American Journal of Signal Processing 2012, 2(3): 52-59
DOI: 10.5923/j.ajsp.20120203.03

Multiplexing the Elementary Streams of H.264 Video and MPEG4 HE AAC v2 Audio, De-multiplexing and Achieving Lip Synchronization

Naveen Siddaraju, K.R. Rao

Electrical Engineering Department, University of Texas at Arlington, Arlington, TX 76019, USA

Abstract Television broadcasting applications such as ATSC-M/H, DVB[16] require that the encoded audio and video streams to be transmitted across a network in a single transport stream containing fixed sized data packets that can be easily recognized and decoded at the receiver. MPEG2 part1 specifies two layers of packetization to achieve a transport stream suitable for digital transmission. In a broadcasting system, multiplexing is a process in which two or more elementary streams are converted into a single transport stream ensuring synchronous playback of the elementary streams and proper buffer behavior at the decoder. This paper presents a scheme to multiplex the elementary streams of H.264 video and HE AAC v2 audio using the MPEG2 systems specifications[4], then de-multiplex the transport stream and playback the decoded elementary streams with lip synchronization or audio-video synchronization. This paper briefly introduces the MPEG2 systems, two layers of packetization namely program elementary stream (PES) and transport stream (TS). It also introduces the concept of timestamps. The proposed multiplexing and de-multiplexing algorithms followed to achieve synchronization are explained. It is shown that during decoding the audio-video synchronization is achieved with a maximum skew of 13ms.

Keywords H.264, HEAACv2, multiplexing, MPEG2 systems, PES, TS

1. Introduction

Mobile broadcast systems are becoming increasingly popular as cellular phones and highly efficient digital video compression techniques merge to enable digital TV and multimedia reception on the move. Mobile television broadcast systems like DVB-H (digital video broadcast-handheld)[16] and ATSC-M/H (advanced television systems committee- mobile/handheld)[17,18,21] have relatively small bandwidth allocation and the processing power at the target device (decoder) also varies. Hence, the choice of the compression standards used plays an important role. H.264[1,5,6] and HEAACv2[2,7,13] are the codecs used in the proposed method for the video and audio respectively.

H.264[5,48,51] is the latest and the most advanced video codec available today. It was jointly developed by the VCEG (video coding experts group) of ITU-T (international telecommunication union) and the MPEG (moving pictures experts group) of ISO/IEC (international standards organization). This standard achieves much greater compression than its predecessors like MPEG-2 video[37], MPEG4 part2 visual[38] etc. But the higher coding efficiency comes at the cost of increased complexity. The H.264 has been adopted as the video standard for many applications around the world including ATSC[21]. H.264 covers only video coding and is not of much use unless the video is accompanied by audio. Hence it is relevant and practical to encode/decode and multiplex/demultiplex both video and audio for replay at the receiver.

HEAACv2[49,50] or 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, mobile broadcasting etc. HE AAC v2 has been proven to be the most efficient audio compression tool available today. It comes with a fully featured toolset which enables coding in mono, stereo and multichannel modes (up to 48 channels). HE-AACv2[7] is the adopted standard for ATSC-M/H and many other systems around the world.

The encoded bit streams or elementary streams of H.264 and HEAACv2 are arranged as a sequence of access units. An access unit is a coded representation of a frame. Since each frame is coded differently the size of each access unit also varies. In order to transmit a multimedia content (audio and video) across a channel, the two streams have to be converted in to a single stream of fixed sized packets. For this the elementary streams have to undergo two layers of packetization (Fig. 1). The first layer of packetization yields

* Corresponding author:
rao@uta.edu (K. R. Rao)
Published online at http://journal.sapub.org/ajsp

Packetized Elementary Stream (PES) and the second layer of packetization where the actual multiplexing takes place results in a stream of fixed sized packets called as Transport Stream (TS). These TS packets are what are actually transmitted across the network using broadcast techniques such as those used in ATSC and DVB[16].

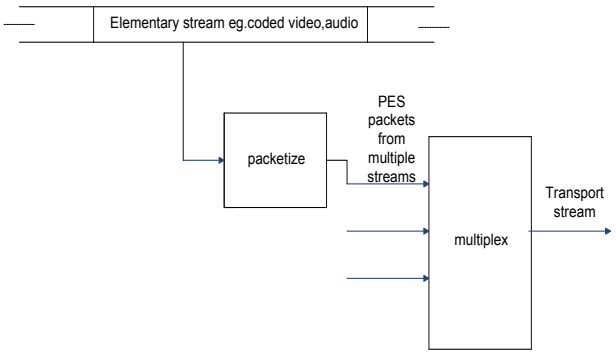

Figure 1. MPEG2 two layers of packetization[22]

1.1. Packetized elementary streams (PES)

PES packets are obtained after the first layer of packetization of coded audio and coded video data. This packetization process is carried out by sequentially separating out the audio and video elementary streams into access units. Hence each PES packet is an encapsulation of one frame of coded data. Each PES packet contains a packet header and the payload data from only one particular stream. PES header contains information which can distinguish between audio and video PES packets. Since the number of bits used to represent a frame in the bit stream varies (for both audio and video) the size of the PES packets also varies. Figure 2 shows how the elementary stream is converted into PES stream.

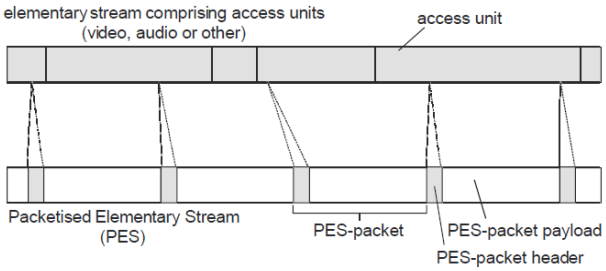

Figure 2. Conversion of an elementary stream into PES packets[29]

The PES header format used is shown in table 1. The PES header starts with a 3 byte packet start code prefix which is always "0x000001" followed by 1 byte stream id. Stream id is used to uniquely identify a particular stream. Stream id along with start code prefix is known as start code (4 bytes). PES packet length may vary and go up to 65536 bytes. In case of longer elementary stream, the packet length may be set as unbound i.e. 0, only in the case of video stream. The next two bytes in the header is the time stamp field, which contains the playback time information. In the proposed method, frame number is used to calculate the playback time, which is explained next.

2. Time Stamp

Time stamps indicate where a particular access unit belongs in time. Audio-video synchronization is obtained by incorporating time stamps into the headers in both video and audio PES packets.

Traditionally to enable the decoder to maintain synchronization between audio track and video frames, a 33 bit encoder clock sample called Program Clock Reference (PCR) is transmitted in the adaptation field of the TS packet from time to time (every 100 ms). This along with the presentation time stamp (PTS) field that resides in the PES packet layer of the transport stream is used to synchronize the audio and video elementary streams.

The proposed method uses the frame numbers of both audio and video as time stamps to synchronize the streams. As explained earlier both H.264 and HE AAC v2 bit streams are organized into access units i.e. frames separated by their respective sync sequence. A particular video sequence will have a fixed frame rate during playback which is specified by frames per second (fps). So assuming that the decoder has a prior knowledge about the fps of the video sequence, the presentation time (PT) or the playback time of a particular video frame can be calculated using (1).

$$Video\ PT = \frac{frame\ \ number}{fps} \tag{1}$$

The AAC compression standard defines each audio frame to contain 1024 samples per channel. This is true for HE AAC v2[2,3,7] as well. The sampling frequency of the audio stream can be extracted from the sampling frequency index field of the ADTS header. The sampling frequency remains the same for a particular audio stream. Since both samples per frame and sampling frequency are fixed, the audio frame rate also remains constant throughout a particular audio stream. Hence the presentation time (PT) of a particular audio frame (assuming stereo) can be calculated as follows:

$$Audio\ PT = \frac{1024\ X\ frame\ \ number}{audio\ \ sampling\ \ \ frequency} \tag{2}$$

The same expression can be expanded for multi channel audio streams, just by multiplying the number of channels.

Table 1. PES header format[4]

Name	Size (in Bytes)	Description
Packet start code prefix	3	0x000001
Stream id	1	Unique ID to distinguish between audio and video PES packet Examples: Audio streams (0xC0-0xDF), Video streams (0xE0-0xEF)[3]
		Note: the above 4 bytes together are known as start code.
PES Packet length	2	The PES packet can be of any length. A value of zero for the PES packet length can be used only when the PES packet payload is a video elementary stream
Time Stamp	2	frame number

Once the presentation time of one stream is calculated, the frame number of the second stream that has to be played at that particular time can calculated. This approach is used at the decoder to achieve the audio-video synchronization or lip synchronization; this is explained in detail later on.

Using frame numbers as time stamps has many advantages over the traditional PCR approach. Obvious advantages are that there is no need to send the additional Transport Stream (TS) packets with PCR information, reduced overall complexity, no need to consider clock jitters during synchronization, smaller time stamp field in the PES packet i.e., just 16 bits to encode frame number compared to 33 bits for the Presentation Time Stamp (PTS) which has a sample from the encoder clock. The time stamp field in this project is encoded in 2 bytes in the PES header, which implies that time stamp field can carry frame numbers up to 65536. Once the frame number of either stream exceeds this number, which is a possibility in the case of long video and audio sequences, the frame number is reset to 1. The reset is done simultaneously on both audio and video frame numbers as soon as the frame number of either one of the stream crosses 65536. This will not create a frame number conflict at the de-multiplexer during synchronization because the audio and video buffer sizes are much smaller than the maximum allowed frame number.

1.3. Mpeg transport stream (MPEG-TS)

PES packets are of variable sizes and are difficult to multiplex and transmit in an error prone network. Hence they undergo one more layer of packetization which results in Transport Stream (TS) packets.

MPEG Transport Streams (MPEG-TS)[4] use a fixed length packet size and a packet identifier identifies each transport packet within the transport stream. A packet identifier in an MPEG system identifies the type of packetized elementary stream (PES) whether audio or video. Each TS packet is 188 bytes long which includes header and payload data. Each PES packet may be broken down into a number of transport stream (TS) packets since a PES packet which represents an access unit (a frame) in the elementary stream which is usually much larger than 188bytes. Also a particular TS packet should contain data from only one particular PES. The TS packet header (Fig. 3) is three bytes long; it has been slightly modified from the standard TS header format for simplicity, although the framework remains the same.

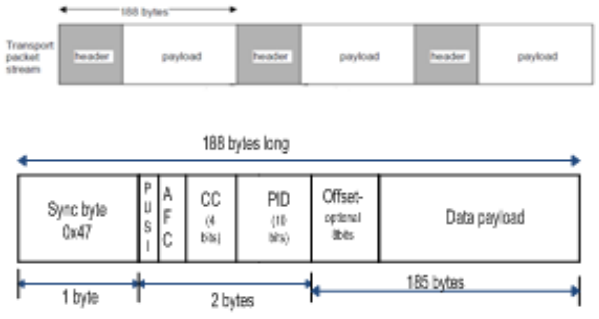

Figure 3. Transport stream (TS) packet format[4]

The sync byte (0x47) indicates the start of the new TS packet. It is followed by a payload unit start indicator (PUSI) flag, which when set indicates that the data payload contains the start of new PES packet. The Adaptation Field Control (AFC) flag when set indicates that all the allotted 185 bytes for the data payload are not occupied by the PES data. This occurs when the PES data is less than 185 bytes. When this happens the unoccupied bytes of the data payload are filled with filler data (all zeros or all ones), and the length of the filler data is stored in a byte called the offset right after the TS header. Offset is calculated as 185 – length of PES data. The Continuity Counter (CC) is a 4 bit field which is incremented by the multiplexer for each TS packet sent for a particular stream I.e. audio PES or video PES, this information is used at the de-multiplexer side to determine if any packets are lost, repeated or is out of sequence. Packet ID (PID) is a unique 10 bit identification to describe a particular stream to which the data payload belongs in the TS packet.

2. Multiplexing

Multiplexing is a process where Transport Stream (TS) packets are generated and transmitted in such a way that the data buffers at the decoder (de-multiplexer) do not overflow or underflow. Buffer overflow or underflow by the video and audio elementary streams can cause skips or freeze/mute errors in video and audio playback.

The flow chart of the proposed multiplexing scheme is shown in figures 4 and 5. The basic logic is based on both audio and video sequences having constant frame rates. For video, the number of frames per second value will remain the same throughout the video sequence. In an audio sequence since sampling frequency remains constant throughout the sequence and samples per frame is fixed (1024 for stereo), the frame duration also remains constant.

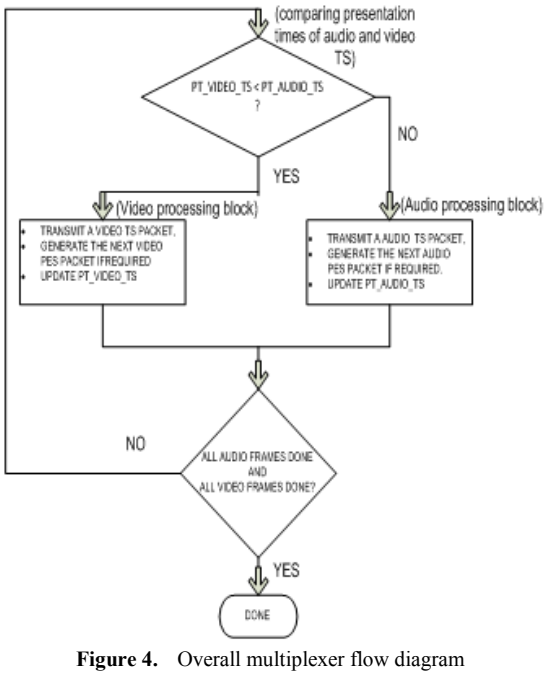

Figure 4. Overall multiplexer flow diagram

For transmission a PES packet which represents a frame is logically broken down to n (n depends on PES packet size) number of TS packets of 188 bytes each. The exact presentation time of each TS packet ($PT_{Audio/VideoTS}$) may be calculated as shown in (3) through (8), where $N_{TSVideo/Audio}$ is the number of TS packets required to represent corresponding PES packet or frame:

$$N_{TSVideo} = \frac{Video\ \ PES\ lengt\ h}{185} \quad (3)$$

$$TS_{VIDEODuration} = \frac{1}{FPS_{video} \times N_{TSVideo}} \quad (4)$$

$$PT_{VideoTS} = PT_{VideoTS} + TS_{VIDEODuration} \quad (5)$$

Similarly for audio:

$$N_{TSAudio} = \frac{Audio\ \ PES\ lengt\ h}{185} \quad (6)$$

$$TS_{AUDIODuration} = \frac{1}{FPS_{audio} \times N_{TSVideo}} \quad (7)$$

Where FPS_{audio} is given by $\frac{sampling\ \ frequency}{1024}$

$$PT_{AudioTS} = PT_{AudioTS} + TS_{AUDIODuration} \quad (8)$$

From (5) and (8) it may be observed that the presentation time of a current TS packet is the cumulative sum of presentation time of previous TS packet (of the same type) and the current TS duration. The decision to transmit a particular TS packet (audio or video) is made by comparing their respective presentation times. Whichever stream has a lower value; it is scheduled to transmit a TS packet. This makes sure that both audio and video content get equal priority and also transmitted uniformly. Once the decision about which TS to transmit is made, the control goes to one of the blocks where the actual generation and transmission of TS and PES packets take place.

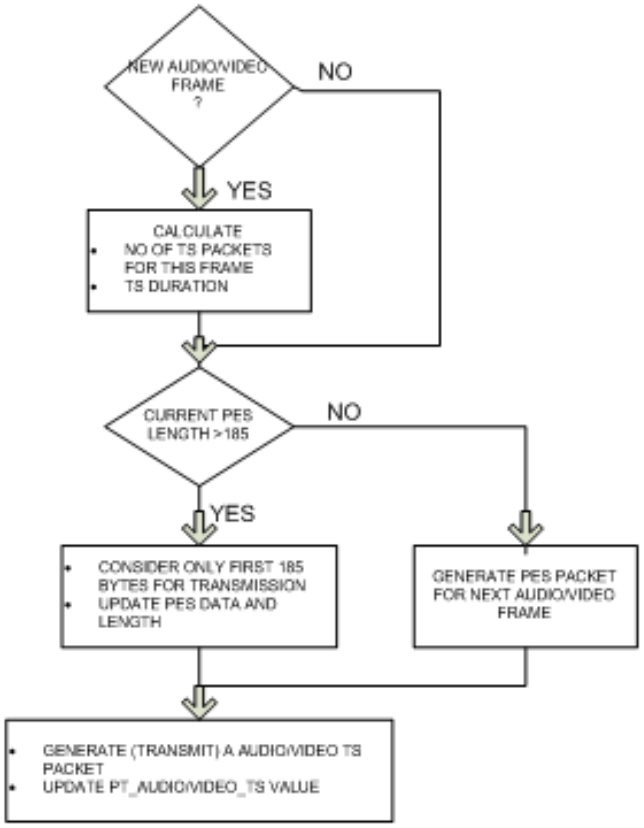

Figure 5. Audio/Video processing block

In the audio/video processing block (Fig. 5), the first step is to check whether the multiplexer is still in the middle of a frame or in the beginning of a new frame. If a new frame is being processed, (4) or (7) is executed appropriately, to find out the TS duration. This information is used to update the TS presentation time at a later stage. Next data is read from the concerned PES packet, if PES is larger than 185 bytes then only the first 185 bytes are read out and the PES packet is adjusted accordingly. If the current TS packet is the last packet for that PES packet, a new PES packet for the next frame (for that stream) is generated. Now the 185 bytes payload data and all the remaining information are ready to generate the transport stream (TS) packet.

Once a TS packet is generated, the TS presentation time is updated using (5) and (8). Then the control goes back to the presentation time decision block and the entire process is repeated till all the video and audio frames are transmitted.

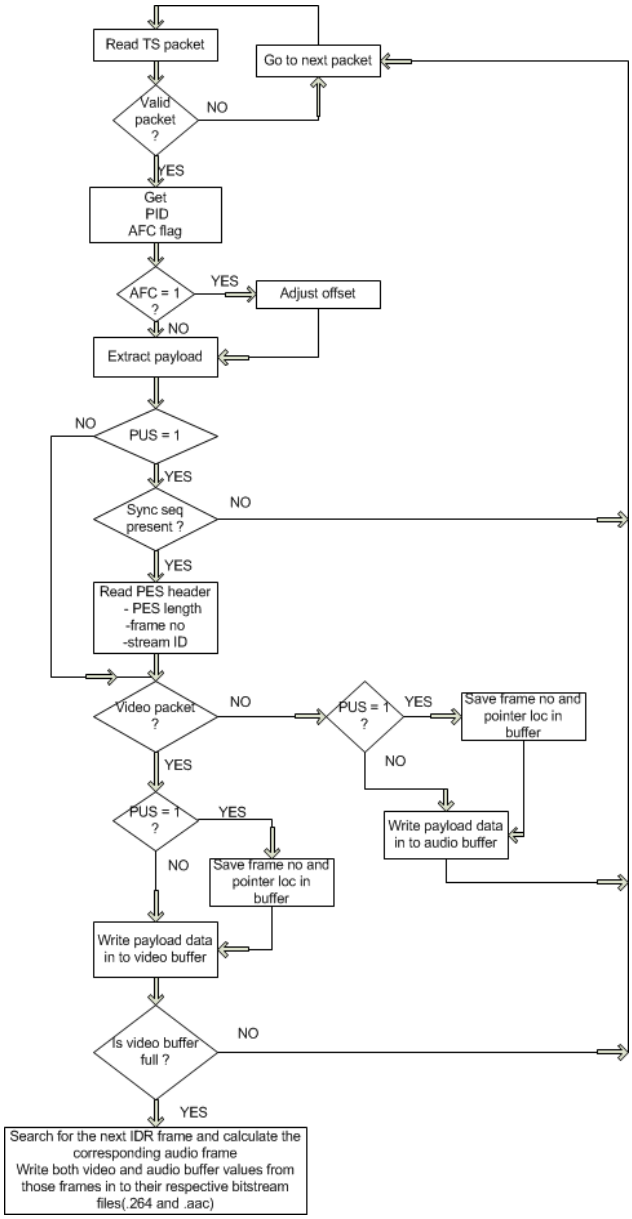

Figure 6. De-multiplexer flow chart

It has to be noted here that one of the streams i.e. video or audio may get transmitted completely before the other. In that case only that particular processing block is operated which is still pending transmission.

3. De-Multiplexing

The Transport Stream (TS) input to a receiver is separated into a video elementary stream and audio elementary stream by a de-multiplexer. At this time, the video elementary stream and the audio elementary stream are temporarily stored in the video and audio buffers, respectively.

The basic flow chart of the de-multiplexer is shown in the figure 6. After receiving a TS packet, it is checked for the sync byte (0X47), to check if the packet is valid or not. If invalid that packet is skipped and de-multiplexing is continued with the next packet. The valid TS packet header is read to extract fields like packet ID (PID), adaptation field control flag (AFC), payload unit start (PUS) flag, 4 bit continuity counter etc. Now the payload is prepared to be read into the appropriate buffer. By checking the AFC flag it can be known that an offset value has to be calculated or all 185 bytes in the TS packet have payload data. If the AFC is set then the payload is extracted by skipping the stuffing bytes.

The Payload Unit Start (PUS) bit is checked to see if the present TS packet contains a PES header. If so then, the PES header is first checked for the presence of the sync sequence (i.e. 0X000001). If not, the packet is discarded and the next TS packet is processed. If valid then the PES header is read and fields like stream ID, PES length, frame number are extracted. Now the PID is checked to see if it is an audio TS packet or video TS packet. Once this decision is made, the payload is written into its respective buffer. If the TS packet payload contained the PES header, information like frame number, its location in the corresponding buffer, PES length are stored in a separate array variable which is later used for synchronizing the audio and video streams.

Once the payload has been written into the audio/video buffer, video buffer is checked for fullness. Since video files are always much larger than audio files, the video buffer gets filled up first. Once the video buffer is full, the next occurring IDR frame is searched in the video buffer. Once found, the IDR frame number is noted and is used to calculate the corresponding audio frame number (AF) that has to be played at that time, given by (9).

$$AF = \frac{(sampling\ \ freq \times video\ \ frame\ \ number\ )}{(1024 \times fps)} \quad (9)$$

The above equation is used to synchronize the audio and video streams. Once the frame numbers are obtained, the audio and video elementary streams can be constructed by writing the audio and video buffer contents from that point (frame) into their respective elementary streams i.e. .aac and .264 files respectively. Then the streams are merged into a container format by using mkv merge[31] which is a freely available software. The resulting container format can be played back by media players like VLC media player[32] or Gom media player[33]. In the case of video sequence, to ensure proper playback, picture parameter sets (PPS) and sequence parameter sets (SPS) must be inserted before the first IDR frame, because both PPS and SPS information are used by the decoder to find out the encoding parameters used.

The reason that the de-multiplexing is carried out from an IDR (instantaneous decoder refresh) frame is because the IDR frame breaks the video sequence making sure that the later frames like *P* or *B*-frames do not use frames before the IDR frame for motion estimation. This is not true in the case of normal *I*- frame. So in a long sequence, the GOPs after the IDR frame are treated as new sequences by the H.264 decoder. In the case of the audio HE AAC v2 decoder can playback the sequence from any audio frame.

Table 2. Characteristics of test clips used

Test clip	1	2
Clip length (sec)	30	50
Video FPS	24	24
Audio sampling frequency (Hz)	24000	24000
total video frames	721	1199
Total audio frames	704	1173
Video raw file (.yuv) size(kB)	105447	175354
Audio raw file (.wav) size(kB)	5626	9379
H.264 file size(kB)	1273	1365
AAC file size (kB)	92	204
Video compression ratio	82.82	128.4
Audio compression ratio	61.15	45.97
H.264 encoder bitrate(kBps)	42.43	27.3
AAC encoder bitrate(kbps)	32	32
Total TS packets	8741	9858
Transport stream size(kB)	1605	1810
Transport stream bitrate (kBps)	53.49	36.2
Test clip size (kB)	1376.78	1576.6
Reconstructed clip size (kB)	1312.45	1563.22

Table 3. Video and audio buffer sizes and their playback times

video frames in buffer	Audio frames in buffer	video buffer size (in KB)	audio buffer size (in KB)	video content play back time (in sec)	audio content play back time (in sec)
100	98	771.076	17.49	4.166	4.181
200	196	1348.359	34.889	8.333	8.362
300	293	1770.271	52.122	12.5	12.51
400	391	2238.556	69.519	16.666	16.682
500	489	2612.134	86.949	20.833	20.864
600	586	3158.641	104.165	25	25.002
700	684	3696.039	121.627	29.166	29.184
800	782	4072.667	139.043	33.333	33.365
900	879	4500.471	156.216	37.5	37.504
1000	977	4981.05	173.657	41.666	41.685

3.1. Audio-Video synchronization

Synchronization in multimedia systems refers to the temporal relations that exist between the media objects in a system. A temporal or time relation is the relation between a video and an audio sequence during the time of recording. If these objects are presented, the temporal relation during the presentation of the two media objects must correspond to the temporal relation at the time of recording.

Table 4. Observed skew during playback

Transport stream packet number	Video IDR frame number chosen	Audio frame number chosen	presentation time (s) of chosen video IDR frame	presentation time (s) of chosen audio frame	delay (ms)	Perceptible?
100	13	13	.5416	.5546	13	no
300	29	28	1.208	1.1946	13	no
400	33	32	1.375	1.365	9.6	no
500	45	44	1.875	1.877	2.3	no
600	53	52	2.208	2.218	10.6	no
800	73	71	3.041	3.03	11	no
100	89	87	3.708	3.712	4	no

Since the output of (9) may not be a whole number, it is rounded off to the closest integer value. The theoretical maximum rounding off error is half the audio frame duration. This depends on the sampling frequency of the audio stream. For example for a sampling frequency of 24000Hz, the frame duration is 1024/24000 i.e. 42.6ms and the maximum possible latency will be 21.3ms. This latency/time difference is known as a "skew"[47]. The "in-sync" region spans a skew of -80 ms (audio behind video) and +80 ms (video behind audio)[47]. In-sync refers to the range of skew values where the synchronization error is not perceptible. The MPEG-2 systems define a skew threshold of ±40 ms[4]. In the proposed method once the streams are synchronized the skew remains constant throughout. This possible maximum skew is the limitation of the method; however the value remains well below the allowed range.

4. Results

Table 2 shows the results and various parameters of the test clips used. The results show that, the compression ratio achieved by HEAACv2 is of the order of 45 to 65 which is at least three times better than that achieved by just core AAC. Also H.264 video compression is of the order of 100, which is due to the fact that baseline profile is used. The net transport stream bitrate requirement is about 50 kBps, which can be easily accommodated in systems such as ATSC-M/H, which has an allocated bandwidth of 19.6 Mbps[17] or 2450 kBps.

4.1. Buffer fullness

As stated earlier buffer overflow or underflow by the video and audio elementary streams can cause skips or freeze/mute errors in video and audio playback. Table 3 shows the values of video buffer and the corresponding audio buffer sizes 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 20ms, 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. This demonstrates the effectiveness of the proposed multiplexing method.

4.2. Synchronization and skew calculation

Table 4 shows the skew for various start TS packets. The delay column indicates the skew achieved when de-multiplexing was started from different TS packet number. The maximum theoretical value is 21 ms because the sampling frequency used is 24,000 Hz (audio frame duration is 42 ms). The worst skew is 13 ms, but in most cases the skew rate is below 10ms. This is well below the MPEG2 threshold of 40 ms[4].

5. Conclusions

This paper presents a method to implement an effective multiplexing and de-multiplexing scheme with synchronization. The latest codecs H.264 and HE AAC v2 were used. Both encoders achieve very high compression ratios. Hence the transport stream bitrate requirement can be contained to about 50 kBps. Using the proposed method buffer fullness can be effectively handled with maximum buffer difference observed being around 20ms of media content and also during decoding, the audio-video synchronization was achieved with a maximum skew of 13ms.

6. Future Work

This paper shows the implementation of a multiplexing/de-multiplexing algorithm for one audio and one video stream i.e. a single program. The same scheme can be expanded to multiplex multiple programs by having a program map table (PMT). Also the same algorithm can be modified to multiplex other elementary streams like VC1[44], Dirac video[45], AC3[46] etc. The present method uses standards specified by MPEG2 systems[4]. The same multiplexing scheme can be applied to other transport streams like RTP/IP, which are used for applications such as streaming videos over the internet.

Since transport stream is sent across networks that are prone to errors, an error correction schemes like Reed-Solomon[43] or CRC can be added while coding the transport stream (TS) packets.

REFERENCES

[1] MPEG-4: ISO/IEC JTC1/SC29 14496-10: Information technology – Coding of audio-visual objects - Part 10: Advanced Video Coding, ISO/IEC, 2005.

[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 BSAC extensions.

[3] MPEG–2: ISO/IEC JTC1/SC29 13818–7, advanced audio coding, AAC. International Standard IS WG11, 1997.

[4] MPEG-2: ISO/IEC 13818-1 Information technology—generic coding of moving pictures and associated audio—Part 1: Systems, ISO/IEC: 2005.

[5] Soon-kak Kwon et al, "Overview of H.264 / MPEG-4 Part 10 (pp.186-216)", Special issue on "Emerging H.264/AVC video coding standard", J. Visual Communication and Image Representation, vol. 17, pp.183-552, April. 2006.

[6] A. Puri et al, "Video coding using the H.264/MPEG-4 AVC compression standard", Signal Processing: Image Communication, vol.19, pp. 793-849, Oct. 2004.

[7] 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

[8] ETSI TS 101 154 "Implementation guidelines for the use of video and audio coding in broadcasting applications based on the MPEG-2 transport stream".

[9] 3GPP TS 26.401: General Audio Codec audio processing functions; Enhanced aacPlus General Audio Codec; 2009

[10] 3GPP TS 26.403: Enhanced aacPlus general audio codec; Encoder Specification AAC part.

[11] 3GPP TS 26.404 : Enhanced aacPlus general audio codec; Encoder Specification SBR part.

[12] 3GPP TS 26.405: Enhanced aacPlus general audio codec; Encoder Specification Parametric Stereo part.

[13] E. Schuijers et al, "Low complexity parametric stereo coding ",Audio engineering society, May 2004 , Link: http://www.jeroenbreebaart.com/papers/aes/aes116_2.pdf

[14] MPEG Transport Stream. Link: http://www.iptvdictionary.com/iptv_dictionary_MPEG_Transport_Stream_TS_definition.html

[15] MPEG-4: ISO/IEC JTC1/SC29 14496-14 : Information technology — coding of audio-visual objects — Part 14 :MP4 file format, 2003

[16] DVB-H : Global mobile TV. Link : http://www.dvb-h.org/

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

[18] Open mobile vidéo coalition. Link : http://www.openmobilevideo.com/about-mobile-dtv/standards/

[19] VC-1 Compressed Video Bitstream Format and Decoding Process (SMPTE 421M-2006), SMPTE Standard, 2006 (http://store.smpte.org/category-s/1.htm).

[20] Henning Schulzrinne's RTP page. Link: http://www.cs.columbia.edu/~hgs/rtp/

[21] G.A.Davidson et al, "ATSC video and audio coding", Proc. IEEE, vol.94, pp. 60-76, Jan. 2006 (www.atsc.org).

[22] I. E.G.Richardson, "H.264 and MPEG-4 video compression: video coding for next-generation multimedia", Wiley, 2003.

[23] European Broadcasting Union, http://www.ebu.ch/

[24] S. Ueda, et al, "NAL level stream authentication for H.264/AVC", IPSJ Digital courier, vol. 3, Feb.2007.

[25] World DMB: link: http://www.worlddab.org/

[26] ISDB website. Link: http://www.dibeg.org/

[27] 3gpp website. Link: http://www.3gpp.org/

[28] M Modi, "Audio compression gets better and more complex", link: http://www.eetimes.com/discussion/other/4025543/Audio-compression-gets-better-and-more-complex

[29] PA Sarginson,"MPEG-2: Overview of systems layer", Link: http://downloads.bbc.co.uk/rd/pubs/reports/1996-02.pdf

[30] MPEG-2 ISO/IEC 13818-1: GENERIC CODING OF MOVING PICTURES AND AUDIO: part 1- SYSTEMS Amendment 3: Transport of AVC video data over ITU-T Rec H.222.0 |ISO/IEC 13818-1 streams, 2003

[31] MKV merge software. Link: http://www.matroska.org/

[32] VLC media player. Link: http://www.videolan.org/

[33] Gom media player. Link: http://www.gomlab.com/

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

[35] H.264/AVC JM Software link: http://iphome.hhi.de/suehring/tml/download/.

[36] 3GPP Enhanced aacPlus reference software. Link: http://www.3gpp.org/ftp/

[37] MPEG–2: ISO/IEC JTC1/SC29 13818–2, Information technology -- Generic coding of moving pictures and associated audio information: Part 2 - Video, ISO/IEC, 2000.

[38] MPEG–4: ISO/IEC JTC1/SC29 14496–2, Information technology – Coding of audio visual objects: Part 2 - visual, ISO/IEC, 2004.

[39] T. Wiegand et al, "Overview of the H.264/AVC Video Coding Standard ", IEEE Trans. CSVT, Vol. 13, pp. 560-576, July 2003.

[40] ATSC-Mobile DTV Standard, part 7 – AVC and SVC video system characteristics. Link: http://www.atsc.org/cms/standards/a153/a_153-Part-7-2009.pdf

[41] ATSC-Mobile DTV Standard, part 8 – HE AAC audio system characteristics. Link: http://www.atsc.org/cms/standards/a153/a_153-Part-8-2009.pdf

[42] H.264 Video Codec - Inter Prediction. Link: http://mrutyunjayahiremath.blogspot.com/2010/09/h264-inter-predn.html

[43] B.A. Cipra, "The Ubiquitous Reed-Solomon Codes". Link: http://www.eccpage.com/reed_solomon_codes.html

[44] VC1 technical overview .link: http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx

[45] Dirac video compression website. Link: http://diracvideo.org/

[46] MPEG2: ISO-IEC JTCl/SC29/WGll 13818-3 : Coding Of Moving Pictures and Associate Audio : Part 3 – audio Nov.1994

[47] G. Blakowski et al, "A Media Synchronization Survey: Reference Model, Specification, and Case Studies", IEEE Journal on selected areas in communications, vol. 14, PP 5 - 35, Jan 1996.

[48] I. E. Richardson, "The H.264 advanced video compression standard", Hoboken, NJ: Wiley 2010.

[49] M. Bosi and R. E. Goldberg, "Introduction to digital audio coding standards", Norwell, MA : Kluwer, 2003.

[50] T. Ogunfunmi and M. Narasinha, "Principles of speech coding", Boca Raton, FL: CRC press, 2010.

[51] Special issue: "Frontiers of audiovisual communications: convergence of broadband computing & rich media", Proc. IEEE, vol. 100, pg. 816-1009, April 2012.