

**MUX/DEMUX OF HEVC/H.265 VIDEO STREAM WITH HE-
AAC V2 AUDIO BIT STREAM AND TO ACHIEVE LIP SYNCH**

by

DEEPIKA SREENIVASULU PAGALA

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

August 2016

Copyright © by Deepika Sreenivasulu Pagala 2016

All Rights Reserved



ACKNOWLEDGEMENTS

Firstly, I would like to express my heartfelt gratitude to my Professor, Dr. K. R. Rao. This work would not have been possible without his continuous support and encouragement. He has been a constant source of inspiration right from inception to the conclusion of this thesis and throughout my master's journey. It is an honor for me to work under Dr. K. R. Rao and be a part of his lab.

I thank Dr. Bredow and Dr. Alavi for taking interest in my work and accepting to be a part of my thesis defense committee.

I thank Mrs. Ashwini Urs, for providing her valuable support and helping me in my thesis work.

I thank Mr. Naveen Siddaraju and Mr. Swaminathan Sridhar for their inputs that helped me during my thesis.

I thank my company Realtime Data LLC and the team for giving me an opportunity to carry out the research work and also for their encouragement to complete my thesis.

I thank my Under Graduate Professors for their continuous support and motivation.

I am grateful and indebted to my mother Mrs. Nagarathna P. K. and my father Mr. Sreenivasulu Pagala for their encouragement, support and standing by me all throughout my life.

Finally, I am thankful to my friend, Mr. Sudheer Alluru, for his continuous support, motivation and being with me through all the times of struggle and celebration.

July 1, 2016

ABSTRACT

MUX/DEMUX OF HEVC/H.265 VIDEO STREAM WITH MPEG-4 HE-AAC V2 AUDIO BIT STREAM AND TO ACHIEVE LIP SYNCH

Deepika Sreenivasulu Pagala, MS

The University of Texas at Arlington, 2016

Supervising Professor: Dr. K. R. Rao

The increasing demand for video and the increase in the number of devices capable of supporting digital media facilitate the enormous demand for video streaming over internet and Internet Protocol Television (IPTV) applications. The high bit rates that result from the various types of digital video make their transmission through their intended channels very difficult. Hence it becomes necessary to select a good compression scheme to overcome storage and transportation problems of the digital video.

High Efficiency Video Coding (HEVC) [1] is an international standard for video compression developed by a working group of ISO/IEC MPEG (Moving Picture Experts Group) and ITU-T VCEG (Video Coding Experts Group). The main goal of HEVC standard is to significantly improve compression performance compared to existing standards (such as H.264/Advanced Video Coding [4]) in the range of 50% bit rate reduction at similar visual quality [1]. HEVC is designed to address existing applications of H.264/MPEG-4 AVC and to focus on two key issues: increased video resolution and increased use of parallel processing architectures [1].

In the case of audio content, the MPEG-4 High Efficiency AAC v2 profile (HE-AAC v2) has proven, in several independent tests, to be the most efficient audio compression scheme available worldwide. High efficiency advanced audio codec version 2 also known as enhanced aacplus is a low bit rate audio codec defined in MPEG4 audio profile [2] belonging to the AAC family. It is specifically designed for low bit rate applications such as streaming.

The audio and video codec standards have been chosen based on ATSC-M/H (advanced television systems committee – mobile handheld) [19].

The objective of this thesis is to implement a multiplexing scheme for the elementary schemes of HEVC main profile and HE-AAC V2 using MPEG2 systems specifications [24] and demultiplex the transport stream at the receiving end with audio – video synchronization. Since audio and video codecs have frame wise arrangement, frame numbers are used as information to achieve audio-video synchronization. Two layers of packetization of audio and video streams is involved in the multiplexing process where the first layer results in Program Elementary Stream (PES) packets which are of variable size. The synchronization information is embedded in the headers of the first layer of packetization. Since PES packets are not suitable for transport due to their variable size, these are further packetized as Transport Stream (TS) packets of fixed length and 188 bytes long. These packets are decoded by the receiver and the original elementary streams

are reconstructed. Playback time is used as criteria for allocating data packets at multiplexer to prevent overflow and underflow of buffers during demultiplexing. The net transport stream bitrates for the sequences obtained are 267.2 kbps, 1095.08 kbps and 1093.44 kbps which can be easily accommodated in systems such as ATSC-M/H, which has an allocated bandwidth of 19.6 Mbps [19]. Encoding video using HEVC and audio based on HE-AAC V2, multiplexing the two coded bit-streams, packetization, de-multiplexing the two coded bit-streams, decoding the video (HEVC) and audio (HE-AAC V2) 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:

- Acknowledgements..... iii
- Abstract.....iv
- List of Figures.....ix
- List of Tables.....x
- List of Acronyms and Abbreviations.....xi
- 1. Introduction12
 - 1.1 Introduction12
 - 1.1.1 Evolution of Video Coding standards12
 - 1.1.2 Need for Video compression.....13
 - 1.1.3 Fundamental concepts in video coding13
 - 1.1.4 Picture types 13
 - 1.2 Introduction about Thesis.....15
 - 1.3 Thesis Outline.....17
- 2. Overview of High Efficiency Video Coding18
 - 2.1 Introduction18
 - 2.2 Encoder and Decoder in HEVC.....18
 - 2.3 Features of HEVC.....20
 - 2.3.1 Partitioning.....20
 - 2.3.1.1 Division of the Picture into Coding Tree Units.....20
 - 2.3.1.2 Division of the CTB into CBs.....20
 - 2.3.1.3 PBs and PUs.....21
 - 2.3.1.4 Tree-Structured Partitioning Into Transform Blocks and Units. 22
 - 2.3.2 Prediction23
 - 2.3.2.1 Intra picture Prediction23
 - 2.3.2.2 Inter picture Prediction 24
 - 2.3.2.3 Fractional Sample Interpolation.....25
 - 2.3.3 Transform and Quantization26
 - 2.3.4 In loop Filtering27
 - 2.3.4.1 Deblocking Filter.....27
 - 2.3.4.2 Sample Adaptive Offset27

2.3.5 Entropy coding	28
2.4 Profiles in HEVC.....	28
2.4.1 Main Profile	28
2.4.2 Main 10 Profile	28
2.4.3 Main Still Picture Profile.....	28
2.5 Parallel decoding syntax and modified slice structuring.....	29
2.5.1 Tiles.....	29
2.5.2 Wavefront parallel processing.....	29
2.5.3 Dependent slice segments.....	29
2.5.4 Slices.....	29
2.6 Bit stream syntax of H.265.....	29
2.7 Summary.....	31
3. Overview of HE-AAC V2.....	32
3.1 Introduction.....	32
3.2 Architecture of HE-AAC v2	34
3.3 MPEG AAC.....	34
3.4 Spectral Band Replication (SBR).....	35
3.5 Parametric Stereo	36
3.6 Functionality of HE-AAC V2.....	37
3.7 Audio quality evaluation.....	39
3.8 Advanced Audio Coding (AAC).....	40
3.9 HE-AAC v2 bitstream formats.....	43
3.10 Summary.....	45
4. Multiplexing.....	46
4.1 Introduction.....	46
4.2 MPEG-2 System Layers.....	47
4.3 Packetization.....	49
4.4 Packetized Elementary Stream (PES).....	51
4.5 Transport stream format.....	56
4.5.1 TS packet header.....	57
4.6 Frame number as time stamp.....	59
4.7 Advantages of frame numbers over the existing method for time stamps.....	60

4.8 Proposed multiplexing method.....	60
4.9 summary.....	61
5. Demultiplexing.....	62
5.1 Introduction.....	62
5.2 Synchronization and Playback.....	64
5.3 Summary.....	64
6. Results and Conclusions.....	65
6.1 Implementation and results.....	65
6.2 Conclusions.....	69
6.3 Future Research.....	69
APPENDIX A	70
Test sequences.....	70
APPENDIX B.....	71
Test Platform.....	71
APPENDIX C.....	72
Using ffmpeg to split avi into audio and video files.....	72
References	73
Biographical Information	79

List of Figures

- Fig 1.1 Evolution of video coding standards
- Fig 1.2 4:2:0 sub-sampling pattern
- Fig 1.3 4:2:2 sub-sampling pattern and 4:4:4 sampling
- Fig 1.4 Group of Pictures
- Fig 2.1 Block Diagram of HEVC Encoder
- Fig 2.2 Block diagram of HEVC Decoder
- Fig 2.3 Picture, Slice, Coding Tree Unit (CTU), Coding Unit
- Fig 2.4 Modes for splitting a CB into PBs, subject to certain size constraints. For intra picture-predicted CBs, only $M \times M$ and $M/2 \times M/2$ are supported
- Fig 2.5 Subdivision of a CTB into CBs [and transform block (TBs)]. Solid lines indicate CB boundaries and dotted lines indicate TB boundaries. (a) CTB with its partitioning. (b) Corresponding quadtree
- Fig 2.6 Modes and angular intra prediction directions in HEVC
- Fig 2.7 Integer and fractional sample positions for luma interpolation
- Fig 2.8 CTU showing range of transform (TU) sizes
- Fig 2.9 Hierarchy of the Main Profiles in HEVC
- Fig 2.10 Comparison of HEVC and H.264 NAL units
- Fig 3.1 HE AAC Audio Codec Family
- Fig 3.2 Original audio signal
- Fig 3.3 High band reconstruction through SBR
- Fig 3.4 Basic principle of the parametric stereo coding process
- Fig 3.5 Block diagram of a complete HE-AAC v2 encoder
- Fig 3.6 Block diagram of a complete HE-AAC v2 decoder
- Fig 3.7 Anticipated audio quality vs. bitrate for the various codecs of the HE-AAC v2 family
- Fig 3.8 AAC Encoder Block Diagram
- Fig 3.9 ADTS elementary stream
- Fig 4.1 Example ATSC transmission/reception block diagram showing the MPEG-2 TS multiplexer
- Fig 4.2 MPEG-2 System Layers
- Fig. 4.3 Two layers of packetization method adopted in MPEG-2 systems
- Fig 4.4 Packet structure hierarchy of a TS
- Fig 4.5 Encapsulation of PES from elementary streams
- Fig. 4.6 Structure of PES packet
- Fig 4.7 A PES packet header
- Fig 4.8 PES packet structure
- Fig 4.9 Encapsulation of TS packets from PES packets
- Fig 4.10 MPEG transport stream and standard structure of MPEG-TS packet
- Fig 5.1 Flowchart of demultiplexer

List of Tables

Table 2.1	The NAL unit types and their associated meanings, classes in the HEVC standard
Table 3.1	ADTS header format
Table 3.2	Profile bits expansion
Table 4.1	Video Elementary Stream Format Glossary
Table 4.2	PES Header Glossary
Table 4.3	TS packet header description as adopted in MPEG 2 systems
Table 4.1	PES header description
Table 4.2	TS packet header description as adopted in MPEG 2 systems
Table 6.1	Inter coding: test clips characteristics
Table 6.2	Clip 1 : Output of de-multiplexer (Oscars)
Table 6.3	Clip 2 : Output of de-multiplexer (starwars2k)
Table 6.4	Clip 3 : Output of de-multiplexer (starwars4k)
Table 6.5	Clip 1: Check for buffer fullness and video/audio content playback time (Oscars)
Table 6.6	Clip 2: Check for buffer fullness and video/audio content playback time (starwars2k)
Table 6.7	Clip 3: Check for buffer fullness and video/audio content playback time (starwars4k)

List of Acronyms and Abbreviations:

AAC	-	Advanced audio coding.
ADIF	-	Audio data interchange format.
ADTS	-	Audio data transport stream.
AES	-	Audio engineering society.
AFC	-	Adaptation field control.
ATSC	-	Advanced Television Systems Committee
AVC	-	Advanced video coding.
AVI	-	Audio Video Interleave
CABAC	-	Context Adaptive Binary Arithmetic Coding
CB	-	Coding Block
CPU	-	Central Processing Unit
CTB	-	Coding Tree Block
CTU	-	Coding Tree Unit
CU	-	Coding Unit
DCT	-	Discrete Cosine Transform
DEMUX	-	Demultiplexing
DMB	-	Digital multimedia broadcasting
DTS	-	Decoding Time Stamp
ES	-	Elementary Stream
Fps	-	frames per second
GPP	-	General Purpose Processing
HDTV	-	High definition television.
HE	-	High Efficiency
HE-AAC V2	-	High Efficiency Advanced Audio Codec Version 2
HEVC	-	High Efficiency Video Coding
IC	-	Inter Channel Coherence
IDR	-	Instantaneous decoder refresh
IID	-	Information Industry Department
IP	-	Internet Protocol
IPD	-	Inter Channel Phase Differences
ISDB	-	International Services Digital Broadcasting System
ISO	-	International Organization for Standardization
ITU-T	-	International Telecommunication Union – Telecommunication Standardization Sector
JCT-VC	-	Joint Collaborative Team on Video Coding
KTA	-	Key Technical Areas
LC	-	Low Complexity
MC	-	Motion Compensation
MDCT	-	Modified Discrete Cosine Transform
M/H	-	Mobile / Handheld
MP4	-	Moving picture experts group -4
MPEG	-	Moving picture experts group
MPTS	-	Multi Program Transport Stream
MTU	-	Maximum Transmission Unit
MUX	-	Multiplexing

NAL	-	Network Adaptation Layer
OPD	-	Overall Phase Difference
PB	-	Prediction Block
PCM	-	Pulse Code Modulation
PCR	-	Program Clock Reference
PES	-	Packetized elementary stream
PID	-	Packet identifiers
PS	-	Parametric Stereo
PS	-	Program Stream
PSIP	-	Program and System Information Protocol (PSIP)
PTS	-	Presentation Time Stamp
PU	-	Prediction Unit
PUS	-	Payload unit start
QMF	-	Quadrature Mirror Filter banks
RTP/IP	-	Real Time Transport protocol/Internet protocol
SBR	-	Spectral Band Replication
SPS	-	Sequence Parameter Set
SSR	-	Scalable sampling rate
SVC	-	Scalable Video Coding
TS	-	Transport stream
TU	-	Transform unit
UHD	-	Ultra High Definition
VCL	-	Video coding layer

CHAPTER 1

INTRODUCTION

1.1 Introduction

1.1.1 Evolution of Video Coding standards :

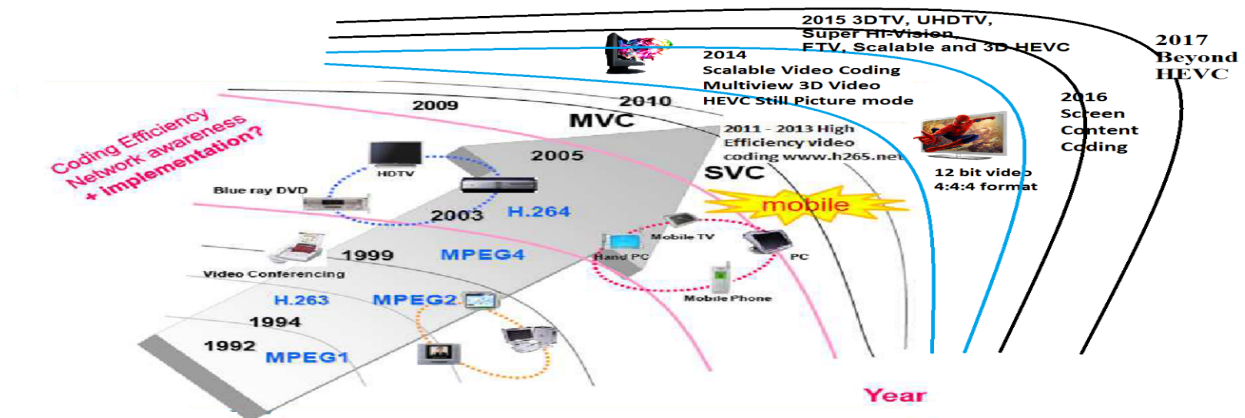


Fig 1.1 Evolution of video coding standards [75]

Major video coding standards have been developed by the International Standardization Organization / International Electro technical Commission (ISO/IEC) and the International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) [8]. Figure 1.1 shows a historical perspective for video coding standards development since the very first ITU-T H.120. The emergence of H.264/AVC doubled the coding efficiency from that of the MPEG-4 simple profile and has therefore gained wide industrial acceptance recently [8]. Further extensions of H.264/AVC include high profiles , scalable video coding (SVC) extension , and multi view video coding (MVC) extension [8] .

Back in 2005, the ITU-T Visual Coding Experts Group (VCEG) considered the future work beyond H.264/AVC [8]. Possible targets and scope of the standard were brainstormed and a software known as Key Technical Area (KTA) was developed and released in 2008 [8] . In 2009, the ISO/IEC Moving Picture Experts Group (MPEG) began a similar call for High-Performance Video Coding (HVC) [8].

1.1.2 Need for Video Compression- Growing demand for video :

- Increase in applications, content, fidelity, etc. -Need higher coding efficiency! [70].
- Ultra-HD 4K broadcast started in Japan in 2014. London Olympics Opening and Closing Ceremonies shot in Ultra-HD 8K. - Need higher throughput! [70].
- 25x increase in mobile data traffic over next five years. Video is a “must have” on portable devices. - Need lower power! [70].

1.1.3 Fundamental Concepts in Video Coding:

Most digital video applications rely on the display of color video and so need a mechanism to capture and represent color information. A monochrome image requires just one number to indicate the brightness or luminance of each spatial sample. Color images, on the other hand, require at least three numbers per pixel position to represent color accurately. The method chosen to represent brightness (luminance or luma) and color is described as a color space.

The common color spaces for digital image and video representation are:

- RGB color space – Each pixel is represented by three numbers indicating the relative proportions of red, green and blue colors
- YC_rC_b color space – Y is the luminance component, a monochrome version of color image. Y is a weighted average of R, G and B

$$Y = k_r R + k_g G + k_b B$$

Where k are the weighting factors.

- The color information is represented as color differences or chrominance components, where each chrominance component is difference between R, G or B and the luminance Y.
- As the human visual system is less sensitive to color than the luminance component, YC_rC_b has advantages over RGB space. The amount of data required to represent the chrominance component reduces without impairing the visual quality [10].

The popular pattern of sampling [10] is:

- 4:4:4 – The three components Y: C_r: C_b has the same resolution, which is for every 4 luminance samples there are 4 C_r and 4 C_b samples.

The popular patterns of sub-sampling [10] are:

- 4:2:2 – For every 4 luminance samples in the horizontal direction, there are 2 C_r and 2 C_b samples. This representation is used for high quality video color reproduction.
- 4:2:0 – The C_r and C_b each have half the horizontal and vertical resolution of Y. This is popularly used in applications such as video conferencing, digital television and DVD storage.

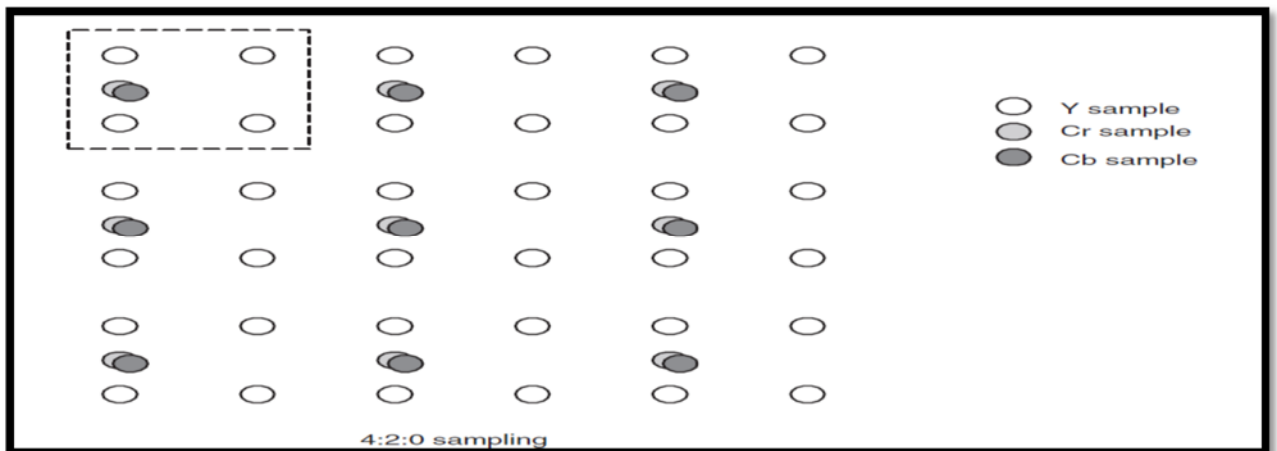


Fig 1.2 4:2:0 sub-sampling pattern [10].

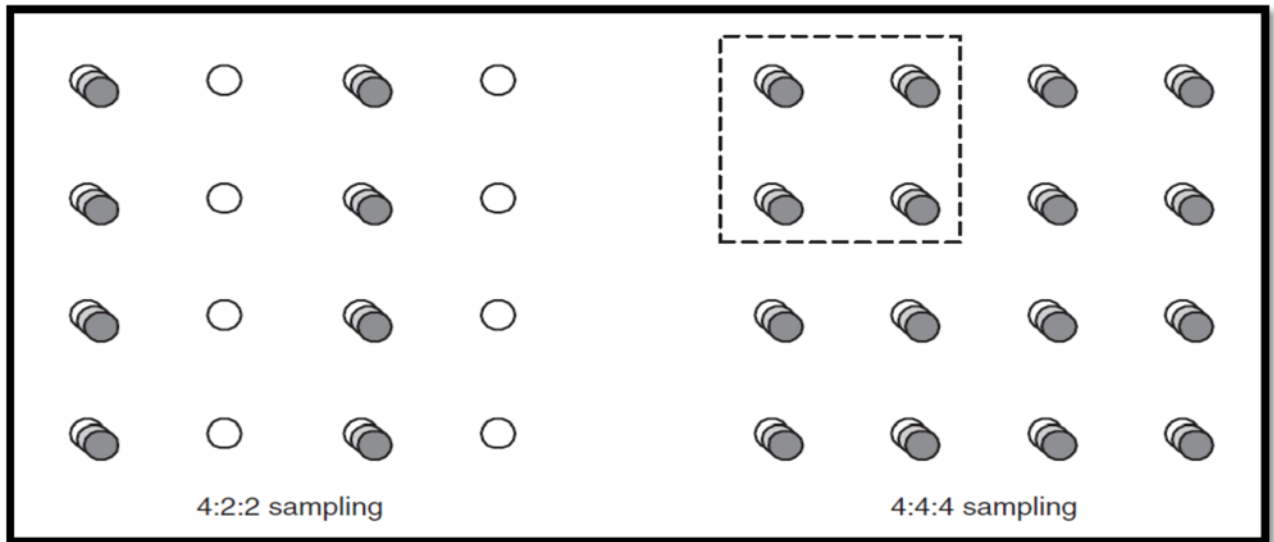


Fig 1.3 4:2:2 sub-sampling pattern and 4:4:4 sampling pattern [10].

1.1.4 Picture Types:

The MPEG standard specifically defines three types of pictures [73]:

1. Intra Pictures (I-Pictures)
2. Predicted Pictures (P-Pictures)
3. Bidirectional Pictures (B-Pictures)

These three types of pictures are combined to form a group of picture.

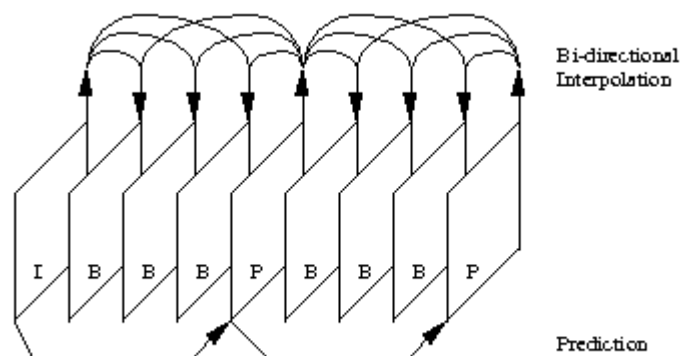


Fig 1.4 Group of Pictures [73]

Intra Pictures

Intra pictures, or I-Picture, are coded using only information present in the picture itself, and provides potential random access points into the compressed video data. It uses only transform coding and provide moderate compression. Typically it uses about two bits per coded pixel.

An IDR access unit contains an independently coded picture— i.e., a coded picture that can be decoded without decoding any previous pictures in the NAL unit stream. The presence of an IDR access unit indicates that no subsequent picture in the bitstream will require reference to pictures prior to the picture that it contains in order to be decoded. The IDR picture is used within a coding structure known as a closed GOP (in which GOP stands for group of pictures) [1].

Predicted Pictures

Predicted pictures, or P-pictures, are coded with respect to the nearest previous I- or P-pictures. This technique is called forward prediction and is illustrated in above figure. Like I-pictures, P-pictures also can serve as a prediction reference for B-pictures and future P-pictures. Moreover, P-pictures use motion compensation to provide more compression than is possible with I-pictures.

Bidirectional Pictures

Bidirectional pictures, or B-pictures, are pictures that use both a past and future picture as a reference. This technique is called bidirectional prediction. B-pictures provide the most compression since they use the past and future picture as a reference, however, the computation time is the largest.

1.2 Introduction about Thesis

With the increase in demand for video and the increase in the number of devices capable of supporting digital media, there is an enormous demand for video streaming over internet and Internet Protocol Television (IPTV) applications. The high bit rates that result from the various types of digital video make their transmission through their intended channels very difficult. Hence it becomes necessary to select a good compression scheme to overcome storage and transportation problems of the digital video.

High Efficiency Video Coding (HEVC) [1] is an international standard for video compression developed by a working group of ISO/IEC MPEG (Moving Picture Experts Group) and ITU-T VCEG (Video Coding Experts Group). The main goal of HEVC standard is to significantly improve compression performance compared to existing standards (such as H.264/Advanced Video Coding [4]) in the range of 50% bit rate reduction at similar visual quality [1].

HEVC is designed to address existing applications of H.264/MPEG-4 AVC and to focus on two key issues: increased video resolution and increased use of parallel processing architectures [1]. Version 1 of the HEVC standard defines three profiles: **Main**, **Main 10**, and **Main Still Picture**. Version 2 of HEVC adds 21 range extensions profiles, two scalable extensions profiles, and one multi-view profile. It primarily targets consumer applications as pixel formats are limited to 4:2:0 8-bit and 4:2:0 10-bit. The next revision of the standard, finalized in 2014, enables new use-cases with the support of additional pixel formats such as 4:2:2 and 4:4:4 and bit depth higher than 10-bit [9], embedded bit-stream scalability, Screen content coding, 3D video [5] and multiview video [6].

In the case of audio content, the MPEG-4 High Efficiency AAC v2 profile (HE-AAC v2) has proven, in several independent tests, to be the most efficient audio compression scheme available worldwide. It has recently been selected within DVB as part of its overall codec toolbox [23].

High efficiency advanced audio codec version 2 also known as enhanced aacplus is a low bit rate audio codec defined in MPEG4 audio profile [2] belonging to the AAC family. It is specifically designed for low bit rate applications such as streaming.

HE AAC v2 is a combination of three technologies: AAC (advanced audio codec), SBR (spectral band replication) and PS (parametric stereo). All the 3 technologies are defined in MPEG4 audio standard [2]. The combination of AAC and SBR is called HE-AAC (also known as “aacplus v1”). AAC is a general audio codec, SBR is a bandwidth extension technique offering substantial coding gain in combination with AAC, and Parametric Stereo (PS) enables stereo coding at very low bitrates.

The video and audio streams obtained from these compression schemes need to be multiplexed before they are transmitted over a broadcast channel. The multiplexing process aims at providing efficient transmission schemes such as time stamps for lip synchronization between the video and audio streams during playback, robust error codes to detect packet losses etc. In this thesis an effective algorithm is proposed for multiplexing the HEVC video with HE-AAC V2 audio elementary streams using MPEG-2 systems specifications [24]. The encoded audio and video undergoes two layers of packetization where first layer results in Program Elementary stream (PES) packets and second layer of packetization results in Transport Stream (TS) packets. Once these TS packets are multiplexed, they are de-multiplexed and decoded at the receiver end achieving lip synchronization during playback.

Figures 1.5 and 1.6 show the multiplexing and demultiplexing scheme implemented in this thesis.

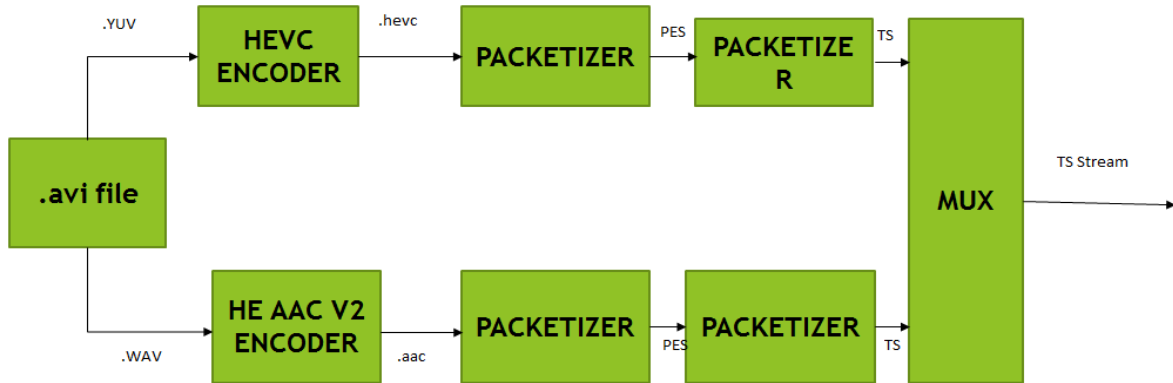


Fig 1.5 Block diagram of Multiplexing Scheme

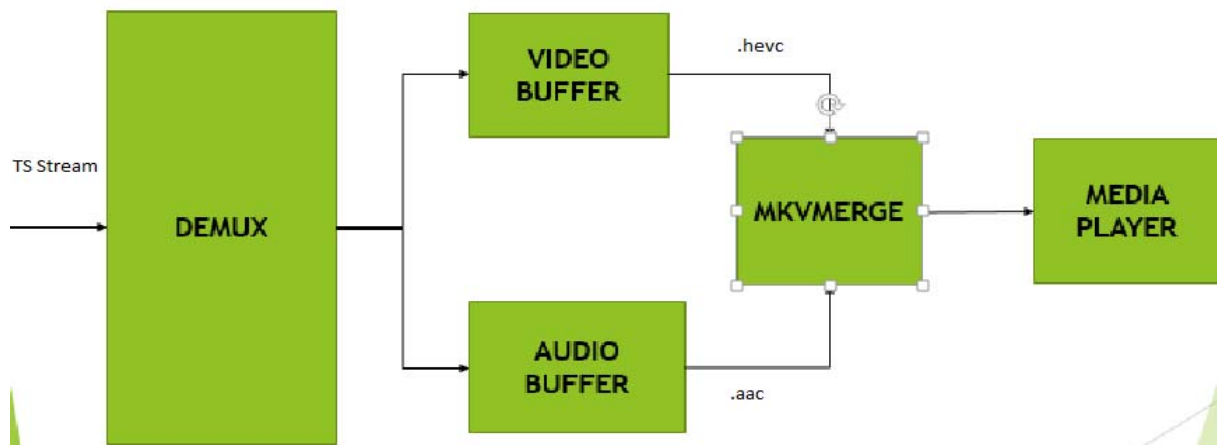


Fig 1.6 Block diagram of Demultiplexing Scheme

1.3 Thesis Outline

Chapters 2 and 3 give the brief overview of HEVC and HE-AAC v2 compression standards respectively.

Chapters 4 and 5 explain the multiplexing and de-multiplexing schemes used in this thesis.

Chapter 6 outline the results, conclusions and future work respectively.

CHAPTER – 2

Overview of High Efficiency Video Coding (HEVC / H.265)

2.1 Introduction

High Efficiency Video Coding (HEVC) [1] is an international standard for video compression developed by a working group of ISO/IEC MPEG (Moving Picture Experts Group) and ITU-T VCEG (Video Coding Experts Group). The main goal of HEVC standard is to significantly improve compression performance compared to existing standards (such as H.264/Advanced Video Coding [4]) in the range of 50% bit rate reduction at similar visual quality [1].

HEVC is designed to address existing applications of H.264/MPEG-4 AVC and to focus on two key issues: increased video resolution and increased use of parallel processing architectures [1] . Version 1 of the HEVC standard defines three profiles: **Main**, **Main 10**, and **Main Still Picture**. Version 2 of HEVC adds 21 range extensions profiles, two scalable extensions profiles, and one multi-view profile. It primarily targets consumer applications as pixel formats are limited to 4:2:0 8-bit and 4:2:0 10-bit. The next revision of the standard, finalized in 2014, enables new use-cases with the support of additional pixel formats such as 4:2:2 and 4:4:4 and bit depth higher than 10-bit [9], embedded bit-stream scalability, screen content coding [79] , 3D video [5] and multiview video [6] [80] .

2.2 Encoder and Decoder in HEVC [12]

Source video, consisting of a sequence of video frames, is encoded or compressed by a video encoder to create a compressed video bit stream. The compressed bit stream is stored or transmitted. A video decoder decompresses the bit stream to create a sequence of decoded frames.

The video encoder performs the following steps:

- Partitioning each picture into multiple units
- Predicting each unit using inter or intra prediction, and subtracting the prediction from the unit
- Transforming and quantizing the residual (the difference between the original picture unit and its prediction)
- Entropy encoding transform output, prediction information, mode information and headers

The video decoder performs the following steps:

- Entropy decoding and extracting the elements of the coded sequence
- Rescaling and inverting the transform stage
- Predicting each unit and adding the prediction to the output of the inverse transform
- Reconstructing a decoded video image

The Figures 2.1 [5] and 2.2 [13] represent the detailed block diagrams of HEVC encoder and decoder respectively:

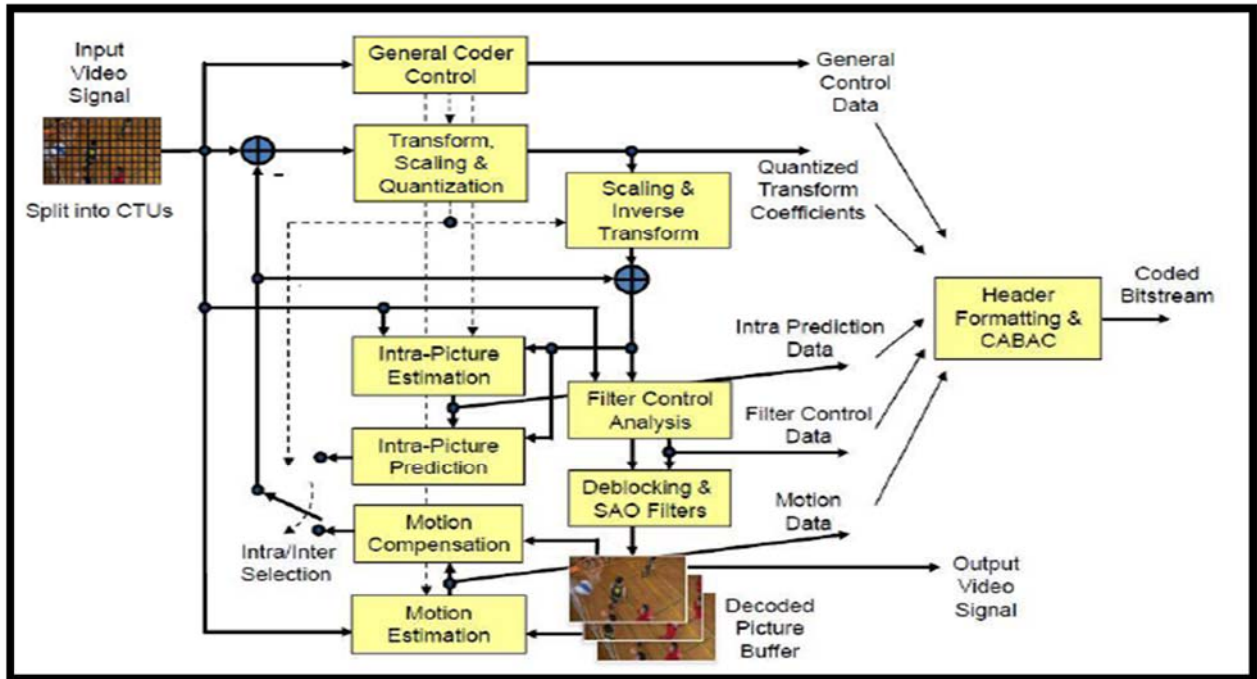


Fig 2.1 Block Diagram of HEVC Encoder [5]

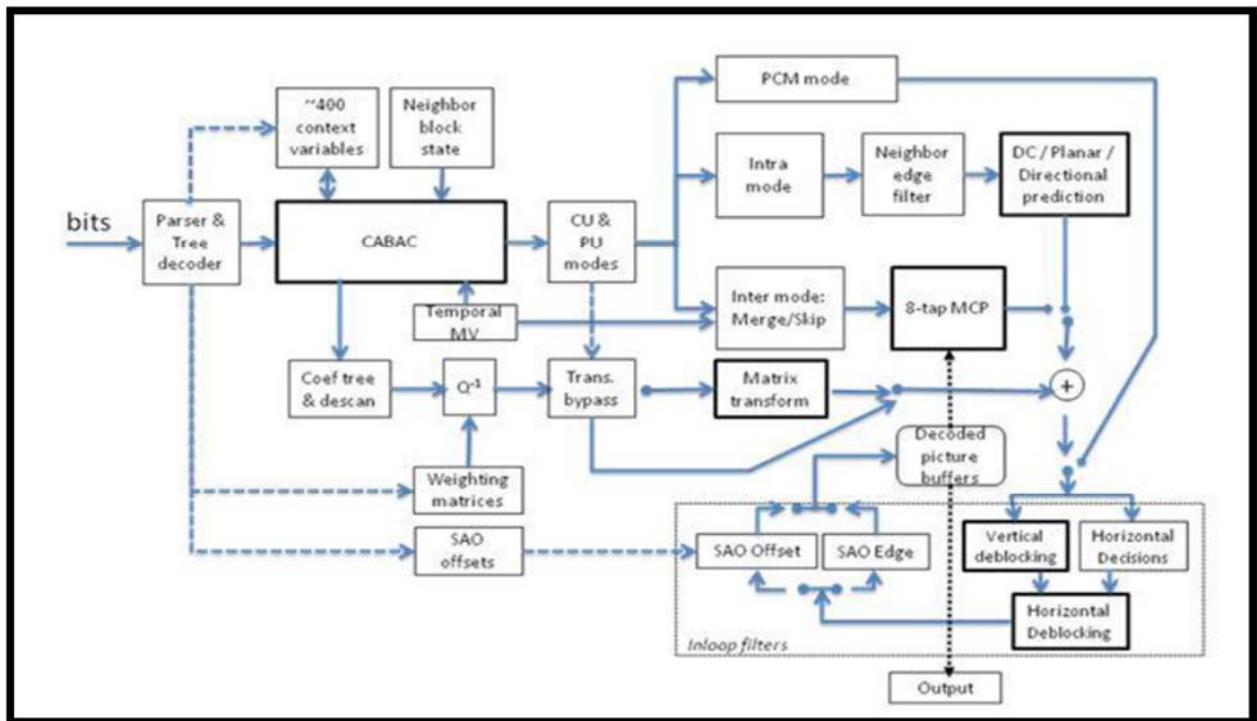


Fig 2.2 Block diagram of HEVC Decoder [13]

2.3 Features of HEVC

2.3.1 Partitioning :

2.3.1.1 Division of the Picture into Coding Tree Units:

A picture is partitioned into coding tree units (CTUs), which each contain luma CTBs and chroma CTBs. A luma CTB covers a rectangular picture area of $L \times L$ samples of the luma component and the corresponding chroma CTBs cover each $L/2 \times L/2$ samples of each of the two chroma components. The value of L may be equal to 16, 32, or 64 as determined by an encoded syntax element specified in the SPS. Compared with the traditional macroblock using a fixed array size of 16×16 luma samples, as used by all previous ITU-T and ISO/IEC JTC 1 video coding standards since H.261 (that was standardized in 1990), HEVC supports variable-size CTBs selected according to needs of encoders in terms of memory and computational requirements. The support of larger CTBs than in previous standards is particularly beneficial when encoding high-resolution video content. The luma CTB and the two chroma CTBs together with the associated syntax form a CTU. The CTU is the basic processing unit used in the standard to specify the decoding process [1].

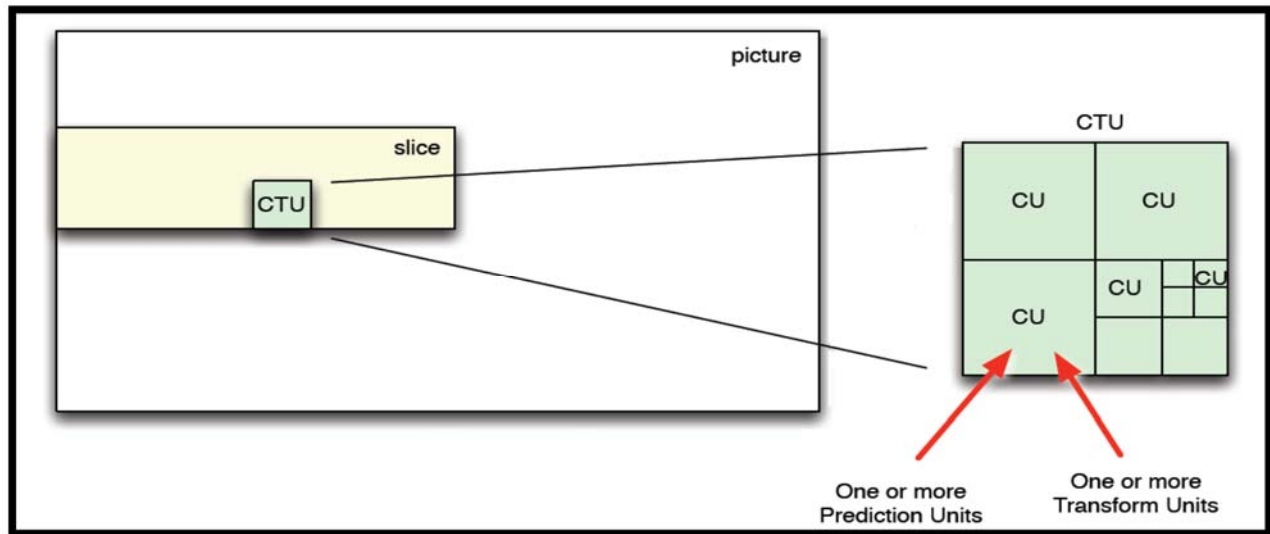


Fig 2.3 Picture, Slice, Coding Tree Unit (CTU), Coding Unit (CU) [12]

2.3.1.2 Division of the CTB into CBs:

The blocks specified as luma and chroma CTBs can be directly used as CBs or can be further partitioned into multiple CBs. Partitioning is achieved using tree structures. The tree partitioning in HEVC is generally applied simultaneously to both luma and chroma, although exceptions apply when certain minimum sizes are reached for chroma. The CTU contains a quadtree syntax that allows for splitting the CBs to a selected appropriate size based on the signal characteristics of the region that is covered by the CTB. The quadtree splitting process can be iterated until the size for a luma CB reaches a minimum allowed luma CB size that is selected by the encoder using

syntax in the SPS and is always 8×8 or larger (in units of luma samples). The boundaries of the picture are defined in units of the minimum allowed luma CB size. As a result, at the right and bottom edges of the picture, some CTUs may cover regions that are partly outside the boundaries of the picture. This condition is detected by the decoder, and the CTU quadtree is implicitly split as necessary to reduce the CB size to the point where the entire CB will fit into the picture [1].

2.3.1.3 PBs and PUs:

The prediction mode for the CU is signaled as being intra or inter, according to whether it uses intrapicture (spatial) prediction or interpicture (temporal) prediction. When the prediction mode is signaled as intra, the PB size, which is the block size at which the intrapicture prediction mode is established is the same as the CB size for all block sizes except for the smallest CB size that is allowed in the bitstream. For the latter case, a flag is present that indicates whether the CB is split into four PB quadrants that each have their own intrapicture prediction mode. The reason for allowing this split is to enable distinct intrapicture prediction mode selections for blocks as small as 4×4 in size.

When the luma intrapicture prediction operates with 4×4 blocks, the chroma intrapicture prediction also uses 4×4 blocks (each covering the same picture region as four 4×4 luma blocks). The actual region size at which the intrapicture prediction operates (which is distinct from the PB size, at which the intrapicture prediction mode is established) depends on the residual coding partitioning that is described as follows. When the prediction mode is signaled as inter, it is specified whether the luma and chroma CBs are split into one, two, or four PBs. The splitting into four PBs is allowed only when the CB size is equal to the minimum allowed CB size, using an equivalent type of splitting as could otherwise be performed at the CB level of the design rather than at the PB level. When a CB is split into four PBs, each PB covers a quadrant of the CB. When a CB is split into two PBs, six types of this splitting are possible. The partitioning possibilities for interpicture-predicted CBs are depicted in Figure 2.4 [1].

The upper partitions illustrate the cases of not splitting the CB of size $M \times M$, of splitting the CB into two PBs of size $M \times M/2$ or $M/2 \times M$, or splitting it into four PBs of size $M/2 \times M/2$. The lower four partition types in Fig. 3 are referred to as asymmetric motion partitioning (AMP), and are only allowed when M is 16 or larger for luma. One PB of the asymmetric partition has the height or width $M/4$ and width or height M , respectively, and the other PB fills the rest of the CB by having a height or width of $3M/4$ and width or height M . Each interpicture-predicted PB is assigned one or two motion vectors and reference picture indices. To minimize worst-case memory bandwidth, PBs of luma size 4×4 are not allowed for interpicture prediction, and PBs of luma sizes 4×8 and 8×4 are restricted to unipredictive coding. The interpicture prediction process is further described as follows. The luma and chroma PBs, together with the associated prediction syntax, form the PU [1].

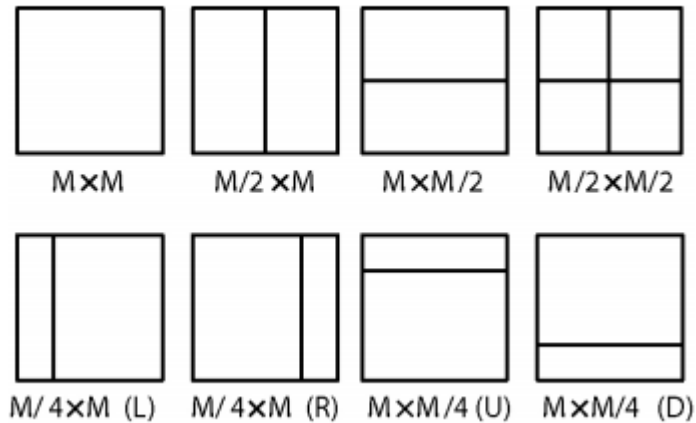


Fig 2.4 Modes for splitting a CB into PBs, subject to certain size constraints. For intrapicture-predicted CBs, only $M \times M$ and $M/2 \times M/2$ are supported [1]

2.3.1.4 Tree-Structured Partitioning Into Transform Blocks and Units:

For residual coding, a CB can be recursively partitioned into transform blocks (TBs). The partitioning is signaled by a residual quadtree. Only square CB and TB partitioning is specified, where a block can be recursively split into quadrants, as illustrated in Figure 2.5. For a given luma CB of size $M \times M$, a flag signals whether it is split into four blocks of size $M/2 \times M/2$. If further splitting is possible, as signaled by a maximum depth of the residual quadtree indicated in the SPS, each quadrant is assigned a flag that indicates whether it is split into four quadrants. The leaf node blocks resulting from the residual quadtree are the transform blocks that are further processed by transform coding. The encoder indicates the maximum and minimum luma TB sizes that it will use. Splitting is implicit when the CB size is larger than the maximum TB size.

Not splitting is implicit when splitting would result in a luma TB size smaller than the indicated minimum. The chroma TB size is half the luma TB size in each dimension, except when the luma TB size is 4×4 , in which case a single 4×4 chroma TB is used for the region covered by four 4×4 luma TBs. In the case of intrapicture-predicted CUs, the decoded samples of the nearest-neighbor TBs (within or outside the CB) are used as reference data for intrapicture prediction. In contrast to previous standards, the HEVC design allows a TB to span across multiple PBs for interpicture-predicted CUs to maximize the potential coding efficiency benefits of the quadtree-structured TB partitioning [1].

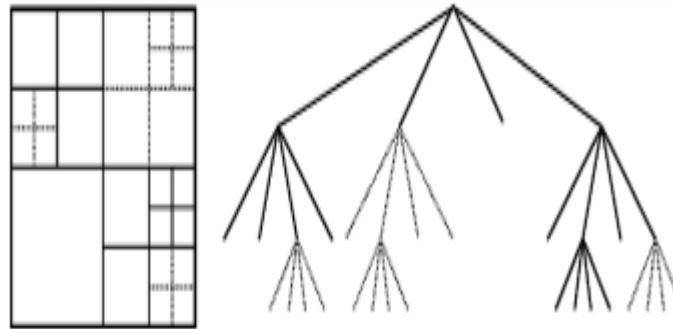


Fig 2.5 Subdivision of a CTB into CBs [and transform block (TBs)]. Solid lines indicate CB boundaries and dotted lines indicate TB boundaries. (a) CTB with its partitioning. (b) Corresponding quadtree [1]

2.3.2 Prediction :

2.3.2.1 Intrapicture Prediction :

Intrapicture prediction operates according to the TB size, and previously decoded boundary samples from spatially neighboring TBs are used to form the prediction signal. Directional prediction with 33 different directional orientations is defined for (square) TB sizes from 4×4 up to 32×32 . The possible prediction directions are shown in Figure 2.6. Alternatively, planar prediction (assuming an amplitude surface with a horizontal and vertical slope derived from the boundaries) and DC prediction (a flat surface with a value matching the mean value of the boundary samples) can also be used. For chroma, the horizontal, vertical, planar, and DC prediction modes can be explicitly signaled, or the chroma prediction mode can be indicated to be the same as the luma prediction mode (and, as a special case to avoid redundant signaling, when one of the first four choices is indicated and is the same as the luma prediction mode, the Intra-Angular mode is applied instead) [1].

HEVC supports various intrapicture predictive coding methods referred to as Intra-Angular, Intra-Planar, and Intra-DC. HEVC supports a total of 33 Intra-Angular prediction modes and Intra-Planar and Intra-DC prediction modes for luma prediction for all block sizes. Due to the increased number of directions, HEVC considers three most probable modes (MPMs) when coding the luma intrapicture prediction mode predictively, rather than the one most probable mode considered in H.264/MPEG-4 AVC.

Among the three most probable modes, the first two are initialized by the luma intrapicture prediction modes of the above and left PBs if those PBs are available and are coded using an intrapicture prediction mode. Any unavailable prediction mode is considered to be Intra-DC. The PB above the luma CTB is always considered to be unavailable in order to avoid the need to store a line buffer of neighboring luma prediction modes. When the first two most probable modes are not equal, the third most probable mode is set equal to Intra-Planar, Intra-DC, or Intra-Angular

(vertical), according to which of these modes, in this order, is not a duplicate of one of the first two modes. When the first two most probable modes are the same, if this first mode has the value Intra-Planar or Intra-DC, the second and third most probable modes are assigned as Intra-Planar, Intra-DC, or Intra-Angular, according to which of these modes, in this order, are not duplicates. When the first two most probable modes are the same and the first mode has an Intra-Angular value, the second and third most probable modes are chosen as the two angular prediction modes that are closest to the angle (i.e., the value of k) of the first. In the case that the current luma prediction mode is one of three MPMs, only the MPM index is transmitted to the decoder.

Otherwise, the index of the current luma prediction mode excluding the three MPMs is transmitted to the decoder by using a 5-b fixed length code. For chroma intrapicture prediction, HEVC allows the encoder to select one of five modes: Intra-Planar, Intra-Angular (vertical), Intra-Angular (horizontal), Intra-DC, and Intra-Derived. The Intra-Derived mode specifies that the chroma prediction uses the same angular direction as the luma prediction. With this scheme, all angular modes specified for luma in HEVC can, in principle, also be used in the chroma prediction, and a good tradeoff is achieved between prediction accuracy and the signaling overhead. The selected chroma prediction mode is coded directly (without using an MPM prediction mechanism) [1].

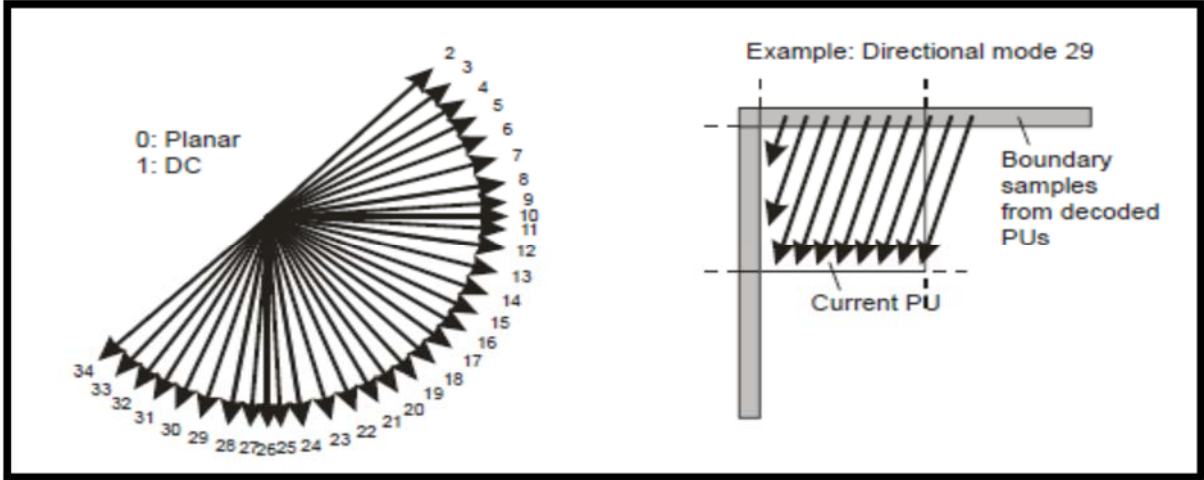


Fig 2.6 Modes and angular intra prediction directions in HEVC [1]

2.3.2.2 Interpicture Prediction :

Compared to intrapicture-predicted CBs, HEVC supports more PB partition shapes for interpicture-predicted CBs. The partitioning modes of PART-2N×2N, PART-2N×N, and PART-N×2N indicate the cases when the CB is not split, split into two equal-size PBs horizontally, and split into two equal-size PBs vertically, respectively. PART-N×N specifies that the CB is split into four equal-size PBs, but this mode is only supported when the CB size is equal to the smallest allowed CB size. In addition, there are four partitioning types that support splitting

the CB into two PBs having different sizes: PART- $2N \times nU$, PART- $2N \times nD$, PART- $nL \times 2N$, and PART- $nR \times 2N$. These types are known as asymmetric motion partitions [1].

2.3.2.2 Fractional Sample Interpolation:

The samples of the PB for an intrapicture-predicted CB are obtained from those of a corresponding block region in the reference picture identified by a reference picture index, which is at a position displaced by the horizontal and vertical components of the motion vector. Except for the case when the motion vector has an integer value, fractional sample interpolation is used to generate the prediction samples for noninteger sampling positions.

As in H.264/MPEG-4 AVC, HEVC supports motion vectors with units of one quarter of the distance between luma samples. For chroma samples, the motion vector accuracy is determined according to the chroma sampling format, which for 4:2:0 sampling results in units of one eighth of the distance between chroma samples. The fractional sample interpolation for luma samples in HEVC uses separable application of an eight-tap filter for the half-sample positions and a seven-tap filter for the quartersample positions.

This is in contrast to the process used in H.264/MPEG-4 AVC, which applies a two-stage interpolation process by first generating the values of one or two neighboring samples at half-sample positions using six-tap filtering, rounding the intermediate results, and then averaging two values at integer or half-sample positions. HEVC instead uses a single consistent separable interpolation process for generating all fractional positions without intermediate rounding operations, which improves precision and simplifies the architecture of the fractional sample interpolation.

The interpolation precision is also improved in HEVC by using longer filters, i.e., seven-tap or eight-tap filtering rather than the six tap filtering used in H.264/MPEG-4 AVC. Using only seven taps rather than the eight used for half-sample positions was sufficient for the quarter-sample interpolation positions since the quarter-sample positions are relatively close to integer sample positions, so the most distant sample in an eight-tap interpolator would effectively be farther away than in the half sample case (where the relative distances of the integer-sample positions are symmetric). The actual filter tap values of the interpolation filtering kernel were partially derived from DCT basis function equations.

In Figure 2.7, the positions labeled with upper-case letters, $A_{i,j}$, represent the available luma samples at integer sample locations, whereas the other positions labeled with lower-case letters represent samples at non integer sample locations, which need to be generated by interpolation.

$A_{-1,-1}$				$A_{0,-1}$	$a_{0,-1}$	$b_{0,-1}$	$c_{0,-1}$	$A_{1,-1}$				$A_{2,-1}$
$A_{-1,0}$				$A_{0,0}$	$a_{0,0}$	$b_{0,0}$	$c_{0,0}$	$A_{1,0}$				$A_{2,0}$
$d_{-1,0}$				$d_{0,0}$	$e_{0,0}$	$f_{0,0}$	$g_{0,0}$	$d_{1,0}$				$d_{2,0}$
$h_{-1,0}$				$h_{0,0}$	$i_{0,0}$	$j_{0,0}$	$k_{0,0}$	$h_{1,0}$				$h_{2,0}$
$n_{-1,0}$				$n_{0,0}$	$p_{0,0}$	$q_{0,0}$	$r_{0,0}$	$n_{1,0}$				$n_{2,0}$
$A_{-1,1}$				$A_{0,1}$	$a_{0,1}$	$b_{0,1}$	$c_{0,1}$	$A_{1,1}$				$A_{2,1}$
$A_{-1,2}$				$A_{0,2}$	$a_{0,2}$	$b_{0,2}$	$c_{0,2}$	$A_{1,2}$				$A_{2,2}$

Fig 2.7 Integer and fractional sample positions for luma interpolation [1]

2.3.3 Transform and Quantization [12] :

Any residual data remaining after prediction is transformed using a block transform based on the Integer Discrete Cosine Transform (DCT) [14] or Discrete Sine Transform (DST). One or more block transforms of size 32x32, 16x16, 8x8 and 4x4 are applied to residual data in each CU. A version related to DST is applied for 4x4 intra luma blocks. Then transformed data is quantized.

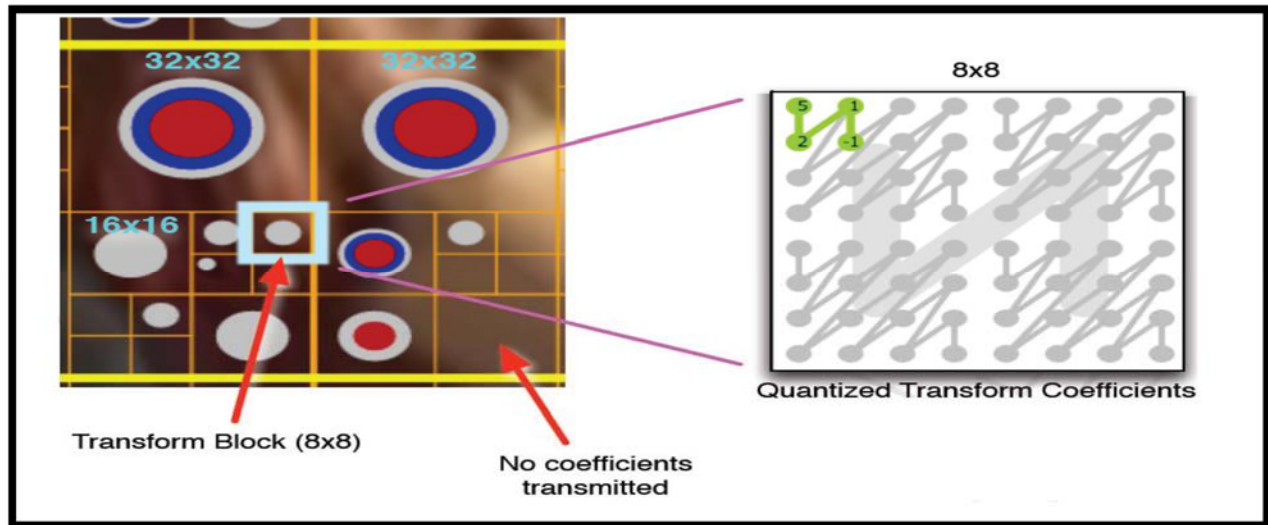


Fig 2.8 CTU showing range of transform (TU) sizes [12]

2.3.4 In loop Filtering [34]:

Loop filtering is a means to improve the reconstruction quality of the picture for display. Since the filter is located within the loop, the enhancement not only affects the quality of the output pictures, but also the reference pictures which are available for prediction when coding succeeding pictures.

2.3.4.1 Deblocking Filter:

The deblocking filter is applied to all samples adjacent to a PU or TU boundary except the case when the boundary is also a picture boundary, or when deblocking is disabled across slice or tile boundaries (which is an option that can be signaled by the encoder). It should be noted that both PU and TU boundaries should be considered since PU boundaries are not always aligned with TU boundaries in some cases of interpicture-predicted CBs. Syntax elements in the SPS and slice headers control whether the deblocking filter is applied across the slice and tile boundaries.[1][3]

2.3.4.2 Sample Adaptive Offset (SAO) :

A non-linear amplitude mapping is introduced in the inter-picture prediction loop after the deblocking filter. The goal is to better reconstruct the original signal amplitudes by using a look-up table that is described by a few additional parameters that can be determined by histogram analysis at the encoder side. [3]

2.3.5 Entropy coding:

A coded HEVC bit stream consists of quantized transform coefficients, prediction information such as prediction modes and motion vectors, partitioning information and other header data. All of these elements are encoded using Context Adaptive Binary Arithmetic Coding (CABAC) [15] similar to H.264/AVC.

2.4 Profiles in HEVC [34]

The three profiles are specified in the first version of HEVC, called the Main, the Main 10, and the Main Still Picture Profile. The profiles are characterized by a hierarchical structure as shown in the figure 2.6

2.4.1 Main Profile :

The Main Profile allows only 8 bit video bit depth. Since a bit depth of 8 bit is employed by many current applications ranging from video communications over streaming to broadcast, this profile is expected to find broad use.

2.4.2 Main 10 Profile :

The Main 10 Profile allows video bit depth in the range of 8 to 10. Thereby, the application range for the specification is extended towards high – quality applications and professional use. Since the bit depth is increased, the memory consumption increases accordingly.

2.4.3 Main Still Picture Profile:

While Main 10 profile is an extension of Main Profile, Main Still Picture Profile represents a subset of it. This requires the bit stream to contain only one single picture.

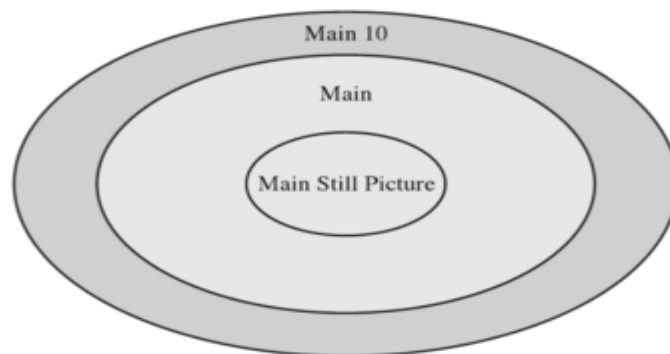


Fig 2.9 Hierarchy of the Main Profiles in HEVC [34]

2.5 Parallel decoding syntax and modified slice structuring

Four new features are introduced in the HEVC standard to enhance the parallel processing capability or modify the structuring of slice data for packetization purposes.

2.5.1 Tiles:

The option to partition a picture into rectangular regions called tiles has been specified. The main purpose of tiles is to increase the capability for parallel processing rather than provide error resilience. Tiles are independently decodable regions of a picture that are encoded with some shared header information. Tiles provide parallelism at a more coarse level of granularity (picture/subpicture), and no sophisticated synchronization of threads is necessary for their use. [1]

2.5.2 Wavefront parallel processing:

When wavefront parallel processing (WPP) is enabled, a slice is divided into rows of CTUs. The first row is processed in an ordinary way, the second row can begin to be processed after only two CTUs have been processed in the first row, and the third row can begin to be processed after only two CTUs have been processed in the second row, and so on. WPP provides a form of processing parallelism at a rather fine level of granularity, i.e., within a slice. [1][2]

2.5.3 Dependent slice segments:

A structure called a dependent slice segment allows data associated with a particular wavefront entry point or tile to be carried in a separate NAL unit, and thus potentially makes that data available to a system for fragmented packetization with lower latency than if it were all coded together in one slice. [1]

2.5.4 Slices:

A slice is a data structure that can be decoded independently from other slices of the same picture, in terms of entropy coding, signal prediction, and residual signal reconstruction. A slice can either be an entire picture or a region of a picture. One of the main purposes of slices is resynchronization in the event of data losses. [1][2]

2.6 Bit stream syntax of H.265:

The high-level syntax of HEVC mainly contains from the Network Adaptation Layer (NAL) [1] of H.264/MPEG-4 AVC. The NAL provides the ability to map the Video Coding Layer (VCL) data that represent the content of the pictures onto various transport layers, including RTP/IP [16], ISO MP4, and H.222.0/MPEG-2 [17] Systems, and provide a framework for packet loss resilience [18]. The comparison between NAL units of H.264 and

HEVC is shown in figure 2.10.

In HEVC each slice is encoded in a single NAL unit. HEVC uses a two byte NAL unit header. The size of a slice (and the subsequent NAL unit) may be matched to that of the Maximum Transmission Unit (MTU) of the network, over which the video will be streamed. NAL units are classified into VCL and non-VCL NAL units according to whether they contain coded pictures or other associated data, respectively as shown in table 2.1 [1] [16].

TABLE I
NAL UNIT TYPES, MEANINGS, AND TYPE CLASSES

Type	Meaning	Class
0, 1	Slice segment of ordinary trailing picture	VCL
2, 3	Slice segment of TSA picture	VCL
4, 5	Slice segment of STSA picture	VCL
6, 7	Slice segment of RADL picture	VCL
8, 9	Slice segment of RASL picture	VCL
10–15	Reserved for future use	VCL
16–18	Slice segment of BLA picture	VCL
19, 20	Slice segment of IDR picture	VCL
21	Slice segment of CRA picture	VCL
22–31	Reserved for future use	VCL
32	Video parameter set (VPS)	non-VCL
33	Sequence parameter set (SPS)	non-VCL
34	Picture parameter set (PPS)	non-VCL
35	Access unit delimiter	non-VCL
36	End of sequence	non-VCL
37	End of bitstream	non-VCL
38	Filler data	non-VCL
39, 40	SEI messages	non-VCL
41–47	Reserved for future use	non-VCL
48–63	Unspecified (available for system use)	non-VCL

Table 2.1 The NAL unit types and their associated meanings, classes in the HEVC standard. [1]

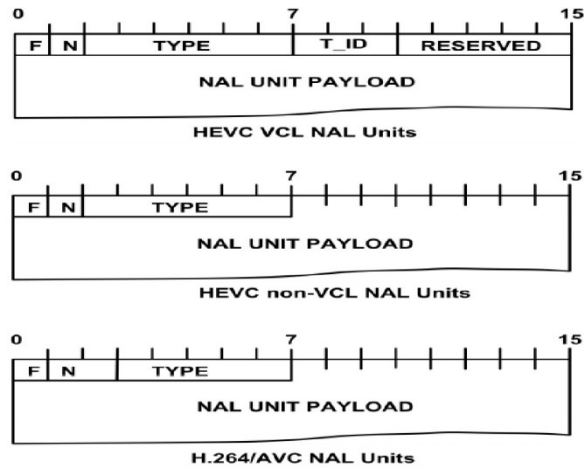


Fig 2.10 Comparison of HEVC and H.264 NAL units [16]

2.7 Summary

This chapter provided an overview of High Efficiency Video Coding. An overview of HE-AAC V2 is presented in the next chapter.

CHAPTER – 3

Overview of HE AAC v2

3.1 Introduction

Delivering broadcast-quality content to consumers is one of the most challenging tasks in the new world of digital broadcasting. One of the most critical aspects is the highly efficient use of the available transmission spectrum. Consequently, a careful choice of compression schemes for media content is essential – for both the technical and the economical feasibility of modern digital broadcasting systems [23].

In the case of audio content, the MPEG-4 High Efficiency AAC v2 profile (HE-AAC v2) has proven, in several independent tests, to be the most efficient audio compression scheme available worldwide. It has recently been selected within DVB as part of its overall codec toolbox [23].

High efficiency advanced audio codec version 2 also known as enhanced aacplus is a low bit rate audio codec defined in MPEG4 audio profile [2] belonging to the AAC family. It is specifically designed for low bit rate applications such as streaming.

HE-AAC v2 comprises a fully-featured tool set for the coding of audio signals in mono, stereo and multichannel modes (up to 48 channels) – at high quality levels using a wide range of bitrates [23].

HE AAC v2 has been proven to be the most efficient audio compression tool available today. The codec’s core components are already in widespread use in a variety of systems and applications where bandwidth limitations are a crucial issue, amongst them [23]:

- XM Satellite Radio – the digital satellite broadcasting service in the USA;
- HD Radio – the terrestrial digital broadcasting system of iBiquity Digital in the USA;
- Digital Radio Mondiale – the international standard for broadcasting in the long-, medium- and short-wave bands.

In Asia, HE-AAC v2 is the mandatory audio codec for the Korean Satellite Digital Multimedia Broadcasting (S-DMB) [22] technology and is optional for Japan’s terrestrial Integrated Services Digital Broadcasting system (ISDB) [20]. HE-AAC v2 is also a central element of the 3GPP and 3GPP2 [21] specifications and is applied in multiple music download services over 2.5 and 3G mobile communication networks [23].

HE AAC v2 is a combination of three technologies: AAC (advanced audio codec), SBR (spectral band replication) and PS (parametric stereo). All the 3 technologies are defined in MPEG4 audio standard [2]. The combination of AAC and SBR is called HE-AAC (also known as “aacplus v1”). AAC is a general audio codec, SBR is a bandwidth extension technique offering substantial coding gain in combination with AAC, and Parametric Stereo (PS) enables stereo coding at very low bitrates.

Figure 3.1 shows the family of AAC audio codecs.

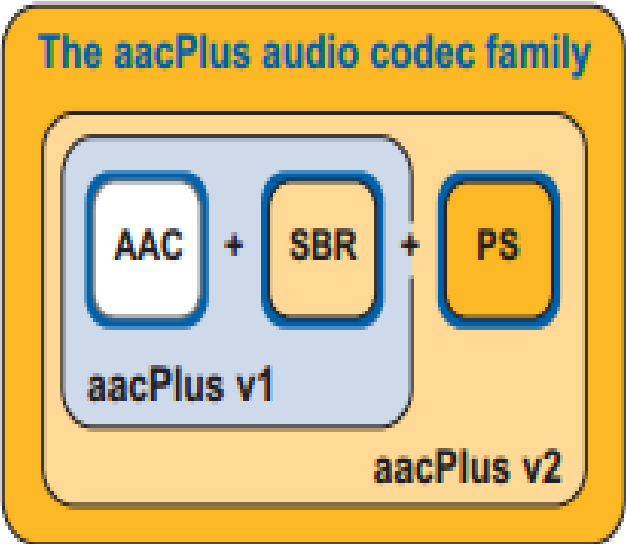


Fig 3.1 HE - AAC Audio Codec Family [23]

3.2 Architecture of HE-AAC v2 [23]

The underlying core codec of HE-AAC v2 is the well-known MPEG AAC codec. AAC is considered state-of-the-art for transparent audio quality at a typical bitrate of 128 kbit/s. Below this rate, the audio quality of AAC would start to degrade, which can be compensated to a maximum degree with the enhancement techniques SBR and PS.

SBR is a bandwidth extension technique that enables audio codecs to deliver the same listening experience at approximately half the bitrate that the core codec would require, if operated on its own.

Parametric Stereo increases the coding efficiency a second time by exploiting a parametric representation of the stereo image of a given input signal. Thus, HE-AAC v2 is a superset rather than a substitute for the AAC core codec and extends the reach of high-quality MPEG-4 audio to much lower bitrates. Given this superset architecture, HE-AAC v2 decoders are also capable of decoding plain AAC bit-streams, as well as bit-streams incorporating AAC and SBR data components, i.e. HE-AAC bit-streams. Hence, HEAAC v2 is also a superset of HE-AAC, providing the highest level of flexibility for broadcasters as it contains all the technical components necessary for audio compression over a high bitrate range.

Another important feature of the HE-AAC and HE-AAC v2 architecture is the extremely flexible transport of metadata. Metadata can be embedded as auxiliary data in a way that only compatible decoders take notice of their existence. Non-compatible decoders simply ignore the metadata. A high flexibility is provided in terms of type, amount and usage of the data. Metadata plays an important role in digital broadcasting, e.g. as content description data such as the name of an artist or song, or as system-related data such as control information for a given decoder. In broadcasting especially, metadata – such as DRC (Dynamic Range Control), DN (Dialog normalization), or down mixing from multichannel to stereo – is widely used to achieve adequate reproduction of the original programme material in particular listening environments [23].

3.3 MPEG AAC

Research on perceptual audio codecs started about twenty years ago [23]. Earlier research on the human auditory system had revealed that hearing is mainly based on a short-term spectral analysis of the audio signal. The so-called masking effect was observed: the human auditory system is not able to perceive distortions that are masked by a stronger signal in the spectral neighborhood. Thus, when looking at the short-term spectrum, a so-called masking threshold can be calculated for this spectrum. Distortions below this threshold are inaudible in the ideal case [23].

The goal is to calculate the masking threshold based on a psychoacoustic model and to process the audio signal in a way that only audible information resides in the signal. Ideally, the distortion introduced is exactly below the masking threshold and thus remains inaudible. If the compression rate is further increased, the distortion introduced by the codec violates the masking threshold and produces audible artefacts [23].

The main method of overcoming this problem in traditional perceptual waveform codecs is to limit the audio bandwidth. As a consequence, more information is available for the remainder of the spectrum, resulting in a clean but dull-sounding signal. Another method, called intensity stereo, can only be used for stereo signals. In intensity stereo, only one channel and some panning information is transmitted, instead of a left and a right channel. However, this is only of limited use in increasing the compression efficiency as, in many cases, the stereo image of the audio signal gets destroyed [23].

At this stage, research on classical perceptual audio coding had reached its limits, as the hitherto known methods did not seem to provide more potential to further increase the coding efficiency. Hence, a shift in paradigm was needed, represented by the idea that different elements of an audio signal, such as spectral components or the stereo image, deserve different tools if they are to be coded more efficiently. This idea led to the development of the enhancement tools, Spectral Band Replication and Parametric Stereo [23].

3.4 Spectral Band Replication (SBR)

SBR [2] is a bandwidth expansion technique; it has emerged as one of the most important tools that have led to the development of audio coding technology.

SBR exploits the correlation that exists between the energy of the audio signal at high and low frequencies also referred to as high and low bands. It is also based on the fact that psychoacoustic importance of high band is relatively low.

SBR uses a well guided technique called transposition to predict the energies at high band from low band. Besides just transposition, the reconstruction of the high band is conducted by transmitting some guiding information such as spectral envelope of the original signal, prediction error, etc. These are referred to as SBR data. The original and the high band reconstructed audio signal are shown in the figures 3.3 and 3.4 respectively.

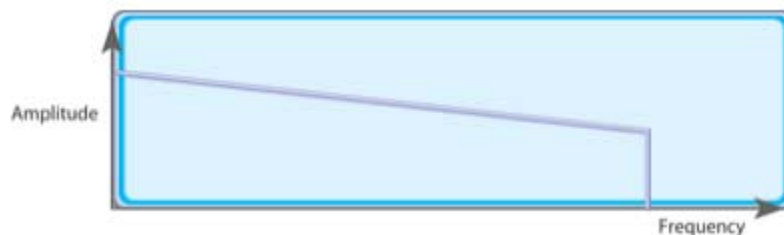


Fig 3.2 Original audio signal [35]

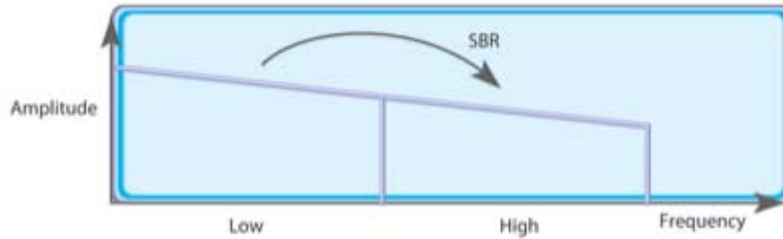


Fig 3.3 High band reconstruction through SBR [35]

SBR has enabled high-quality stereo sound at bitrates as low as 48 kbps. SBR was invented as a bandwidth extension tool when used along with AAC. It was adopted as an MPEG4 standard in March 2004 [2].

3.5 Parametric Stereo (PS)

Parametric stereo coding is a technique to efficiently code a stereo audio signal as a monaural signal plus a small amount of stereo parameters.

The monaural signal can be encoded using any audio coder. The stereo parameters can be embedded in the ancillary part of the mono bit stream creating backwards mono compatibility. In the decoder, first the monaural signal is decoded after which the stereo signal is reconstructed from the stereo parameters. Figure 3.5 shows the basic principle of the parametric stereo coding process.

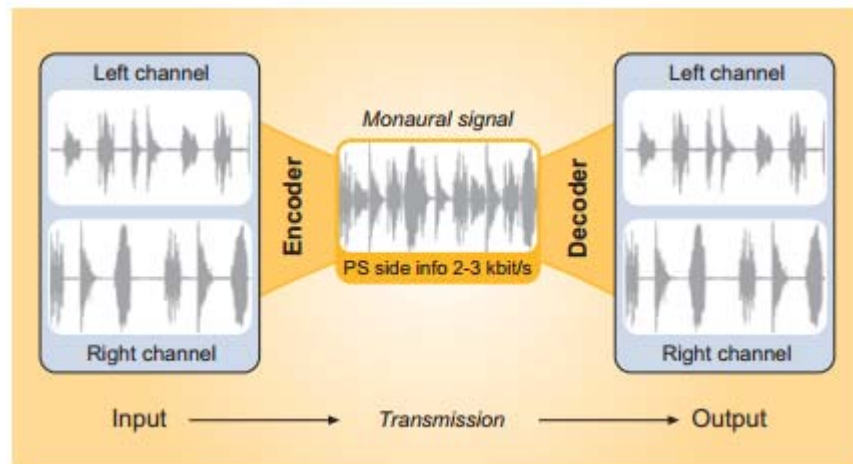


Fig 3.4 Basic principle of the parametric stereo coding process [23].

PS coding has led to a high quality stereo sound reconstruction at relatively low bitrates. In the parametric approach, the audio signal or stereo image is separated into its transient, sinusoid, and noise components. Next, each component is re-represented via parameters that drive a model for the signal, rather than the standard approach of coding the actual signal itself.

PS uses three types of parameters to describe the stereo image:

Inter-channel Intensity Differences (IID): describes the intensity differences between the channels.

Inter-channel Phase Differences (IPD): describes the phase differences between the channels and

Inter-channel Coherence (IC): describes the coherence between the channels. The coherence is measured as the maximum of the cross-correlation as a function of time or phase [23].

In principle, these three parameters allow for a high quality reconstruction of the stereo image. However, the IPD parameters only specify the relative phase differences between the channels of the stereo input signal. They do not prescribe the distribution of these phase differences over the left and right channels.

Hence, a fourth type of parameter is introduced, describing an overall phase offset or Overall Phase Difference (OPD). In order to reconstruct the stereo image, in the PS decoder a number of operations are performed, consisting of scaling (IID), phase rotations (IPD/OPD) and decorrelation (IC).

3.6 Functionality of HE-AAC V2

The described technologies AAC, SBR and PS are the building blocks of the MPEG-4 HE-AAC v2 profile. The AAC codec is used to encode the low band, SBR encodes the high band, and PS encodes the stereo image in a parameterised form. In a typical aacPlus encoder implementation, the audio input signal at an input sampling rate of f_s is fed into a 64-band Quadrature Mirror Filter bank and transformed into the QMF domain.

If the Parametric Stereo tool is used (i.e. for stereo encoding at bitrates below ~ 36 kbit/s), the PS encoder extracts parametric stereo information based on the QMF samples. Furthermore, a stereoto-mono downmix is applied. With a 32-band QMF synthesis, the mono QMF representation is then transformed back into the time domain at half the sample rate of the audio signal, $f_s/2$. This signal is then fed into the AAC encoder. If the Parametric Stereo tool is not used, the audio signal is fed into a 2:1 resampler and, again, the downsampled audio signal is fed into the AAC encoder. The SBR encoder also works in the QMF domain; it extracts the spectral envelope and additional helper information to guide the replication process in the decoder. All encoded data is then multiplexed into a single bit-stream for transmission or storage. Figure 3.7 shows the block diagram of a complete HE-AAC v2 encoder [23].

In the HE-AAC v2 decoder, the bit-stream is first split into the AAC, SBR and PS data portions. The AAC decoder outputs a time domain low-band signal at a sample rate of $f_s/2$. The signal is then transformed into the QMF domain for further processing. The SBR processing results in a reconstructed high band in the QMF domain. The low and high bands are then merged into a full-band QMF representation.

If the Parametric Stereo tool is used, the PS tool generates a stereo representation in the QMF domain. Finally, the signal is synthesized by a 64- band QMF synthesis filter bank. The result is a time domain output signal at the full sampling rate f_s . Figure 3.8 shows the block diagram of a complete HE-AAC v2 decoder [23].

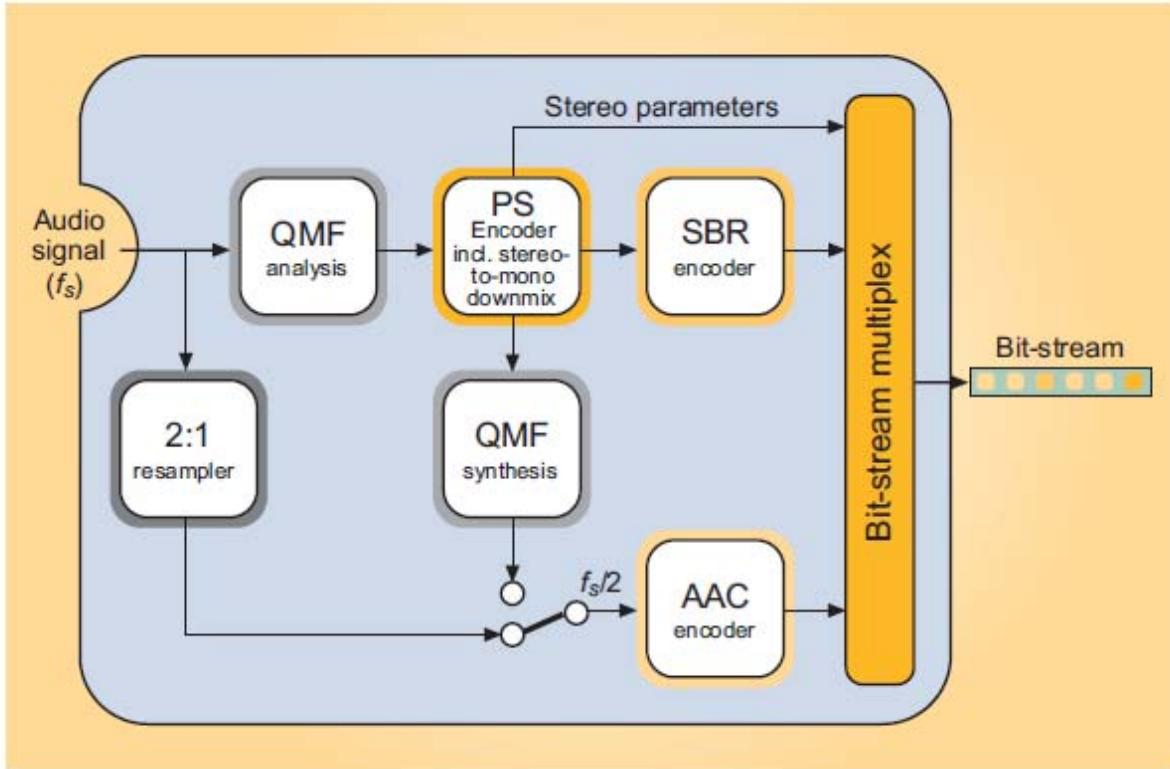


Fig 3.5 Block diagram of a complete HE-AAC v2 encoder [23]

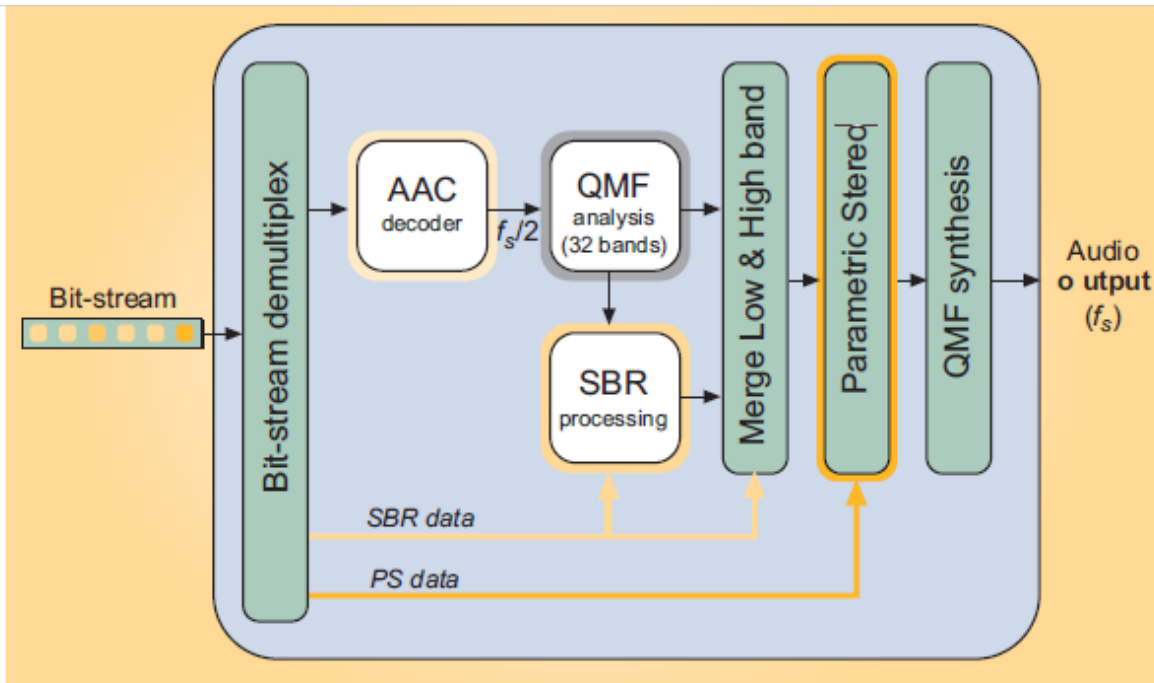


Fig 3.6 Block diagram of a complete HE-AAC v2 decoder [23]

3.7 Audio quality evaluation

The audio quality of HE-AAC and HE-AAC v2 has been evaluated in multiple double-blind listening tests conducted by independent entities such as the European Broadcasting Union (EBU), the Moving Picture Experts Group (MPEG), the 3rd Generation Partnership Project (3GPP), and the Institut für Rundfunktechnik (IRT).

Considering the fact that the quality of compressed audio signals scales with the bitrate, the following interpretation of the available test results can be made. Combining AAC with SBR and PS into HE-AAC v2 results in a very efficient audio codec, providing high audio quality over a wide bitrate range, with only moderate gradual reduction of the perceived audio quality towards very low bitrates. Figure 3.9 gives an impression of the anticipated audio quality vs. bitrate for the various codecs of the HE-AAC v2 family [23].

The diagram shows only a smooth degradation in audio quality of HE-AAC v2 towards low bitrates over a wide range down to 32 kbit/s. Even at bitrates as low as 24 kbit/s, HEAAC v2 still produces a quality far higher than that of any other audio codec available.

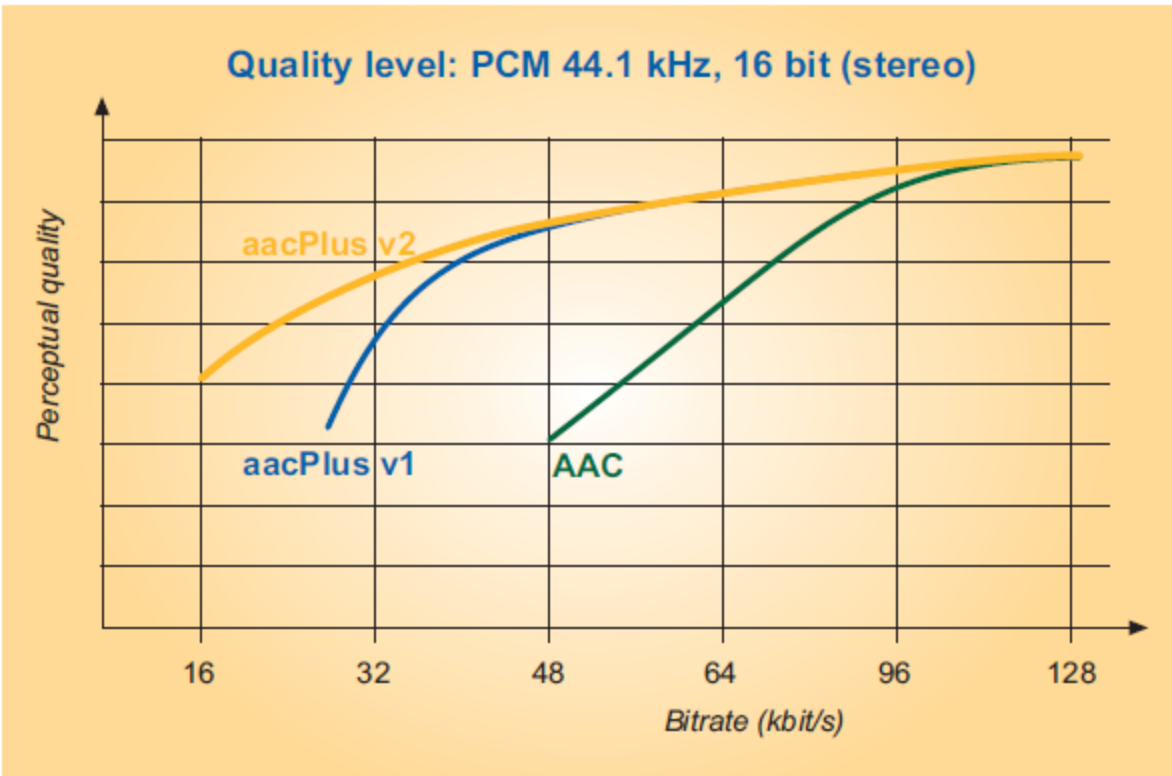


Fig 3.7 Anticipated audio quality vs. bitrate for the various codecs of the HE-AAC v2 family [23]

3.8 Advanced Audio Coding (AAC)

ISO/IEC 13818-7:2006 specifies MPEG-2 Advanced Audio Coding (AAC), a multi-channel audio coding standard that delivers higher quality than is achievable when requiring MPEG-1 backwards compatibility. It provides ITU-R "indistinguishable" quality at a data rate of 320 kbit/s for five full-bandwidth channel audio signals [71].

ISO/IEC 13818-7:2006 also supplements information on how to utilize the bandwidth extension technology (SBR) specified in ISO/IEC14496-3 in conjunction with MPEG-2 AAC [71].

The AAC encoder acts as the core encoding algorithm of the aacPlus system encoding at half the sampling rate of aacPlus. Since aacPlus implements the High Efficiency AAC Profile at Level 2

as defined in [3], the AAC LC object type is used. The AAC LC object type does not implement the Long Term Predictor (LTP) tool. The Level 2 implies a restriction to a maximum of two channels. Furthermore in case of SBR being used, the maximum AAC sampling rate is restricted to 24 kHz whereas if SBR is not used the maximum AAC sampling rate is restricted to 48 kHz [73].

The basic layout is depicted in Figure 3.8.

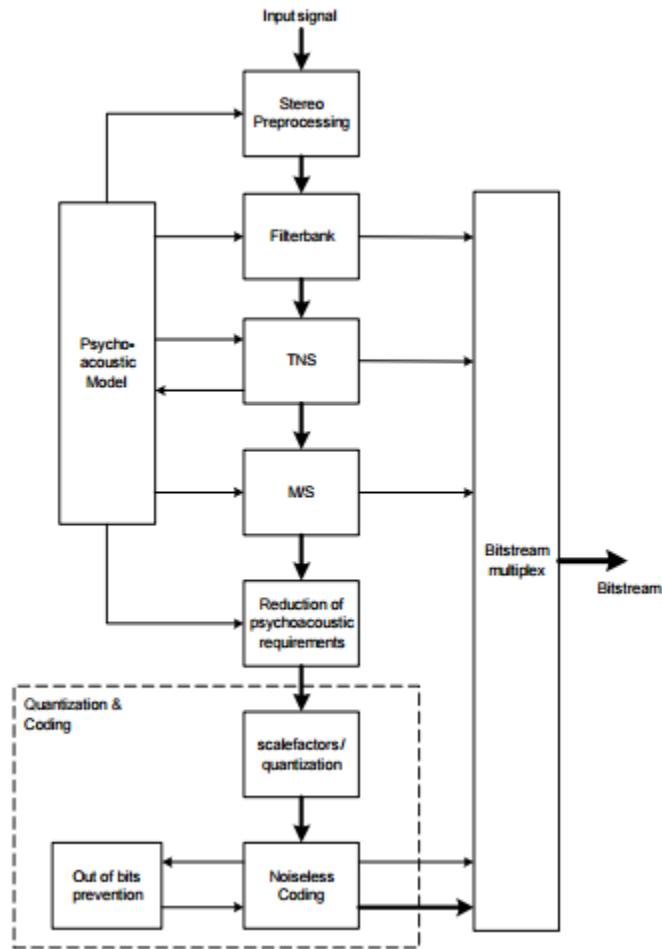


Fig 3.8 AAC Encoder Block Diagram [72]

- With stereo preprocessing, the stereo width of difficult to encode signals at low bitrates is reduced. Stereo preprocessing is active for bitrates less than 60kbit/s.

- The filterbank is an MDCT. The window length of the MDCT is either 2048 for the `only_long_sequence`, `long_start_sequence` and `long_stop_sequence` window sequence or 256 for the `eight_short_sequence` window sequence.
- The psychoacoustic model is simplified and works in combination with the quantization and coding strategy.
- The decision whether to use long windows with a window length of 2048 samples or a sequence of eight short blocks with a window length of 256 samples will be taken in the time domain. It is not possible to switch immediately between an `only_long_sequence` and an `eight_short_sequence`. Thus when switching from the long window transform to frames with eight short windows a `long_start_sequence` has to be inserted, resp. when switching back from short to long a `stop_window_sequence` is needed. Therefore there needs to be a lookahead of 1024+576 samples for the blockswitch decision.
- This technique does noise shaping in the time domain by doing an open loop prediction in the frequency domain. The TNS technique provides enhanced control of the location, in time, of quantization noise within a filter bank window. TNS proves to be especially successful for the improvement of speech quality at low bit-rates.
- M/S stereo coding is another data reduction module based on channel pair coding. In this case channel pair elements are analyzed as left/right and sum/difference signals on a block-by-block basis. In cases where the M/S channel pair can be represented by fewer bits, the spectral coefficients are coded, and a bit is set to note that the block has used M/S stereo coding. During decoding, the decoded channel pair is de-matrixed back to its original left/right state. For normal stereo operation, M/S Stereo, is only required when operating the encoder at bitrates at or above 44 kbps. Below 44 kbps the parametric stereo coding tool is used instead where the AAC core is operated in mono.
- Usually the requirements of the psychoacoustic model are too strong for the desired bitrate. Thus a threshold reduction strategy is necessary, i.e. the strategy reduces the requirements by increasing the thresholds given by the psychoacoustic model.
- A majority of the data reduction generally occurs in the quantization phase after the data has already achieved a certain level of compression when passed through the previous modules. This module contains many other blocks such as Scale factor quantization block, Noiseless coding and Out of Bits Prevention.
- Scale factor quantization: This block consist of two additional blocks called scale factor determination and scale factor difference reduction.
- Scale factor determination: The scale factors determine the quantization step size for each scale factor band. By changing the scale factor, the quantization noise will be controlled.
- Scale factor difference reduction: This block takes into account the difference of the scale factors which will be encoded. A smaller difference between two adjacent scale factors requires fewer bits.
- Noiseless coding: Coding of the quantized spectral coefficients is done by the noiseless coding. The encoder uses a so called greedy merge algorithm to segment the 1024 coefficients of a frame into section and to find the best Huffman codebook for each section.
- Out of Bits Prevention: after noiseless coding, the number of really needed bits is counted. If this number is too high, the number of bits has to be reduced.

3.9 HE-AAC v2 bitstream formats

HE-AAC v2 encoded data has variable file formats with different extensions, depending on the implementation and the usage scenario. The most commonly used file formats are the MPEG-4 file formats MP4 and M4A [26], carrying the respective extensions .mp4 and .m4a. The “.m4a” extension is used to emphasize the fact that a file contains audio only.

Additionally there are other bit stream formats, such as MPEG-4 ADTS (audio data transport stream) and ADIF (audio data interchange format). ADIF format has a single header at the beginning of the bit stream followed by raw audio data blocks. It is used mainly for local storage purposes. ADTS has a header before each access unit or audio frame and also the header information will remain same for all the frames in a stream. ADTS is more robust against errors and is suited for communication applications like broadcasting. For this thesis file format ADTS has been used.

Tables 3.1 and 3.2 describe the ADTS header. This is present before each access unit (a frame). This is later exploited for packetizing the frames into packetized elementary stream (PES) packets, which is the first layer of packetization before transport. Figure 3.7 shows the ADTS elementary stream. In this chapter, an overview of HE-AAC v2 audio coding standard is presented. The encoder, decoder, SBR, PS, AAC encoder and the bit stream format are described. Next chapter gives a brief overview of the transport protocols in particular MPEG 2 systems layer.

Field name	Number of bits		
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			
copyright_identification_start			
aac_frame_length	13	length of the 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	

Table 3.1: ADTS header format [2] [3]

profile bits	bits ID == 1 (MPEG-2 profile) ID == 0 (MPEG-4 Object type)
00 (0)	Main profile AAC MAIN
01 (1)	Low Complexity profile (LC) AAC LC
10 (2)	Scalable Sample Rate profile (SSR) AAC SSR
11 (3)	(reserved) AAC LTP

Table 3.2: Profile bits expansion [2] [3]

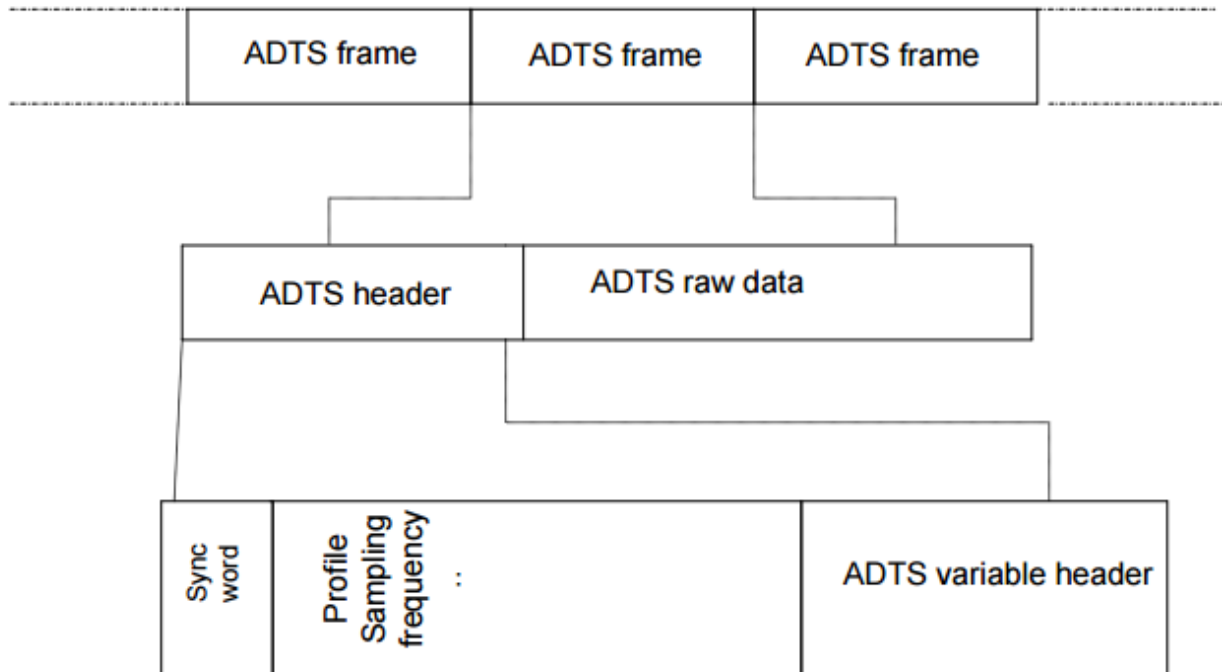


Fig 3.9: ADTS elementary stream [3].

3.10 Summary

In this chapter, an overview of HE-AAC v2 audio coding standard is presented. The encoder, decoder, SBR, PS, AAC encoder and the bit stream format are described. Next chapter gives a brief overview of multiplexing, packetization, packetized elementary streams and transport stream.

CHAPTER – 4

Multiplexing

4.1 Introduction

Multiplexing is the process of converting multiple elementary streams into a single transport stream for data transmission. In the case of digital television transmission standards (DTV) such as ATSC [50,51], DVB-T and DVB-H [52] or in the case of streaming technologies such as IPTV, video telephony services [52], it is necessary to encode both sound and picture signals and multiplex them along with other supplementary information into a format suitable for the transmission link [53]. In the case of high definition television (HDTV) services high quality video and audio data are transmitted which occupy a lot of bandwidth over a broadcast channel [30]. To address this issue the video and audio data are compressed using efficient compression schemes such as HEVC [5] for video and HE - AAC V2 [23] for audio which preserve the data quality but at the same time also reduces the bandwidth for transmission. A standard has been defined in MPEG-2 [54] by the (ISO)/ (IEC) on multiplexing and synchronizing the coded video, audio and other supplementary data. The same frame work with slight modifications has been adopted in this thesis for multiplexing the HEVC video and HE – AAC V2 audio elementary streams.

MPEG-2 TS provides a mechanism to encapsulate and multiplex coded video, coded audio, and generic data into a unified bitstream. In order to facilitate parsing of the information contained within the bitstream, in-band control information, known as PSI, is defined. The TS syntax also includes timing information in the form of timestamps in order to enable the real-time reproduction and precise synchronization of video, audio, and data (as applicable). A further design goal of the MPEG-2 Systems standard was to define a TS packetization format that would facilitate real-time transmission and reception of DTV over error-prone physical transmission paths, including over-the-air broadcasting and cable television networks. Fig. 4.1 contains a block diagram of a generic ATSC transmission/reception system. In typical implementations, the video and audio encoders and the ATSC multiplexer each create output bitstreams in the TS format. The consumer receiver, e.g., an integrated DTV receiver or a digital set-top box, includes inverse TS functions to recover the program information, e.g., a motion picture or a TV program, and deliver it to the decoders, which in turn present it to the viewer. [50]

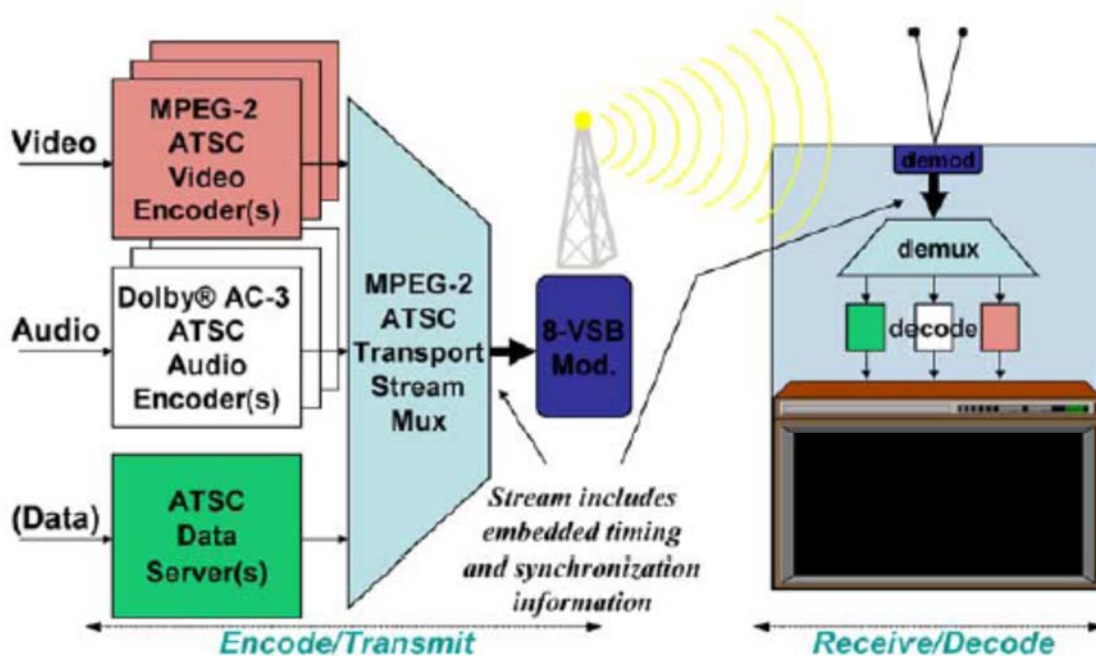


Fig 4.1 Example ATSC transmission/reception block diagram showing the MPEG-2 TS multiplexer. [50]

In order for an effective transmission and reception of the elementary streams the following factors should be considered while multiplexing these streams [30]:

- First, while multiplexing the elementary streams, every elementary stream should be given equal priority which is needed to prevent any buffer overflow or underflow at the receiver side which leads to loss in data packets.
- Secondly, apart from the coded elementary streams the multiplexed streams should also contain information to play the elementary streams with synchronization at the receiver. These types of additional information which are used for synchronization are called time stamps.
- Finally if the transmission takes place in an error prone network, certain additional information needs to be added in the multiplexed stream to detect these errors and rectify them.

4.2 MPEG-2 System Layers

MPEG-2 has three layers as shown in Fig 4.2. The system layer wraps around the compression layer. It is responsible for: Interleaving of Audio and Video/Multiplexing, Synchronization and Continuity, Transmission on Networks. [55]

MPEG-2 adds a transport layer. It has special methods for Error Control and Transmission over networks. [55]

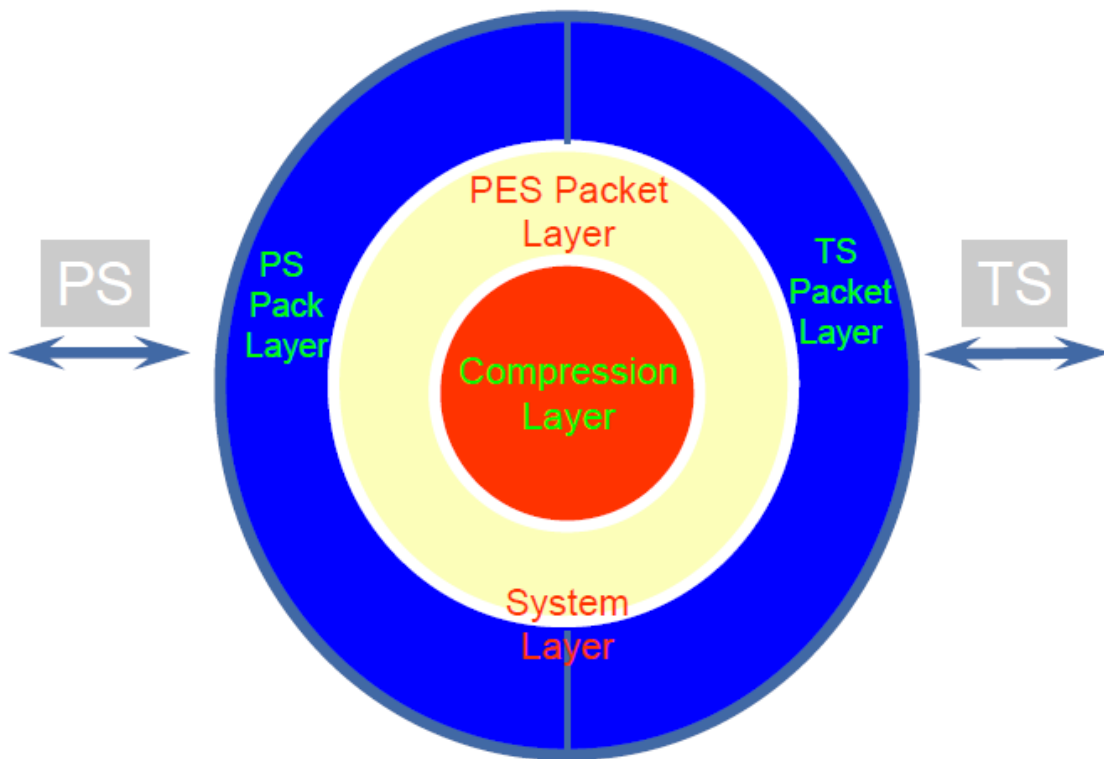


Fig 4.2 MPEG-2 System Layers [56]

The MPEG Systems Committee has defined a systems layer, specified by ISO-IEC 13818-1. (The MPEG-2 systems layer is commonly and confusingly known as MPEG-2 Transport because it defines a transport packet for transmission purposes.) [57] The MPEG-2 systems layer combines the various components of a digital program into a multi-program transport stream. These components include

- Compressed video
- Compressed audio
- Data
- Timing information
- System information
- Conditional access information
- Program-related data

The MPEG-2 systems layer includes the following functions: [57]

- Timing and synchronization—The transmission of timing information in transport packets to allow the receiver to synchronize its decoding rate with the encoder
- Packetization—The segmentation and encapsulation of elementary data streams into 188-byte transport packets.

- Multiplexing—The mechanisms used to combine compressed audio and video streams (elementary streams) into a transport stream.
- Conditional access—Provision for the transmission of conditional access information in the transport stream.

4.3 Packetization

Packetization is the first step in Multiplexing process. Figure 4.3 shows the two layer of packetization adopted in the MPEG-2 systems. A packet usually consists of two kinds of data namely the control information and the user data which is also known as payload. The time stamps and other control information are embedded in the control information header. There are two layers of packetization method adopted in the MPEG-2 systems . The first layer of packetization is known as the packetized elementary stream (PES) format and the second layer is known as the transport stream (TS) format. The multiplexing process takes place after the second layer of packetization which is used for transmission.

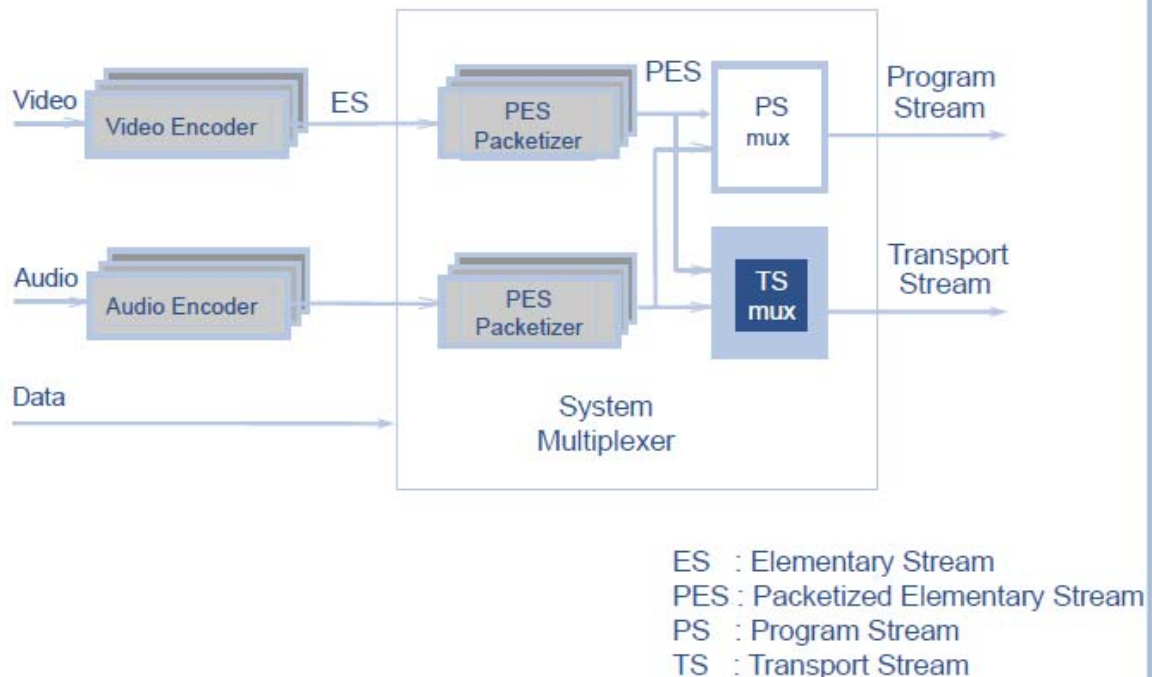


Fig. 4.3 Two layers of packetization method adopted in MPEG-2 systems [56]

The elementary stream produced by the audio or video encoder is segmented into a series of PES packets (as shown in Fig 4.4) typically along frame boundaries in order to facilitate random access to the content. A frame is defined as a progressive picture, both fields of an interlaced

picture, or a fixed number of audio samples. An access unit is either the coded representation of a picture or an audio frame. The PES packets, in turn, are further segmented into fixed-length TS packets to facilitate transmission in real time. The process of interleaving the TS packets of more than one Program into a single unified bitstream, while maintaining timing synchronization of each Program contained within, is known as multiplexing. The unified bitstream is called a multi-Program TS (MPTS) and is also referred to as a service multiplex.

The packet identifier (PID), contained in the header of each TS packet, is the key to sorting out the components or elements in the TS. The PID is used to locate the TS packets of a particular component stream within the service multiplex in order to facilitate the reassembly of the payload of each TS packet back into its higher level constructs, i.e., TS packets into PES packets and PES packets into an elementary stream. A series of TS packets containing the same PID includes either a single program element, e.g., a video elementary stream, or descriptive information about one or more program elements.

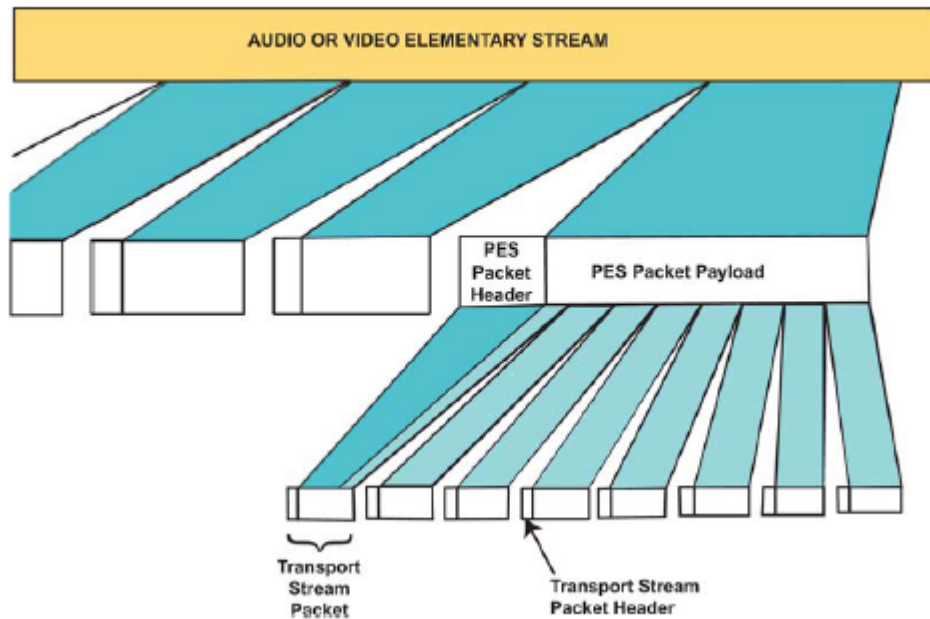


Fig 4.4 Packet structure hierarchy of a TS [50]

4.4 Packetized Elementary Stream (PES)

The MPEG-2 systems layer is responsible for the integration and synchronization of the elementary streams (ES): audio and video streams, as well as an unlimited number of data and control streams that can be used for various applications such as subtitles in multiple languages. This is accomplished by first packetizing the ESs thus forming the packetized elementary streams (PES) [59]. The PESs are subsequently multiplexed to form a single output stream for transmission in one of two modes: program stream (PS) and transport stream (TS). The PS is provided for error-free environments such as storage in CD-ROM. It is used for multiplexing PESs that share a common time-base, using long variable-length packets. The TS is designed for noisy environments such as communication over ATM networks. This mode permits multiplexing streams (PESs and PSs) that do not necessarily share a common time-base, using fixed-length (188 bytes) packets [59].

The packetized elementary streams consist of a header and a payload. The Elementary streams are separated into access units (Encoded audio/video frame data) and the encapsulation of audio/video data is performed. 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 length of PES header is 8 bytes which carries time stamp information in the form of frame numbers and additional information to facilitate in multiplexing the data [68].

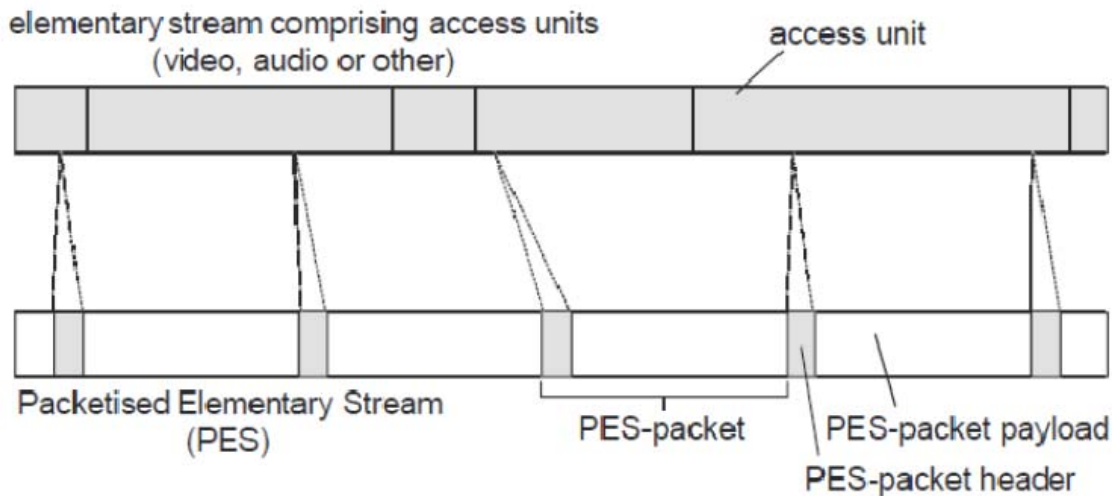


Fig 4.5 Encapsulation of PES from elementary streams [69]

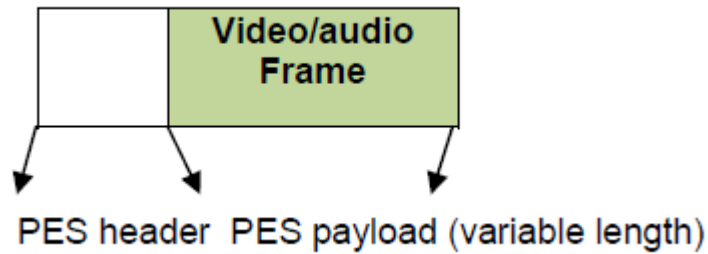


Fig 4.6 Structure of PES packet [29]

Table 4.1 describes the PES header fields.

PES packet start prefix - The first 3 bytes (0x000001) represent the start of a PES packet.

Video/audio stream ID - This is the byte following Start prefix which distinguishes between an audio and a video stream ID. 0xC0 is set for a video stream ID and 0xE0 is set for an audio stream ID chosen from a range of 0xC0 – 0xEF [68]. This byte along with start prefix byte is known as the start code, which determines that the PES packet is valid.

PES packet length - If the length of the PES packet length is not specified or unbounded i.e. (exceeds 65536 bytes), then this value is set to zero . Assuming, that the PES packet length does not exceed the allocated space, the reset option is not included in the algorithm implemented.

Time Stamps - The time stamp information is carried in the form of frame numbers which helps in achieving sync during playback.

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 as time stamps

Table 4.1 PES header description [68]

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

HEVC bitstream is searched for a 4 byte prefix start sequence 0x00000001 which indicates the beginning of a NAL unit. Then the 5 bit frame type is extracted from the NAL header and checked if it is a video frame or a parameter set. Parameter sets are very important and are required for decoding process. So if a parameter set is found (both PPS and SPS) it is packetized separately and transmitted. If NAL unit contains the slice data, then frame number is calculated from beginning of the stream and coded as time stamp in PES. It has to be noted that parameter sets are not counted as frames, so while coding parameter sets the time stamp field is coded as zero. 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. The payload information is encapsulated in the header to form a video PES packet.

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, to form transport streams packets.

The PES packets are used for constructing both of PS and TS. The PES packets include a AU (Access Unit) in the most of applications typically, but not mandatory. The PES packet delivers PTS or DTS. The PES packets are identified by stream_id. [56]

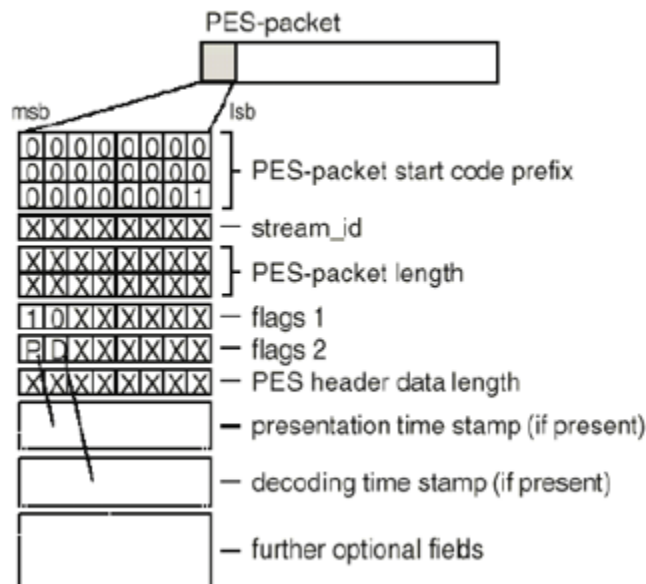


Fig 4.7 A PES packet header [56]

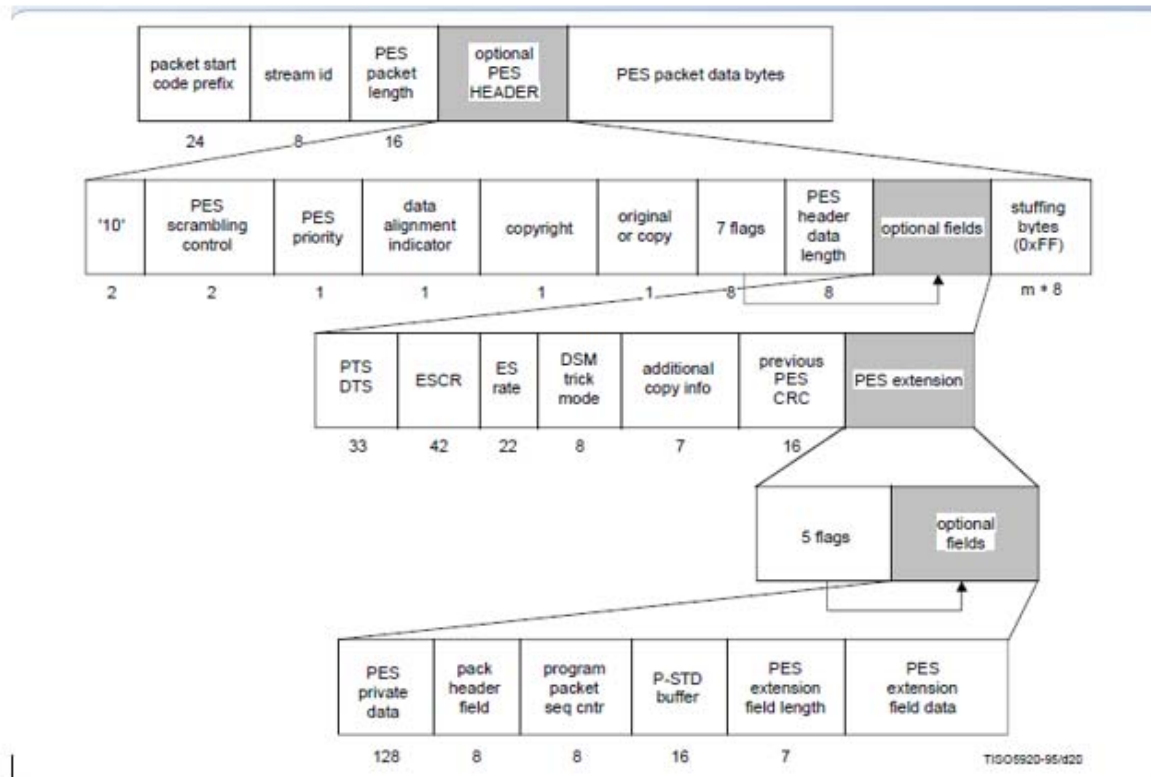


Fig 4.8 PES packet structure [56]

PES packet symantics [56] :

- Packet_start_code_prefix – (0x000001)
- Stream_id – describe the elementary stream type

stream_id ^o	Note ^o	stream coding ^o
1011 1100 ^o	1 ^o	program_stream_map ^o
1011 1101 ^o	2 ^o	private_stream_1 ^o
1011 1110 ^o	^o	padding_stream ^o
1011 1111 ^o	3 ^o	private_stream_2 ^o
110x xxxx ^o	^o	ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx ^o
1110 xxxx ^o	^o	ITU-T Rec. H.262 ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 or ITU-T Rec.H.264 ISO/IEC 14496-10 video stream number xxxx ^o
1111 0000 ^o	3 ^o	ECM_stream ^o
1111 0001 ^o	3 ^o	EMM_stream ^o
1111 0010 ^o	5 ^o	ITU-T Rec. H.222.0 ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6 DSMCC_stream ^o
1111 0011 ^o	2 ^o	ISO/IEC_13522_stream ^o
1111 0100 ^o	6 ^o	ITU-T Rec. H.222.1 type A ^o
1111 0101 ^o	6 ^o	ITU-T Rec. H.222.1 type B ^o
1111 0110 ^o	6 ^o	ITU-T Rec. H.222.1 type C ^o
1111 0111 ^o	6 ^o	ITU-T Rec. H.222.1 type D ^o
1111 1000 ^o	6 ^o	ITU-T Rec. H.222.1 type E ^o
1111 1001 ^o	7 ^o	ancillary_stream ^o
1111 1010 ^o	^o	ISO/IEC14496-1_SL-packetized_stream ^o
1111 1011 ^o	^o	ISO/IEC14496-1_FlexMux_stream ^o
1111.1100 ^o	^o	metadata stream ^o
1111.1101 ^o	8 ^o	extended_stream_id ^o
1111 1110 ^o	^o	reserved data stream ^o
1111 1111 ^o	4 ^o	program_stream_directory ^o

- Packet start code = packet_start_code_prefix + stream_id
- PES_scrambling_control – indicate the scrambling mode of the PES packet payload.

Value ^o	Description ^o
00 ^o	Not scrambled ^o
01 ^o	User-defined ^o
10 ^o	User-defined ^o
11 ^o	User-defined ^o

• Data_alignment_indicator – When set to a value of '1', it indicates that the PES packet header is immediately followed by the video syntax element or audio sync word indicated in the data_stream_alignment_descriptor.

• Copyright - When set to '1' it indicates that the material of the associated PES packet payload is protected by copyright.

- **Original_or_copy** - When set to '1' the contents of the associated PES packet payload is an original.
- **PTS (presentation time stamp)** - indicates the time of presentation, $tpn(k)$, in the system target decoder of a presentation unit k of elementary stream n .
 - $PTS(k) = ((\text{system_clock_frequency} \cdot tpn(k)) \text{ DIV } 300) \% 233$
 - where $tpn(k)$ is the presentation time of presentation unit $Pn(k)$.
- **DTS (decoding time stamp)** - It indicates the decoding time, $tdn(j)$, in the system target decoder of an access unit j of elementary stream n .
 - $DTS(j) = ((\text{system_clock_frequency} \cdot tdn(j)) \text{ DIV } 300) \% 233$
 - where $tdn(j)$ is the decoding time of access unit $An(j)$.
- **ES_rate** – specify the rate at which the system target decoder(STD) receives bytes of the PES packet in the case of a PES stream.
- **Trick_mode_control** – indicate which trick mode is applied to the associated video stream.
- **Additional_copy_info** – contain private data relating to copyright information.

4.5 Transport stream format

A transport stream packet is of fixed length of 188 bytes and it always begins with a synchronization byte of 0x47. The choice of this packet structure was motivated by few important factors. They are,

- The packets need to be large enough so that the overhead of the transport headers does not become a significant portion of the total data being carried [60].
- The packet size should not be too large that the probability of packet error becomes significant under the standard operating conditions [60].
- Another factor is the interoperability with the ATM packets as each MPEG-2 transport stream packet is transmitted in four ATM packets [60,61]

There are few factors which need to be considered while forming the transport stream packets [30]. They are,

- The total packet size should be of fixed length i.e. 188 bytes.
- Each packet can have data from only one PES.
- PES header should be the first byte of the transport packet payload.
- If the above constraints are not met, the PES packet is split and additional stuffing bytes are added.

The encapsulation process of TS packets from PES packets is shown in the Fig 4.9

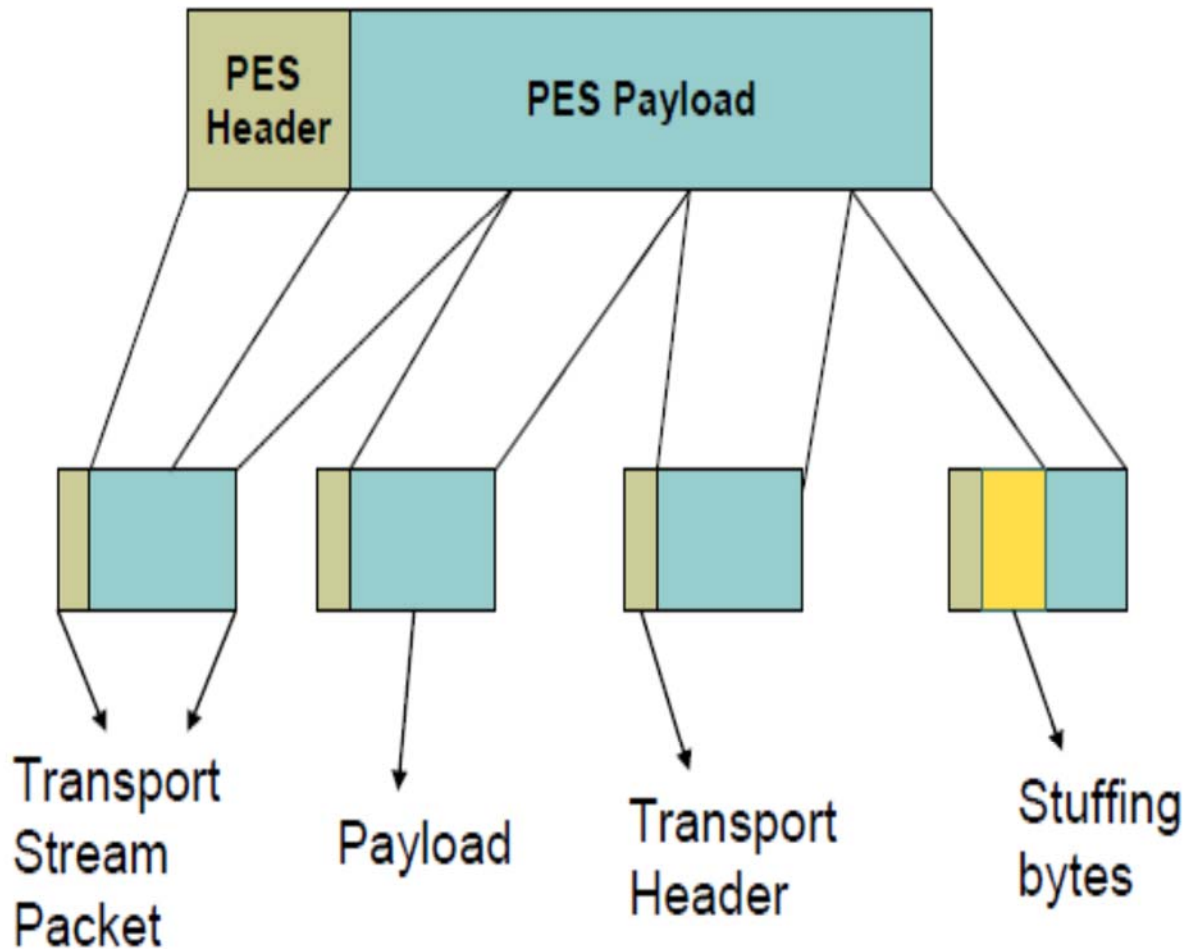


Fig 4.9 Encapsulation of TS packets from PES packets [61]

4.5.1 TS packet header

A TS packet consists of 3 byte (4 bytes if adaptation field is present) header with 185 bytes allocated for payload. If adaption field is present, then 184 bytes are allocated for the payload. The TS packet header description as adopted in MPEG-2 systems is shown in Table 4.2.

Syntax	Number of Bits
<pre> transport_packet(){ sync_byte transport_error_indicator payload_unit_start_indicator transport_priority PID transport_scrambling_control adaptation_field_control continuity_counter if(adaptation_field_control == '10' adaptation_field_control == '11'){ adaptation_field() } if(adaptation_field_control == '01' adaptation_field_control == '11') { for (i = 0; i < N; i++){ data_byte } } } </pre>	<p>8</p> <p>1</p> <p>1</p> <p>1</p> <p>13</p> <p>2</p> <p>2</p> <p>4</p> <p>8</p>

Table 4.2 TS packet header description as adopted in MPEG 2 systems [50]

The description of TS packet header is as follows:

- Sync byte: The value of sync byte is '0x47'. TS packet always starts with this byte.
- Payload unit start indicator: This bit helps in determining the start of PES packet data in the payload of a TS packet. If PUSI = '1', TS payload consists of start of PES packet. If PUSI = '0', TS payload consists of PES data of same PES.
- Adaptation field control: This bit helps in determining the last TS packet for given PES packet. If AFC='1', then it is the last TS packet of given PES packet. If AFC='0', then there are more TS packets following for same PES packet. If AFC='1' and there are exactly 185 bytes of PES data, then byte offset field is not set.
- PID (Packet Identifier): This is a 10 bit packet identifier value. This is used to uniquely identify the video or audio elementary streams. A TS packet with an unknown PID is discarded at the receiver.
- The particular PID value of '0x1024' is used to indicate a null packet. The null TS packet is a special TS packet designed to create a specific overall constant bit stream. Null TS packets are transmitted when there are no other TS packets available for transmission [50]. These packets are discarded when received at the de-multiplexer.
- Continuity counter: This is a 4 bit counter which is incremented by one every time the data from the same PES is encapsulated into a TS packet i.e. for each consecutive TS packet of the same PID. 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.

- Byte offset: 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 = 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.

Figure 4.10 shows an MPEG transport stream and a transport packet structure. This diagram shows that an MPEG-TS packet is fixed size of 188 bytes including a 4-byte header. The header contains various fields including an initial synchronization (time alignment) field, flow control bits, packet identifier (which PES stream is contained in the payload) and additional format and flow control bits.

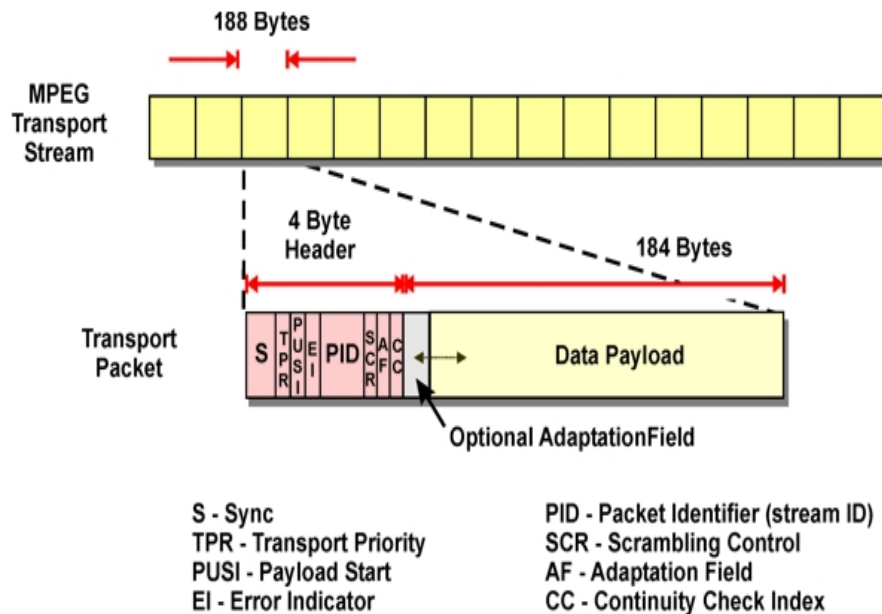


Fig 4.10 MPEG transport stream and standard structure of MPEG-TS packet [78]

4.6 Frame number as time stamp

The method adopted in this thesis uses the frame numbers as time stamps [28][30] for the synchronization of HEVC Video and HE-AAC V2 audio streams. HEVC video and HE-AAC V2 audio bit streams are arranged in frame wise format. This factor is used for synchronizing the audio and video elementary streams at the de-multiplexer for playback. The frame rate for the video stream is specified by the frames per second (fps) value. Since the frame rate is already known, the time of occurrence of a particular frame in the video sequence while playback can be calculated using the formula specified in (4-1).

$$\text{Playback time} = \text{Frame number}/\text{fps} \quad (4-1)$$

The AAC compression defines an AAC frame to contain 1024 samples. The same is applicable for HE-AAC V2. Although HE-AAC V2 standard supports different sampling frequencies, the

sampling rate is fixed while encoding the audio bit streams hence the frame duration also remains constant throughout a particular audio stream. Since the sampling frequency is already known, the time of occurrence of a particular audio frame can be calculated using the formula specified in (4.2).

$$\text{Playback time} = 1024 * (1/\text{sampling frequency}) * \text{frame number} \quad (4.2)$$

Thus by using (4.1) and (4.2) the playback time of the encoded video and audio sequences can be calculated using frame numbers as time stamps which means that given the frame number of one stream, the frame number of the other streams played at the same time during playback can be calculated. This helps in synchronizing the video and audio sequences during playback.

The time stamps are assigned in the last 2 bytes of the PES header, hence the maximum frame number that can be carried in this header is 65535. In the case of long video or audio elementary streams the frame number is rolled over to zero and this takes place simultaneously on both the video and audio frame numbers. Since the buffer size at the de-multiplexer is much smaller than the maximum allowed frame number, at no point of time there will be two frames in the buffer with the same time stamp.

4.7 Advantages of frame numbers over the existing method for time stamps

In the MPEG-2 systems standard the method used for audio-video synchronization is based on the presentation time stamps (PTS). The encoder attaches a PTS to video and audio frames which is a 33 bit value in cycles of a 90-kHz system time clock (STC) [50]. This STC clock is regenerated at the receiver and the clock samples are compared before presentation to achieve synchronization. For this to work effectively, the clocks at the multiplexer and de-multiplexer should be exactly synchronized. For achieving this, additional information known as program clock reference (PCR) which is the value of the STC at the encoder is periodically transmitted. The advantages of using frame numbers over the adopted method in MPEG-2 systems are as follows.

- Less complex and is suitable for software implementation.
- No synchronization problem due to clock jitters.
- No propagation of delay between audio and video elementary streams.
- Saves the extra overhead in the PES header bytes used for sending the PCR bytes.

4.8 Proposed multiplexing method

After the TS packets are formed, these packets are multiplexed to form the final transmission stream. The presentation time of audio and video packets is used as a criteria to multiplex the data. If more video TS packets are sent as compared to audio TS packets, then at the receiver there might be a situation when video buffer is full and is overflowing whereas audio buffer does not have enough data. This will prevent the demultiplexer from starting a playback and will lead to loss of data from the overflowing buffer.

Hence, counters for audio and video TS are maintained at the multiplexer end. The TS packet with lesser presentation time is transmitted first and the corresponding counter is incremented. The Video TS is transmitted first. This plays an important part in avoiding buffer overflow or underflow at the demultiplexer..

The playback time of each PES can be calculated since the frame duration is constant in both audio and video elementary streams. The elementary stream whose counter has the least timing value is always given preference in packet allocation. This method will make sure that at any point of time, the difference in the fullness of the buffers, in terms of playback time is less than the playback time of one TS packet. This is never more than the duration of a single frame and is typically in milliseconds.

However, the proposed method is not a dynamically changing method that can adopt to the varying frame sizes of the elementary streams. Also, this method requires us to have the full size of the files before beginning the multiplexing process. This might not be possible if encoding is done in real time.

4.9 summary

This chapter provides with the information of the standard MPEG -2 transport stream and how it is used for multiplexing video and audio bitstreams. Eventually, a method for multiplexing the TS packets is proposed that can prevent buffer overflow or underflow at the demultiplexer. The next chapter describes the de-multiplexing algorithm used and the method used to achieve audio-video synchronization.

CHAPTER – 5

Demultiplexing

5.1 Introduction

Demultiplexing is the inverse process of multiplexing, which involves the process of recovering elementary streams from the multiplexed Transport Stream. This is the initial step performed at the receiver end during the process of delivering a complete multimedia program to the end user. The flowchart of the demultiplexer is as shown in Figure 5.1.

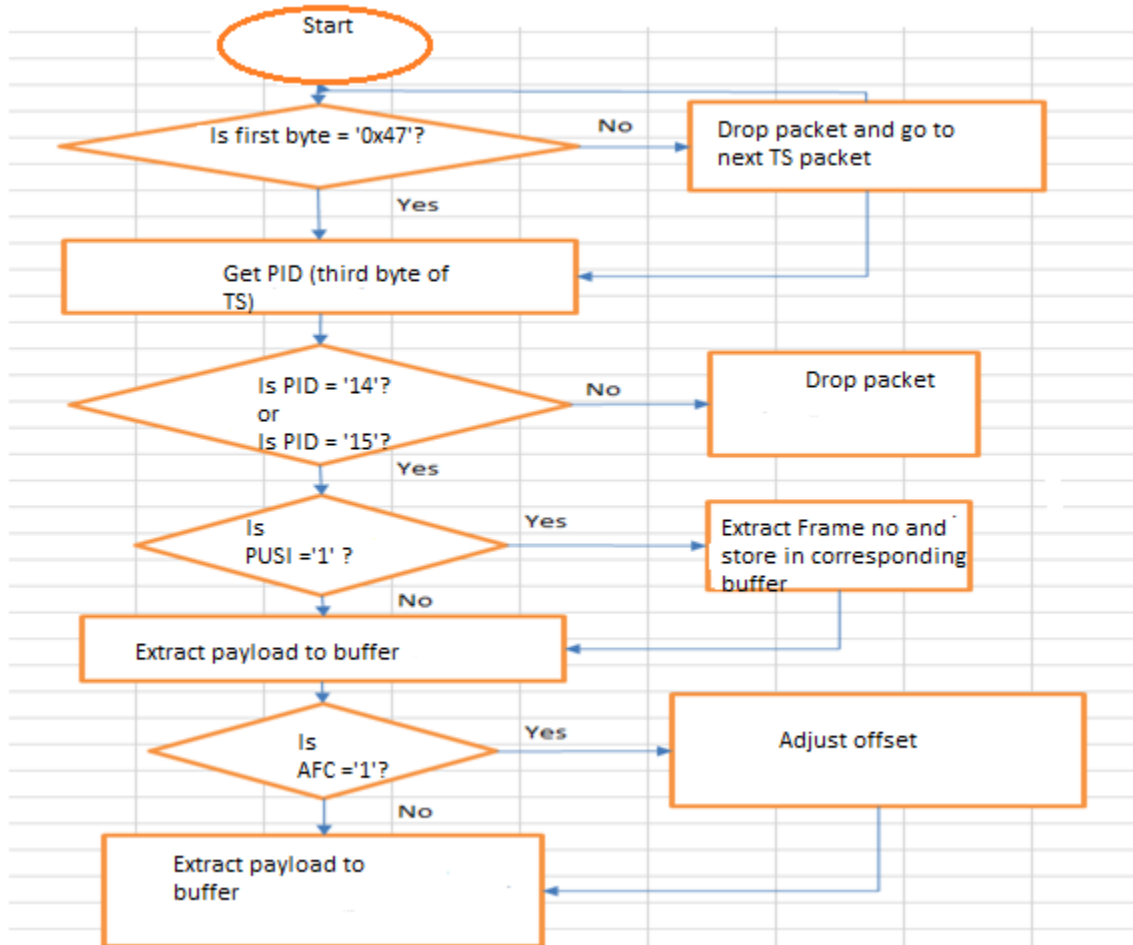


Fig 5.1 Flowchart of demultiplexer

Once TS packets are multiplexed, they are received at the demultiplexer end and checked for the sync byte (0x47), to check if the packet is valid 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 packet 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 '14' it is an audio packet else if '15' it is a video packet. If the packet has any PID value that is not relevant to the multimedia program that is being recovered, the packet is dropped and the next packet is analyzed. All the TS packets from other programs or null packets are eliminated at this stage. This is done to prevent using the resources on unwanted data. Once the packet has been identified to be a required one, further analysis of the packet is carried out.

Now the payload is prepared to be read into the appropriate buffer. 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 AFC and PUSI are set to '1', they indicate the start and end of TS packet corresponding to the same PES. It means that it is the only packet of a PES and the payload is extracted.
- 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.

This process is carried for the entire multiplexed stream to extract video and audio elementary streams. The frame number and location of the frame in the data buffer are stored in a separate buffer. This process is continued until one of the elementary stream buffers is full. In order to detect packet losses, 4 bit continuity counter value is continuously monitored for each PID separately, to check if the counter value increments in sequence. If not a packet loss is declared and the particular frame in the buffer, which is involved in the loss, is marked to be erroneous.

The buffer fullness is also continuously monitored at the de-multiplexer end to prevent any buffer overflow or underflow as it would result in out of sync problems. 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. 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.

$$\text{Video content playback time} = \text{Number of video frames} \times (1/\text{fps}) \dots\dots\dots (5.1)$$

$$\text{Audio content playback time} = \text{Number of audio frames} \times (1024/\text{sampling frequency}) \dots\dots (5.2)$$

5.2 Synchronization and Playback

Once the elementary stream buffer is full, the content is ready to be played back for the viewer. The audio bit stream format i.e., audio data transport stream (ADTS), enables us to begin decoding from any frame. However, the video bit stream does not have the same kind of sophistication. The decoding can start only from the anchor frames, which are the IDR frames. IDR frames are forced during the encoding process at regular intervals. Hence, the video buffer is first searched from the top to get the first occurring IDR frame. Once this is found, the timestamp or the frame number is obtained for that IDR frame. Then audio stream is aligned accordingly to achieve synchronization. This is done by calculating the audio frame number that would correspond to the IDR frame in terms of playback time. This is calculated as follows:

$$\text{Audio frame number} = \frac{\text{Video framenum} \times \text{sampling frequency}}{1024 \times \text{fps}} \dots\dots (5.3)$$

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. The maximum delay that is allowed by MPEG-2 standard is $\pm 40\text{ms}$. If the difference between audio and video results in a negative threshold, audio lags video else, in case of positive threshold, audio leads video [67]. 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 [66] and decoded from that point and played back using the VLC media player [65].

The results tabulated in Chapter 6 shows that visual delay is not perceptible and is almost consistent once synchronized. Thus, the demultiplexing 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 [74] and are used in this thesis. The elementary streams are in raw formats, with video stream in the YUV format and audio stream in the WAVE format. There are no standard video and corresponding audio test sequences, freely available. Hence these raw sequences were extracted from the existing AVI file format using ffmpeg [63] software. This is an open source software, which helps extract YUV format video and wave format audio from any AVI format.

The YUV video file is of very large size and it is encoded using x265 software [64]. The raw YUV file is used as an input to the x265 and output is a .hevc file which is compressed. The encoder setting used is main inter profile with GOP structure as IBBBP. The raw audio stream is encoded using open source software called 3gpp Enhanced aacplus encoder [27] and audio encoding bit rate was set at 16 kbps. .WAVE is used as an input to the aacplusenc and output is an .aac file which is compressed.

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. Sequences of length 35 seconds, 12 seconds and 18 seconds are used. A single program stream comprising of an audio and video elementary stream is implemented. The Table 6.1 provides the information of video frame rate, audio sampling frequency, compression ratio, bit rates and file size of the test sequences used.

The net transport stream bitrates for the sequences obtained are 267.2 kbps, 1095.08 kbps and 1093.44 kbps which can be easily accommodated in systems such as ATSC-M/H, which has an allocated bandwidth of 19.6 Mbps [19]. Since, I (Intra) only coding requires more number of bits when compared to inter coding, inter coding of pictures is used in most practical applications.

Details of test clips	Clip 1 Oscars	Clip 2 starwars2k	Clip 3 Starwars4k
Clip Duration (seconds)	35	12	18
Resolution	1280x720	2560x1440	3840x2160
Video frame rate (fps)	59.94	25	25
Audio sampling frequency(kHz)	44.1	44.1	44.1
YUV file size (kB)	2,83,0950	1,657,800	5,601,150
WAV file size (kB)	6,029	2,111	3,178
.HEVC file size(kB)	1,844	2,821	4,195
# of video frames	2097	307	461
.AAC file size (kB)	74	27	40
# of audio frames	752	262	396
Video compression ratio	1535.22 : 1	587.66 : 1	1335.20 : 1
Audio compression ratio	81.47 : 1	78.19 : 1	158.74 : 1
HEVC video bitrate (kbps)	52.69	235.08	233.06
HE-AAC V2 audio bitrate (kbps)	2.114	2.25	2.22
TS packets	11496	15780	23667
TS file size(kB)	9,352	13,141	19,682
TS bitrate(kbps)	267.2	1095.08	1093.44
Original AVI file size (kB)	6,365	24,194	86,806

Table 6.1 Inter coding: test clips characteristics

The results in Table 6.1 clearly show the compression achieved, using HEVC video codec and HE-AAC V2 audio codec. The results show that compression ratio achieved by HE-AAC V2 is in the order of 45 to 75 which is at least three times better than that achieved by just core AAC. The .avi file is played using the VLC media player by VideoLAN [65]. Then the multiplexed output is de-multiplexed into video and audio buffers using the de-multiplexing algorithm. Tables 6.2, 6.3 and 6.4 tabulate de-multiplexed results.

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?
36	4	3	0.0667	0.0697	2.9	No
500	48	35	0.8008	0.8127	11.9	No
2500	214	154	3.5702	3.5759	5.6	No
5900	537	386	8.959	8.9629	3.9	No
8700	782	562	13.0464	13.0496	3.2	No

Table 6.2 Clip 1 : Output of de-multiplexer (Oscars)

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?
300	6	11	0.2400	0.2554	15.4	No
500	8	14	0.3200	0.3251	5.1	No
2100	24	42	0.9600	0.9752	15.2	No
5500	64	111	2.5600	2.5774	17.4	No
9000	102	176	4.0800	4.0867	6.7	No

Table 6.3 Clip 2 : Output of de-multiplexer (starwars2k)

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?
9	3	6	0.1200	0.1393	19.3	No
700	15	26	0.6000	0.6037	3.7	No
3300	45	78	1.8000	1.8112	11.2	No
6700	92	159	3.6800	3.692	12.0	No
15000	164	283	6.5600	6.5172	11.2	No

Table 6.4 Clip 3 : Output of de-multiplexer (starwars4k)

The data can be demultiplexed 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 19.3 ms, while the delay is almost consistent otherwise. This is well below MPEG-2 threshold of 40 ms . 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 [66], a freeware is used to put the data to matroska multimedia container format (.mkv). Since, mkvmerge tool ignores the fact that HEVC has bidirectional predicted (B) frames some jitters can

be observed in the reconstructed video. Also, due to non-availability of other freeware that supports to check for inter sequence muxing, mkvmerge has only been used in this thesis.

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 [2][3].

The Tables 6.5, 6.6 and 6.7 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.

# 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)
25	9	64	2	0.4171	0.4180
40	17	69	2	0.6673	0.6966
80	29	173	4	1.3347	1.3468
110	40	235	5	1.8352	1.8576
158	57	329	6	2.6351	2.6471
236	85	445	9	3.9373	3.9474

Table 6.5 Clip1: Check for buffer fullness and video/audio content playback time (Oscars)

# 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)
25	22	454	3	1.0000	1.0217
59	51	963	6	2.3600	2.3684
77	67	1238	7	3.0800	3.1115
99	86	1585	9	3.9600	3.9938
132	114	2123	12	5.2800	5.2941
157	136	2503	14	6.2800	6.3158

Table 6.6 Clip 2 : Check for buffer fullness and video/audio content playback time (starwars2k)

# 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)
25	22	350	3	1.0000	1.0217
51	44	650	5	2.0400	2.0434
78	68	1010	7	3.1200	3.1579
135	117	2212	12	5.4000	5.4335
159	137	2640	14	6.3600	6.3623
199	172	3432	18	7.9600	7.9877

Table 6.7 Clip3: Check for buffer fullness and video/audio content playback time (starwars4k)

Tables 6.5, 6.6 and 6.7 shows the values of video buffer and the corresponding audio buffer size at that moment and the playback times of both audio and video contents of buffer. It can be observed the content playback times vary only by about 37.9 ms. This means that when a video buffer is full (for any size of video buffer) almost all the corresponding audio content is present in the audio buffer.

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 by adopting two layers of packetization namely the packetized elementary stream layer and transport stream layer for multiplexing the video and audio elementary streams. Time stamps in the form of frame numbers aid in achieving synchronization. The use of HEVC video bit-stream and HE-AAC V2 audio bit-stream helps to deliver high quality video and audio at a reasonably low bit rate.

The proposed multiplexing procedure enables the user to start demultiplexing from any TS packet and achieve synchronized playback for any program. From the results tabulated in Tables 6.2 to 6.7, 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 37.9 ms between audio and video. Also, during decoding the audio-video synchronization was achieved with a maximum delay of about 18 ms. Thus, the implemented algorithm effectively multiplexes, de-multiplexes and achieves synchronization during the playback.

There is also provision for error detection and correction after receiving the packets. These are absolutely essential for video broadcasting applications. Thus the proposed method meets all the basic requirements to transmit a high quality multimedia program.

6.3 Future Research

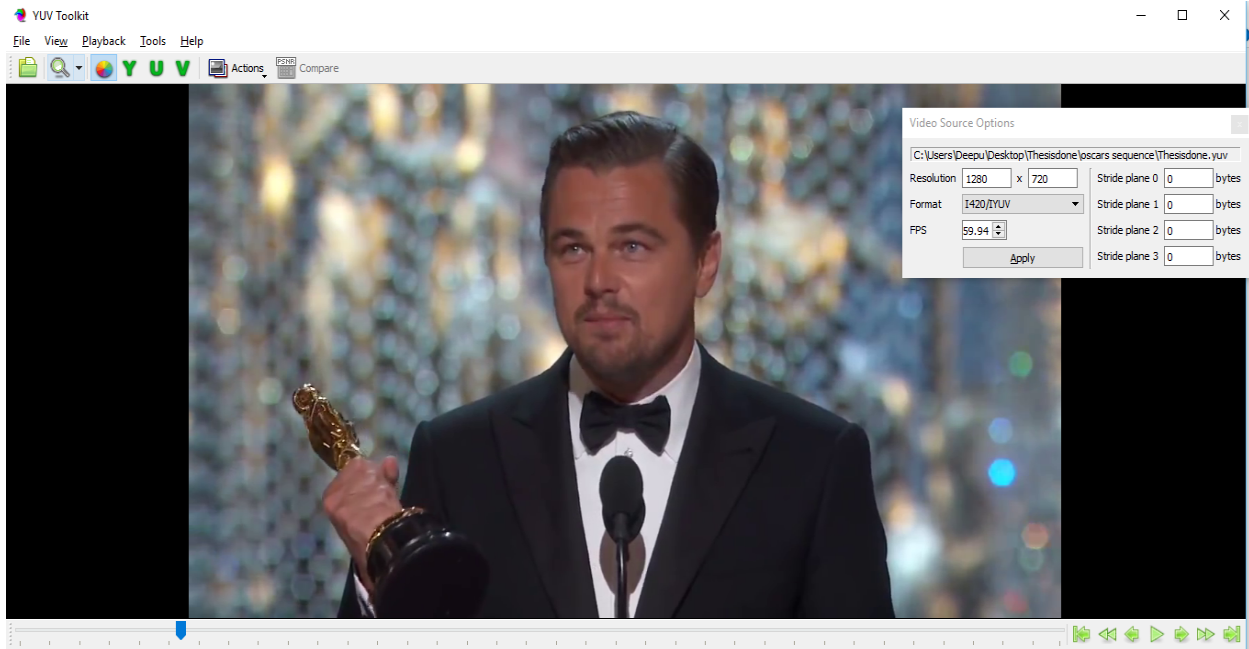
This thesis has focused on implementing one program elementary stream i.e. one video and one audio only. With minor additions to this implementation, multiple program elementary streams can be accommodated to broadcast data simultaneously in multiple channels. Also, subtitles can be incorporated while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

Some robust error correction codes can be integrated into the transport packets, to make them more suitable for applications such as video conferencing and broadcasting where the packets are prone to be lost.

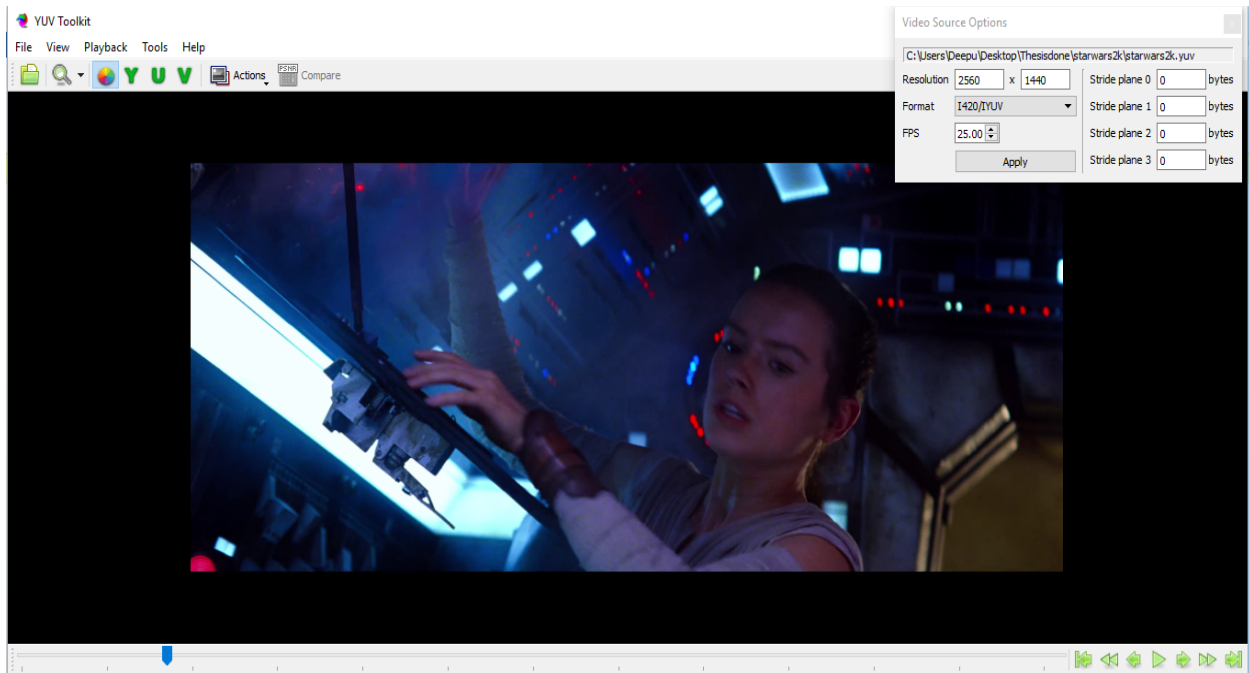
This scheme can be implemented for AV1, VP10 and DAALA video codecs and can also be extended to 8K sequences.

APPENDIX A

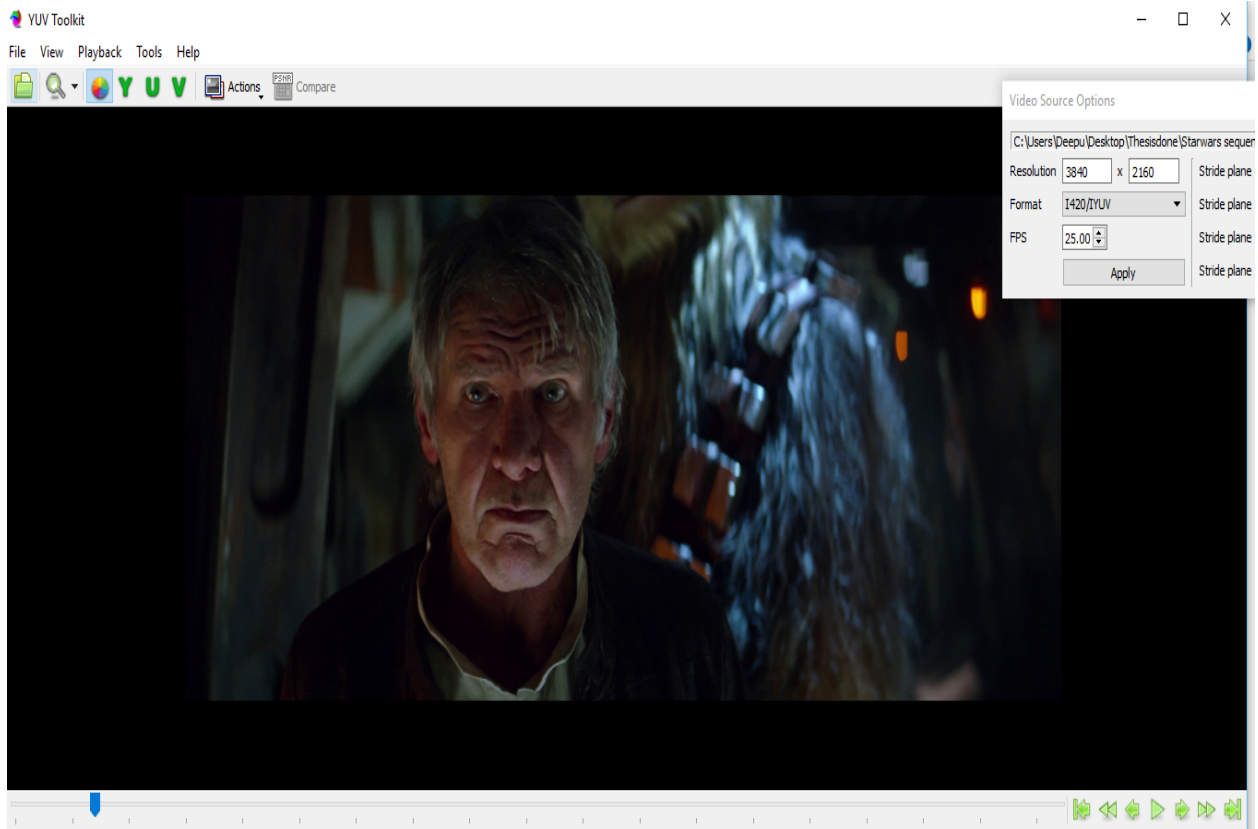
TEST SEQUENCES (can be accessed from www.youtube.com)



Clip 1 : Oscars.avi , 1920x1080 , 2097 frames, 59.94 fps [74]



Clip 2 : Starwars2k.avi, 2560x1440, 307 frames, 25 fps [74]



Clip 3 : starwars4k.avi , 3840x2160, 461 frames, 25 fps [74]

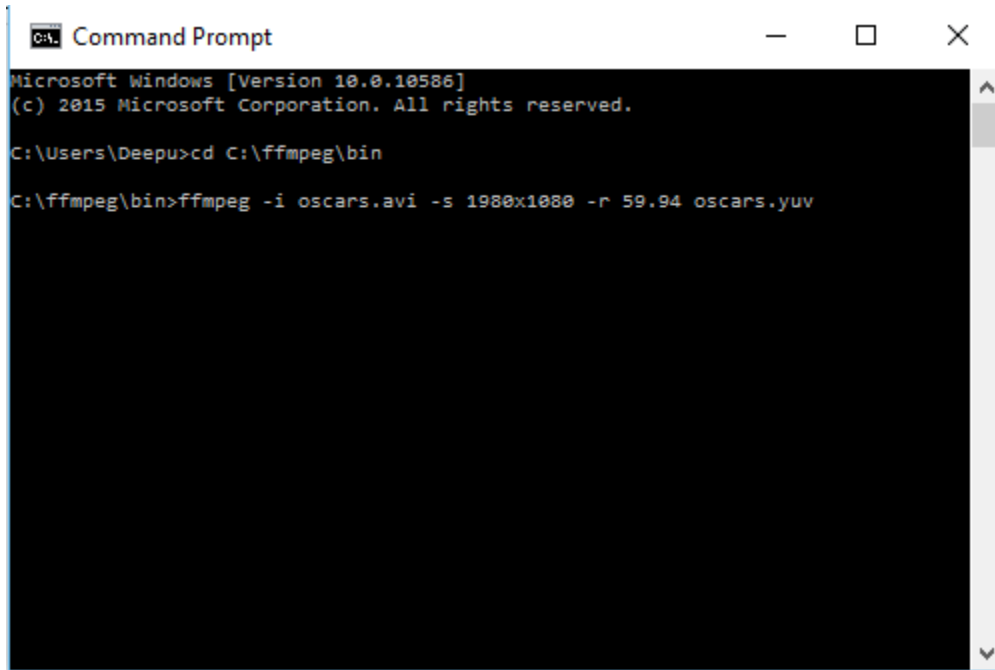
APPENDIX B

Test Platform:

Processor :	Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz
Installed Memory(RAM):	8.00 GB
System Type:	64-bit operating system, x-64 based processor

APPENDIX C

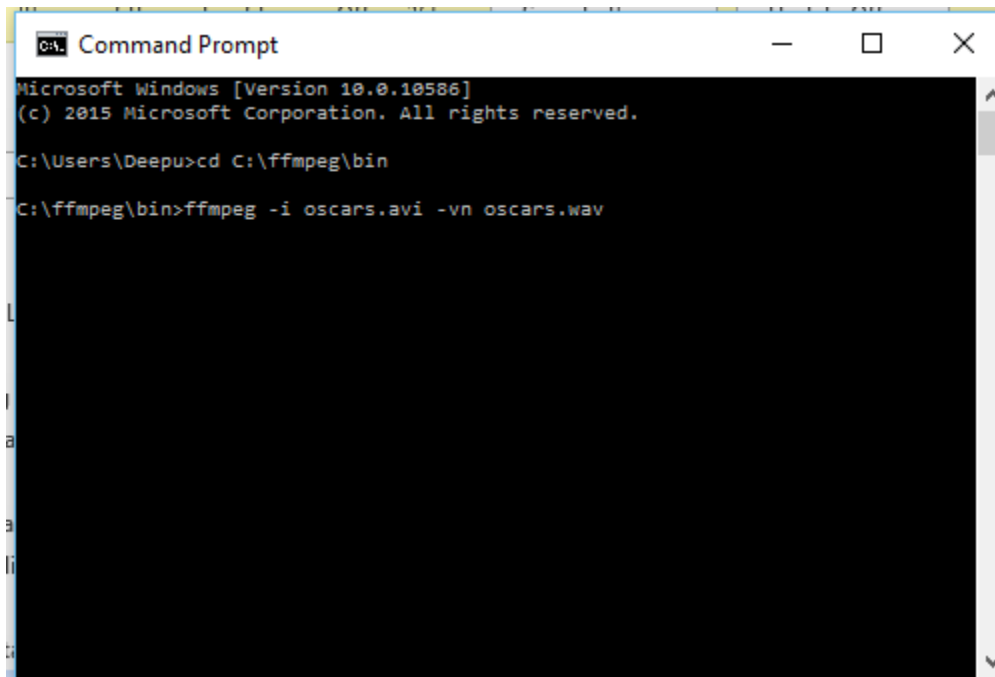
Using ffmpeg to split avi into audio and video files:



```
CA: Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Deepu>cd C:\ffmpeg\bin

C:\ffmpeg\bin>ffmpeg -i oscars.avi -s 1980x1080 -r 59.94 oscars.yuv
```



```
CA: Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Deepu>cd C:\ffmpeg\bin

C:\ffmpeg\bin>ffmpeg -i oscars.avi -vn oscars.wav
```

References

1. G. J. Sullivan et al, “Overview of the High Efficiency Video Coding (HEVC) Standard” , IEEE Trans. on Circuits and Systems for Video Technology, vol 22, pp. 1649 – 1668, Dec 2012.
2. MPEG-4: ISO/IEC JTC1/SC29 14496-3: Information technology – coding of audio-visual objects – part3: Audio, amendment 4: Audio lossless coding (ALS), new audio profiles and ASAC extensions.
3. MPEG-2: ISO/IEC JTC1/SC29 13818-7, advanced audio coding, AAC. International Standard IS WG11, 1997
4. JVT Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264-ISO/IEC 14496-10 AVC), March 2003, JVT-G050 available on http://ip.hhi.de/imagecom_G1/assets/pdfs/JVT-G050.pdf
5. G. J. Sullivan et al, “Standardized Extensions of High Efficiency Video Coding (HEVC)”, IEEE Journal of selected topics in Signal Processing, Vol. 7, No. 6, pp. 1001-1016, Dec. 2013.
6. D. K. Kwon and M. Budagavi , " Combined scalable and mutiview extension of High Efficiency Video Coding (HEVC) " , IEEE Picture Coding Symposium , pp. 414 - 417 , Dec . 2013 .
7. HEVC open source software (encoder/decoder) :
https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-9.1-dev/
8. N. Ling, “High efficiency video coding and its 3D extension: A research perspective,” Keynote Speech, ICIEA, pp 2150-2155, Singapore, July 2012
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6361087>
9. HEVC white paper - <http://www.ateme.com/an-introduction-to-uhdtv-and-hevc>
10. I. E. G. Richardson, “Video Codec Design: Developing Image and Video Compression Systems”, Wiley, 2002
11. K. R. Rao, D. N. Kim and J. J. Hwang,
“Video Coding Standards : AVS China, H.264/MPEG-4 Part10, HEVC, VP6, DIRAC and VC-1”, Springer, 2014.
12. HEVC tutorial by I.E.G. Richardson: <http://www.vcodex.com/h265.html>
13. C. Fogg, “Suggested figures for the HEVC specification”, ITU-T / ISO-IEC Document: JCTVC J0292r1, July 2012.
14. N. Ahmed , T. Natarajan and K. R. Rao, “Discrete Cosine Transform”, IEEE Trans. on Computers, Vol. C-23, pp. 90-93, Jan. 1974.
15. D. Marpe, H. Schwarz, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,” IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, pp. 620–636, July 2003
16. J. Nightingale, Q. Wang and C. Grecos, “HEVStream: A framework for streaming and

- evaluation of High Efficiency Video Coding (HEVC) content in loss-prone networks” , IEEE Trans. Consumer Electronics, vol.59, pp.404-412, May 2012.
17. 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.
 18. Access to HM 16.4 Software Manual:
https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.4+SCM-4.0rc1/doc/software-manual.pdf
 19. ATSC-M/H. Link : <http://www.atsc.org/cms/>
 20. ISDB website. Link: <http://www.dibeg.org/>
 21. 3gpp website. Link: <http://www.3gpp.org/>
 22. World DMB: link: <http://www.worldddab.org/>
 23. MPEG-4 HE-AAC v2 — audio coding for today's digital media world , article in the EBU technical review (01/2006) giving explanations on HE-AAC. Link: http://tech.ebu.ch/docs/techreview/trev_305-moser.pdf
 24. MPEG-2: ISO/IEC 13818-1 Information technology—generic coding of moving pictures and associated audio—Part 1: Systems, ISO/IEC: 2005.
 25. 3GPP TS 26.401: General Audio Codec audio processing functions; Enhanced aacPlus General Audio Codec; 2009.
 26. MPEG-4: ISO/IEC JTC1/SC29 14496-14 : Information technology — coding of audio-visual objects — Part 14 :MP4 file format, 2003.
 27. 3GPP Enhanced aacPlus reference software. Link: <http://www.3gpp.org/ftp/>
 28. H. Murugan, “Multiplexing H.264 video with AAC audio bit streams, demultiplexing and achieving lip sync during playback”, M.S. Thesis, EE Dept, UTA, Arlington, TX, May 2007.
 29. A. Urs, “Multiplexing of Dirac Video with AAC Audio bit-stream, demultiplexing and achieving lip synchronization”, M.S. Thesis, EE Dept, UTA, Arlington, TX, May 2011.
 30. S. Sridhar, “Multiplexing/ demultiplexing AVS-china video with AAC audio bit-streams, achieving lip sync”, M.S. Thesis, EE Dept, UTA, Arlington, TX, May 2010.
 31. N. Siddaraju, “Multiplexing/ demultiplexing H.264 video with HE-AAC audio bit-streams, achieving lip sync”, M.S. Thesis, EE Dept, UTA, Arlington, TX, 2010.
 32. M. Warriar, “Multiplexing/Demultiplexing of main profile of HEVC/H.265 video stream with AAC Audio bit-stream and achieving lip synchronization”, M.S. Thesis, EE Dept, UTA, Arlington, TX, May 2014.
 33. J. Herre and H. Purnhagen, “General audio coding,” in The MPEG-4 Book (Prentice Hall IMSC Multimedia Series), F. Pereira and T.Ebrahimi, Eds. Englewood Cliffs, NJ: Prentice-Hall, 2002.

Mentioned References [28] to [32] can be accessed from

http://www.uta.edu/faculty/krrao/dip/Courses/EE5359/index_tem.html

34. M. Wien, "High Efficiency Video Coding : Coding Tools and Specification" , Springer , 2014.
35. M Modi, "Audio compression gets better and more complex",
link: <http://www.eetimes.com/discussion/other/4025543/Audio-compression-gets-better-andmore-complex>
36. **HOW TO ACCESS JCT-VC DOCUMENTS** - JCT-VC DOCUMENTS can be found in JCT-VC document management system <http://phenix.int-evry.fr/jct>

All JCT-VC documents can be accessed. [online].

http://phenix.int-evry.fr/jct/doc_end_user/current_meeting.php?id_meeting=154&type_order=&sql_type=document_number

37. <http://media.xiph.org/video/derf/>
38. 2) <http://trace.eas.asu.edu/yuv/>
39. 3) <http://media.xiph.org/>
40. 4) <http://www.cipr.rpi.edu/resource/sequences/>
41. 5) [HTTP://BASAK0ZTAS.NET](http://BASAK0ZTAS.NET)
42. 6) www.elementaltechnologies.com – 4K Video Sequences
4K (3840x2160) UHD video test sequences
43. Multimedia communications with SVC, HEVC, and HEVC <http://r2d2n3po.istory.com/50>
44. Elemental 4K Test Clips:
[online] Available: <http://www.elementaltechnologies.com/resources/4k-test-sequences> ,
accessed Aug. 1, 2014
45. Harmonic 4K Test Clips:
[online] Available: <http://www.harmonicinc.com/resources/videos/4k-video-clip-center>,
accessed Aug. 1, 2014.

References [37] to [45] have links to download video test sequences

46. M. Bosi and R.E. Goldberg, "Introduction to digital audio coding standards", Norwell, MA: Kluwer, 2002. (reviewed in EURASIP Newsletter, vol.15, pp.7-8, March 2004).
47. T. Ogunfunmi and M. Narasimha, " Principles of speech coding", Boca Raton, FL: CRC Press, 2010.
48. D. Flynn, J. Sole and T. Suzuki, "High efficiency video coding (HEVC) range extensions text specification", Draft 4, JCT-VC. Retrieved 2013-08-07.
49. W. Zhu et al, "Screen content coding based on HEVC framework", IEEE Trans. Multimedia , vol.16, pp.1316-1326 Aug. 2014 (several papers related to MRC) MRC: mixed raster coding.

50. B. J. Lechner et al, "The ATSC transport layer, including program and system information protocol (PSIP)", Proceedings of the IEEE, vol. 94, no. 1, pp. 77-101, Jan. 2006.
51. G. A. Davidson et al, "ATSC video and audio coding", Proceedings of the IEEE, vol. 94, no. 1, pp. 60-76, Jan. 2006.
52. Special issue on Global Digital Television: Technology and Emerging Services, Proceedings of the IEEE, vol. 94, pp. 5-7, Jan. 2006.
53. J. Stott, "Design technique for multiplexing asynchronous digital video and audio signals", IEEE Trans. on communications, vol. 26, no. 5, pp. 601-610, May 1978.
54. Information Technology – Generic coding of moving pictures and associated audio- Part 1: Systems, ISO/IEC 13818-1:2005, International Telecommunications Union.
55. P. V. Rangan, "Lecture Notes 8,9", CSE 126 Multimedia Systems, Spring 2003.
Link : <https://cseweb.ucsd.edu/classes/sp03/cse126/lecture/lecture8.pdf>
56. K. kim, "MPEG-2 ES/PES/TS/PSI", PPT, MEDIA LAB, Kyung-Hee University.
Link : http://cmm.khu.ac.kr/korean/files/02.mpeg2ts1_es_pes_ps_ts_psi.pdf
- 57. Digital broadcast technologies, link:**
<http://www.ciscopress.com/articles/article.asp?p=106971&seqNum=7>
- 58. MPEG channel Multiplexing, link :** <http://www.althosbooks.com/intomp.html>
- 59. MPEG-2 systems layer, link :**
https://www.uic.edu/classes/ece/ece434/chapter_file/chapter7.htm#_Toc498451653
60. X. Chen, "Transporting compressed digital video", Kluwer, 2002.
61. W. Zhu, "End-to-End modeling and simulation of MPEG-2 transport streams over ATM networks with jitter", IEEE Trans. on circuits and systems on video technology, vol. 8, no. 1, pp. 9-12, Feb. 1998.
62. P. V. Rangan, S. S. Kumar, and S. Rajan, "Continuity and Synchronization in MPEG," IEEE Journal on Selected Areas in Communications, Vol. 14, pp. 52-60, Jan. 1996.
63. Ffmpeg software and official website.
Link: <http://ffmpeg.mplayerhq.hu/>
64. x265 software and official website.
Link : <http://x265.ru/en/builds/>
65. VLC software and source code.

Link : www.videolan.org

66. Mkvmerge software: <http://www.bunkus.org/videotools/mkvtoolnix/downloads.html>
67. G. Blakowski et al, “A media synchronization survey: Reference model, specification and case studies”, IEEE Journal on Selected Areas in Communications, vol.14, no.1, pp. 5 – 35, January 1996.
68. Information Technology – Generic coding of moving pictures and associated audio: Systems, International Standard 13818-1, ISO/IEC JTC1/SC29/WG11 N0801, 1994.
69. P.A. Sarginson, “MPEG-2: Overview of systems layer”, BBC RD 1996/2.
70. Cisco Visual Networking Index - <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>
71. Information Technology – Generic coding of moving pictures and associated audio information -- Part 7: Advanced Audio Coding (AAC)
72. Details about AAC Encoder :
http://www.etsi.org/deliver/etsi_ts/126400_126499/126403/13.00.00_60/ts_126403v130000p.pdf
73. HEVC white paper-Ateme: <http://www.ateme.com/an-introduction-to-uhdtv-and-hevc>
74. Test sequences downloaded from youtube : www.youtube.com
75. Access the website <http://www.uta.edu/faculty/krrao/dip/Courses/EE5359/>
76. I.E.G. Richardson, “Coding video: A practical guide to HEVC and beyond”, Wiley, 11 May 2015.
77. V. Sze, M. Budagavi and G. J. Sullivan (Editors) “High Efficiency Video Coding (HEVC) – Algorithms and Architectures”, Springer, 2014.
78. “Introduction to MPEG”, excerpted from IPTV basics, Lawrence Harte.
79. J. Xu, R. Joshi and R. A. Cohen, “Overview of the Emerging HEVC Screen Content Coding Extension,” IEEE Trans. CSVT. vol. 26, pp.50-62, Jan. 2016.
80. D. Flynn et al, “Overview of the Range Extensions for the HEVC Standard: Tools, Profiles and Performance,” IEEE Trans. CSVT. vol. 26, pp.4-19, Jan. 2016.
81. G. Tech et al, “Overview of the Multiview and 3D extensions of high efficiency video coding”, IEEE Trans. CSVT. vol. 26, pp.35-49, Jan. 2016.

82. S. Sridhar and K.R. Rao, " Multiplexing and Demultiplexing of AVS China video with AAC audio", TELSIKS 2011, pp. 84-91, Nis, Serbia, 5-8 Oct. 2011.
83. A. Urs and K.R. Rao, "Multiplexing/Demultiplexing Dirac video with AAC audio bit stream", TELSIKS 2011, pp. 80-83, Nis, Serbia, 5-8 Oct. 2011.
84. N. Siddaraju and K.R. Rao, " Multiplexing the elementary streams of H.264 video and MPEG4 HE AAC v2 audio, demultiplexing and achieving lip synchronization", American Journal of signal processing, Vol.2, No.3, pp. 1051 – 1054, June 2012.
85. H. Murugan and K.R. Rao, "Multiplexing H.264 video with AAC audio bit streams, demultiplexing and achieving lip sync", ICEAST 2007, Bangkok, Thailand, 21-23 Nov. 2007.

BIOGRAPHICAL INFORMATION

Deepika Sreenivasulu Pagala graduated with the Bachelor of Engineering degree in Electrical and Electronics Engineering from Bangalore Institute of Technology, Bangalore, affiliated to the Visvesvaraya Technological University, Karnataka, India in May 2012. She has remarkable achievements during her pursuit of Bachelors degree and a testimony to the fact was she being one of the University toppers during various semesters.

She joined IBM India Pvt. Ltd as an Application Developer and worked for two years. She pursued her master's degree at the University of Texas at Arlington in Electrical Engineering from August 2014 to August 2016.

She was a member of the multimedia processing research group guided by Dr. K. R. Rao. Her areas of interest include research and development in image processing, Video coding and Communication Theory. She was awarded with Electrical Engineering Scholarship during her course of study at University of Texas at Arlington. She worked as an intern and is currently employed with Realtime Data LLC., Plano, TX.