.

[46] G.A. Davidson, et al, “ATSC video and audio coding”, Proceedings of the IEEE, vol. 94, pp. 60 - 76, Jan. 2006.

[47] J.Jain and P. Desale, “Low power transport de-multiplexer for ATSC and DVB broadcast format”, LSI research and development Pune pvt. Ltd.

[48] A.Ravi and K.R.Rao, “Performance analysis and comparison of the Dirac video codec with H.264/ MPEG-4, part 10”, IJWMIP, accepted : July 2011.

[49] AVI sequences: www.youtube.com

[50] AVAREX YUV PLAYER: http://www.avarexgroup.com/en/solutions.php?sub=2&product=rawplayer

[51] Information technology- advanced coding of audio and video part -7: mobile video

[52] Geraold Blakowski et.al “A media synchronization survey: Reference model, specification and case studies”, IEEE journal on selected areas in communications, vol.14, no.1, January 1996.

BIOGRAPHICAL INFORMATION

Ashwini Urs graduated with the Bachelor of Engineering degree in Electrical and Electronics Engineering from Visveswaraiah technological University, Karnataka, India in May 2005. She joined University of Texas - Arlington in Spring 2009 and pursued Master of Science in Electrical Engineering. She will be graduating in Spring 2011 and the degree will be conferred in May 2011. Her area of specialization is image processing and video coding. She was a member of the multimedia processing research group under supervision of Dr. K. R. Rao from January 2009. Her areas of interest include research and development in image processing and video coding, digital signal processing and wireless communications.