Interactive Multi-source Media Synchronisation for HbbTV

Lourdes Beloqui Yuste
Discipline of Information Technology
College of Engineering & Informatics
National University of Ireland, Galway
lbeloqui@gmail.com

Padraig O Flaithearta
Discipline of Information Technology
College of Engineering & Informatics
National University of Ireland, Galway
peegrace@gmail.com

Hugh Melvin
Discipline of Information Technology
College of Engineering & Informatics
National University of Ireland, Galway
hugh.melvin@nuigalway.ie

***Abstract*—Hybrid Broadcast Broadband TV (HbbTV), unifying broadcast and broadband TV, is a TV platform which potentially provides users with new interactive services such as content related multi choice synchronised media streams. One such example is synchronising an audio stream via Internet Radio whilst watching an IPTV TV program.**

The principal scenario presented in this paper is that of providing users with a wide range of audio choice for sports transmission via IPTV. Users can select an IPTV channel plus an additional Internet Radio channel, where both channels are transmitting the same sports event. The independently produced media streams are then integrated and synchronised into a new IPTV stream.

The quality of the new service relies on the synchronised play-out of the multi source media at client-side. In sports, timing is critical for user's Quality of Experience (QoE). Two main issues shall be addressed/discussed, firstly the initial and secondly the continuous synchronisation of the sports play-out. We implement the former and outline our strategy for the latter.

This paper explains the main issues relating to multi-source synchronisation, within an HbbTV platform, and the prototype developed whereby video and audio from different sources are merged into a single MPEG-2 Transport Stream (MP2T) at client-side. We outline details of our initial synchronisation method. Finally we describe future work which will detect clock skew between the various media encoders/decoders to apply recovery techniques to ensure continuous synchronised play-out.

I. Introduction

Media Synchronisation is a key issue when merging content related media streams at client-side. Real-time sports transmissions are often simultaneously though independently transmitted via multiple TV and Radio channels. As such HbbTV provides an excellent platform to provide users with the choice between multiple media streams that broadcast and broadband technologies offer.

HbbTV aims to integrate all broadband and broadcast delivery systems. Broadband includes TV delivered over an IP Network whereas broadcast digital TV is delivered via satellite, cable or terrestrial platforms following Digital Video Broadcast (DVB) standards.

Broadband TV is composed of IPTV and TV on the Web. Internet Protocol TV (IPTV) is a paid service delivered over a privately owned managed network which employs DVB-IPTV. Conversely, TV on the Web delivered over the Internet is free and utilises a different range of protocols depending on the private media server that provides the service.

In our scenario, the client's interaction determines the audio chosen from Internet Radio to listen to, while watching the transmission of a sports event delivered over IPTV. This service should thus provides a synchronised play-out of the personalised video stream with the added audio selected by the user.

At this stage it is important to distinguish between time and timing synchronisation. The former, time sync refers to having multiple clocks agreeing on a time-of-day (thus with zero offset between them), whereas the latter, timing synch refers to two clocks running at the same frequency, regardless of any offset in time-of-day between them.

Synchronisation types can be also divided into different categories. Firstly there is intra-media sync, which tackles clock frequency assurance within a media stream. Secondly, inter-media stream, whereby multiple media streams from the same source synchronised for play-out. E. g., lip sync whereby video and audio media streams are synchronised for play-out. The thirdly type, multi-destination synchronisation, where different clients synchronise the play-out of the same received media. This is important in Massibely Multiplayer Online Gaming (MMOG) scenarios whereby each client should see the same response times. [1] Finally the forth type, multi-source synchronisation is where media streams are streamed from multiple servers and sync at client-side.

HbbTV provides an excellent platform for media synchronisation such as synchronising an IPTV channel with an RSS/Atom feed [2], or media received via different platforms, broadcast or broadband [3]. In order to provide interactive customised services the significant challenges of synchronised integration of media has to be performed at client-side following the client's media selection.

To provide this new service, the play-out of the final MP2T stream requires that the embedded media streams be synchronised. Although synchronisation parameters are typically not as restrictive as lip-sync, in sports events timing is still crucial.

In our completed and proposed prototypes the initial and continuous synchronisation is based on the requirement that all media providers are synchronised via Network Time Protocol (NTP) [4] to guarantee wall-clock synchronisation. Note

ISBN - 978-90-5986-410-8

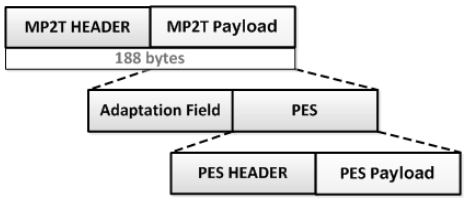

Fig. 1. High level MP2T packet structure

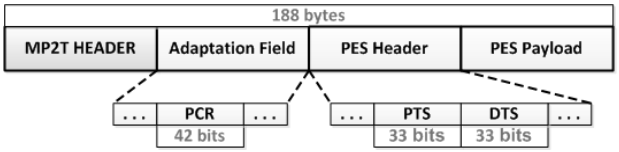

Fig. 2. Clock reference, PCR, and timestamps, PTS and DTS, within a MP2T packet

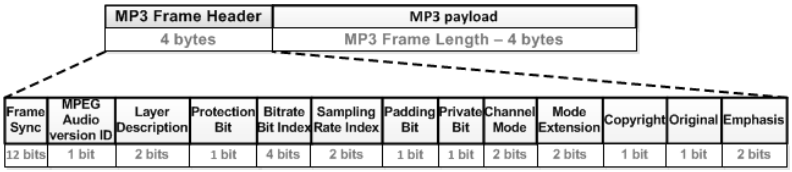

Fig. 3. MP3 frame header structure

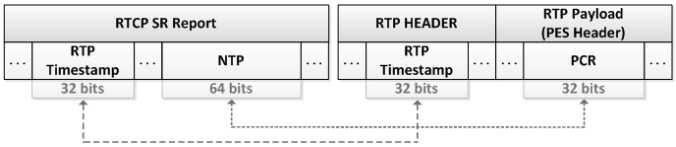

Fig. 4. RTCP SR packet relationship with a RTP packet conveying a PCR

that if NTP is also correctly implemented at the receiving client, this facilitated skew detection and prevents buffer overflow/underfill problems at client, though this problem can also be addressed though more conventional Phase-Locked Loop (PLL) solutions. NTP time synchronisation on the client also facilitated accurate each-way delay measurements, but if used with RTCP enables encoder's clock frequencies need to be synchronised with decoder clock frequency without the need for PLL.

Due to the requirement that media streamers will be synchronised via a protocol such as NTP there exists work that allows media streamers to share information about the sources of synchronisation. In [5] a the proposed solution involves to provide receivers sender's clock via the use of attributes in Session Description Protocol (SDP) which will provide details such as of the media clock source (NTP, Precision Time Protocol (PTP), Global Positioning System (GPS), Galileo or local) and its parameters.

In our prototype, the following synch issues will be addressed. Firstly the initial synchronisation of the audio/video of a football game and secondly the continuous synchronisation over the sport's event duration. We implement the first one and discuss the second one. The application protocols used by IPTV, Real-Time Protocol (RTP) and its companion, Real-Time Control Protocol (RTCP) provide the main tools to accomplish both of these synchronisation tasks.

The organisation of this paper is as follows: Section II explains synchronisation issues. Section III illustrates HbbTV. Section IV introduces MPEG-2 Transport Streams, MP3 and RTP/RTCP protocols. The prototype developed for the audio addition scenario within a MP2T stream is described in section V and the media synchronisation, initial and continuous, is discussed in section VI. Finally Section VII draws the conclusions and outlines future work.

II. Synchronisation Issues

The synchronisation of two or more media streams from different sources via different IP Networks adds extra complexity at the client-side or decoder. In a traditional streaming scenario we would have an encoder/sender and decoder/receiver. However, in this scenario we have two clocks at the two media servers and a third one at client-side.

Intra-stream media synchronisation, listed in previous section as the first synch issue, addresses the synchronisation within a single media stream. [1] As outlined above, encoder and decoder clock's frequency are typically synchronised via a PLL implemented at decoder. Encoder's clock is reproduced at decoder via encoder clock references which decoder uses to detect and fix clock skew. PLL works well in deterministic network environments but can perform very poorly in non-deterministic networks, such as congested public IP Networks.

In our scenario, we are synchronising different source media streams, which, in our case, consist of an MP3 audio and a separate MP2T audio/video stream at the decoder. Therefore our scenario is mostly concerned with inter-stream multi-source media synchronisation although intra-stream synchronisation will be addressed in a latter stage prototype. [1]

We have an audio/video stream embedded within a MP2T stream from the same source, the IPTV streamer, and an the extra audio stream from a different source, the Internet Radio media server. This thus presents multi-source, inter-stream and intra-stream media synchronisation issues though we deal only with first two.

III. HBBTV

HbbTV is a broadcast and broadband TV media receiver [6] with each technology employing different protocols. In our case the purpose is to synchronise an IPTV channel and an Internet Radio. Therefore the media protocols used in each delivery method shall be considered.

A. IPTV Standards and protocols

IPTV follows DVB-IPTV standard which specifies the use of MP2T streams as a media delivery method. [7] [8] [9] In Section IV.A MP2T packets structure is described in more detail.

User Datagram Protocol (UDP) is used at the transport protocol layer [10]. RTP and RTCP, are used at the Application Layer and although their use is not obligatory, their utilization

ISBN - 978-90-5986-410-8

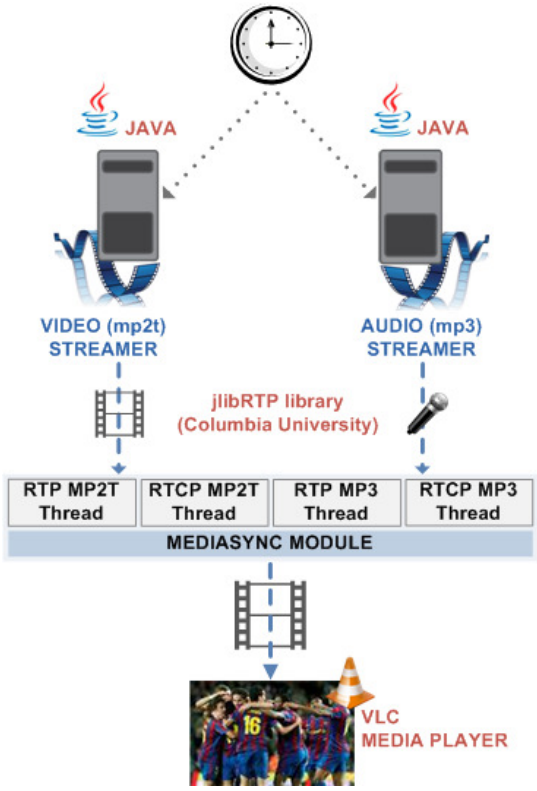

Fig. 5. Prototype High level design

is highly recommended. [11] [12]

In the prototype we use MP2T media delivery encapsulated in RTP packets. RTP/RTCP encapsulation of MP2T packets is explained in Section IV.C.

B. Internet Radio protocols

Internet Radio, uses multiple protocols depending on the media server provider. RTP is typically not used because Internet providers often penalise its traffic and RTP traffic is frequently blocked by firewalls.

The delivery methods, as opposed to RTP streaming, are mainly HTTP progressive download and adaptive HTTP. The Transport Protocol used is Transmission Control Protocol (TCP) [13]. Some of the protocols are HTTP based [14] such as Microsoft Smooth Streaming Protocol Specification (MS-SSTR) or HTTP Live Streaming (HLS) developed by Apple. [15] [16]. Finally Dynamic Adaptive Streaming over HTTP (MPEG-DASH) is an adaptive HTTP protocol fully supported by HbbTV.

Independent of the Applications Protocols used in the Internet Radio, MP3 is one of the most popular audio file delivery formats. Therefore it is the audio file format chosen for the prototype. In our case, as a preliminary work, we have used RTP at the Application Layer as media delivery protocol in order to prove the concept.

IV. Background Information

A. MPEG-2 Transport Stream

Media delivery in IPTV employs MP2T streams. An MP2T packet has a fixed 188 byte size that includes the header (4 bytes) and payload. The payload may include the Adaptation Field or/and the Packetised Elementary Stream (PES) header. In Fig. 1 the high level structure of MP2T packets is shown with the main elements, MP2T header, adaptation field, PES header and PES payload. [17]

Time related fields are divided between clock references and timestamps. The clock references, Program Clock Reference (PCR) are in the adaptation field while the timestamps are located in the PES header. In Fig.2 clock references and timestamps within an MP2T are depicted.

The PCR is a 27MHz clock divided between two fields the *PCR_base* (33 bits) and the *PCR_extension* (9 bits).

The Presentation Timestamp (PTS) and Decoding Timestamps (DTS) are 90KHz timestamp values that indicate when the PES shall be presented/decoded respectively in the play-out. With audio PES the PTS always equals the DTS. Therefore in every PES audio header, only the PTS is present.

B. MP3

MP3 audio file format is specified in [18]. The MP3 frame header is 4 bytes conveying all the audio information. The bit rate and the sample rate are used to calculate the audio frame size. With MPEG audio Layer I, Formula 1 is used, while with MPEG audio Layer II and III, Formula 2 is employed.

$$FrameLength = \left(12 \cdot \frac{BitRate}{SampleRate} + padding\right) \cdot 4 \quad (1)$$

$$FrameLength = 144 \cdot \frac{BitRate}{SampleRate} + padding \quad (2)$$

In Fig. 3 all fields in the MP3 frame header are shown. The 4 bytes of the MP3 frame header are included within the audio frame length.

C. RTP/RTCP

RTP is an Application Protocol for real time media delivery. Among other information, it provides a 32 bit RTP timestamp and a sequence number. Its companion protocol RTCP provides among other things, the relationship between the RTP timestamp and the NTP value which provides the wall-clock time. [19] The RTP payload type for MPEG-1/MPEG-2 video is 33 using the specific payload. [20]

In Fig. 4 the headers of RTP and RTCP Sender Report (SR) are shown with the time related fields providing the means to synchronise the media streams at client-side. The RTP timestamp in the RTCP packet relates the RTP timestamp to an NTP wall-clock time. Therefore RTCP SR provides a mechanism to relate NTP to PCR values, the later carried in RTP media packets. In Fig. 4 values are extracted to show the mapping as used in the prototype.

ISBN - 978-90-5986-410-8

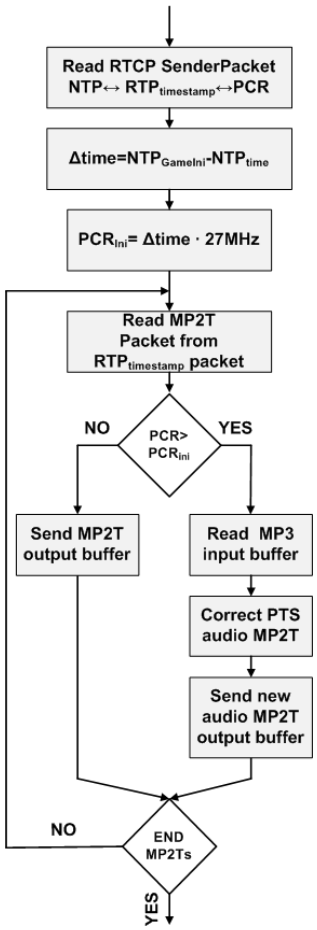

Fig. 6. Initial Sync performed in the MP2T video stream at client-side

V. Prototype developed

The prototype is developed with Java using jlibRTP library [21] with the creation of two media streamers, one for the MP2T video stream and another for the MP3 audio stream. The final video output, which integrates the two streams via the addition of the new audio stream is not played out in real-time (due to processing limitations) but stored in a video file for a later play-out. VLC [22] is the media player used for the resulting MP2T video stream play-out.

In Fig. 5 the High Level diagram of the developed prototype is shown. The key elements are the video and audio streamers, both synchronised via NTP, and the Java threads, for RTP and RTCP data, at client-side.

The prototype video streamed is an MP2T video stream of the 2011 Champions League Final, FC Barcelona versus Manchester United. The additional audio is an MP3 file of the same sports event transmitted by Catalunya Radio (Catalan National Radio Station).

Taking advantage of HTTP adaptive streaming, MPEG-DASH, HbbTV standard used in Internet Radio, where the audio server streams the audio adapted to the clients's requirements, we make the assumption that the MP3 audio characteristics are identical to the audio embedded within the MP2T stream. Thus, the insertion within the MP2T packets of the MP3 audio data is performed following the original audio MP2T packets distribution within the video file. [23]

The video is streamed conveying seven MP2T packets within a RTP packet following recommendation [24], whereas the MP3 file conveys one MP3 frame within an RTP packet.

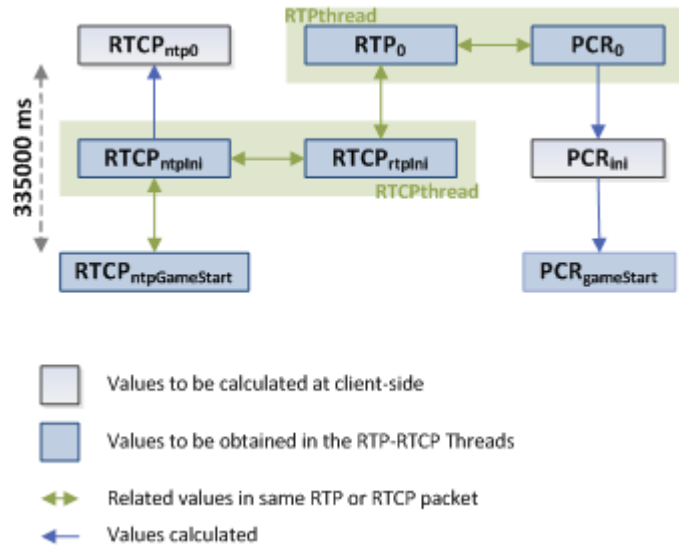

Fig. 7. Initial Sync performed in the MP2T video stream at client-side

TABLE I
Description of symbols used in Fig.7 and Fig.9

Value	Moment	Description
RTCP NTP	$RTCP_{ntp0}$	Wall-clock time first RTP packet received
	$RTCP_{ntpIni}$	Wall-clock time first RTCP SR packet received
	$RTCP_{ntpGameStart}$	Wall-clock time beginning of sport event
RTP timestamp	RTP_0	RTP Timestamp first RTP packet received
	$RTCP_{rtpIni}$	RTP Timestamp first RTCP packet received
MP2T PCR	PCR_0	PCR value first RTP-MP2T packet received
	PCR_{ini}	PCR value first RTCP-MP2T packet received
	$PCR_{gameStart}$	PCR value beginning of sport event

VI. Media Synchronisation

Both the initial and continuous synchronisation is based on the presumption that both media servers are synchronised via NTP. [4] This is a realistic assumption if and only if, the necessary NTP subnet is deployed. Other research based at NUI Galway has shown that single millisecond level sync is achievable over WANs once certain requirements are met. [25]

Synchronisation process is divided into two steps, the initial sports event synchronisation which is fully implemented, and the maintenance of the synchronisation over the period of the sports event being transmitted, which though not yet

ISBN - 978-90-5986-410-8

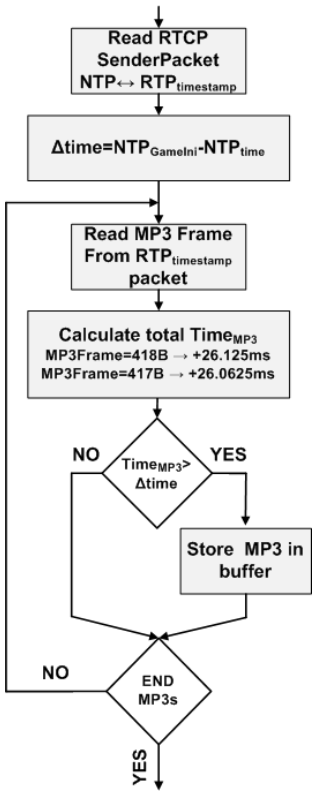

Fig. 8. Initial Sync performed in the MP3 audio stream at client-side

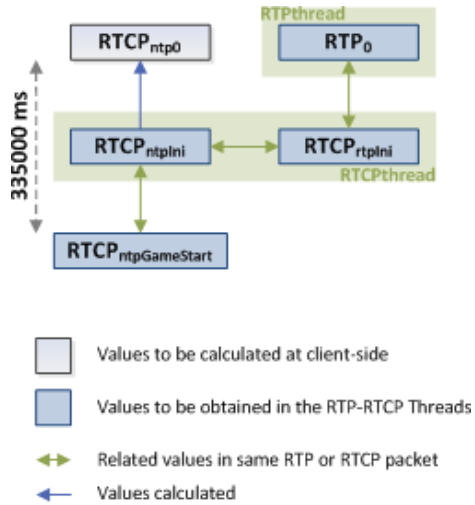

Fig. 9. Initial Sync performed in the MP3 audio stream at client-side

implemented, is described technically.

Time within a DVB-IPTV stream is transmitted via DVB Program Specific Information (DVB-PSI), particularly the Time and Date Table (TDT) and Time Offset Table (TOT) therefore clients received the media servers time information via transmission of time using both tables. TDT provides the UTC-time whereas TOT provides the local offset time and UTC-time. The interval of this tables transmission is between minimum of 25ms and a maximum of 30s. [26]

Our prototype will use the Event Information Table (EIT), part of DVB-PSI, to establish the game initial time of the sport event. The idea will be to establish a specific moment in time to perform the initial synch even if the game begins later on. In prototype we set a value of date 25/5/2011 time of sport event 19:45:00.000.

EIT time field start_time relates to TDT and TOT time field UTC_time (both 40 bits). The first field indicates the time the event begins and the second the time at client-side transmitted via the TDT and TOT table.

We will use the DVB PSI time related tables, TDT and TOT, and event table, EIT, to gather time information about the beginning time of the selected sport event. Note that the use of TDT/TOT is not necessary in a live environment as NTP embedded within RTCP is sufficient to synchronise streams.

A. Initial Synchronisation

In our prototype our initial synchronisation process has two main parts; firstly the processing of the MP2T video stream to extract and relate timestamps and secondly the processing of the MP3 audio stream.

Both parts use RTP/RTCP timestamps and NTP values to relate the beginning of the game. The MP2T initial synchronisation is based in the PCR values within the MP2T stream whereas the MP3 initial synchronisation utilizes the audio MP3 frame equivalent time values.

1) MP2T work flow: In Fig. 6 the work flow performed in the MP2T video file is described. When the first RTCP packet is received we note the RTP timestamp related to the NTP value of the wall-clock time. We label this NTP_{time}, the NTP time of first RTCP SR packet received by the client. Note that $NTP_{gameIni}$ the time the football game is scheduled is transmitted in our system using EIT table information as described above.

$$\begin{cases} NTP_{gameIni} = 1357415100 \leftrightarrow 25/5/2011 \;\; 19:45:00.000 \\ NTP_{time} \;\;\;\;\;\;\;\;\;\; = 1357414765 \leftrightarrow 25/5/2011 \;\; 19:39:25.000 \end{cases}$$

We use this relationship between RTP and NTP to establish a NTP_{time} as determined from the first RTCP SR for video. For all RTP media packets subsequently received, we extract the PCR values and using the RTCP NTP/RTP mapping, we can establishing the relationship to wall-clock time and then map difference to time elapsed since NTP_{time}.

Knowing the time difference in seconds, it is easy to obtain the equivalent in 27MHz clock for the PCR. In our case, the video begins at minute 5:35s therefore the difference between the beginning of the video and the beginning of the sport event is 335000ms.

$$\triangle Time = NTP_{gameIni} - NTP_{time} = 335000ms \qquad (3)$$

When this difference is translated into PCR values then we can obtain the PCR value when the audio insertion shall

ISBN - 978-90-5986-410-8

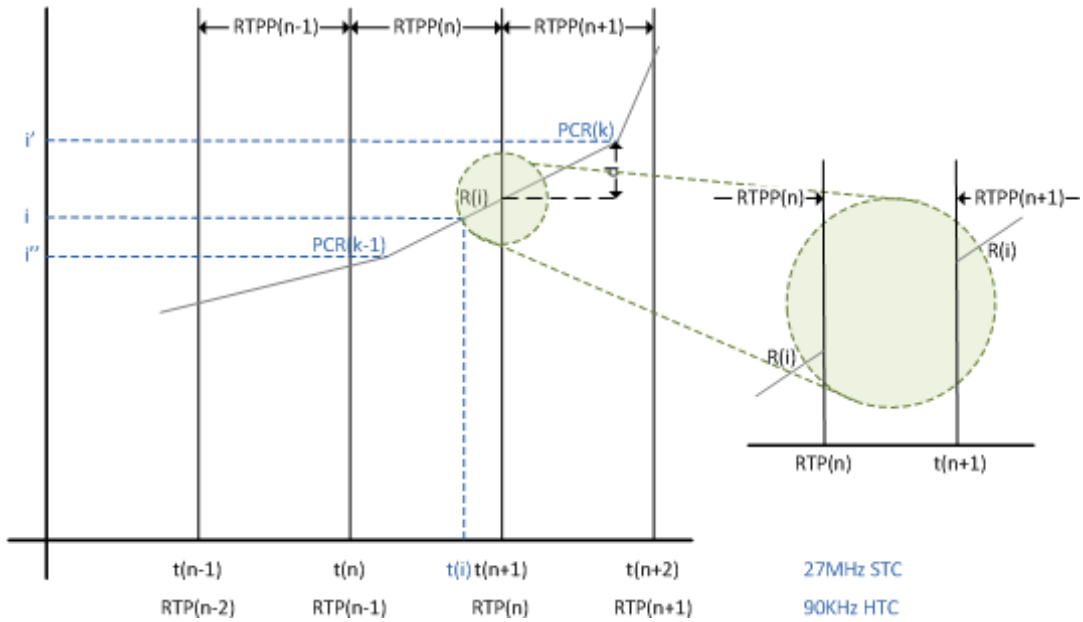

Fig. 10. Figure A.1: Timing from [9]

commence.

$$PCR_{ini} = \triangle Time \cdot 27MHz = 335s \cdot 27MHz = 9045000000 \quad (4)$$

A high level diagram of the threads in the prototype and how values are obtained and calculated can be found in Fig. 7 and the description of all variable names are depicted in Table I. There are two steps in the process, firstly values extracted when the first RTP packet with a PCR value arrives to the client and secondly when the client receives the first RTCP packet. In both stages the system stores the needed values to further initial synchronisation.

When the first RTP packet conveying a PCR value arrives to the client-side in the RTP Thread we store the RTP_0 and PCR_0. When the first RTCP packet arrives then we store $RTCP_{ntpIni}$ and $RTCP_{rtpIni}$.

Knowing $RTCP_{ntpIni}$ and $RTCP_{rtpIni}$ from the RTCP Thread and RTP_0 we calculate $RTCP_{ntp0}$. I.e., the NTP time when the first RTP packet was transmitted. Finally knowing PCR_0 and $RTCP_{ntp0}$ and $RTCP_{ntpGameStart}$ we calculate the final value $PCR_{gameStart}$ used for initial synchronisation the MP2T video stream.

2) MP3 work flow: In Fig. 8 the work flow performed on the MP3 audio file is described. When the first RTCP SR packet is received we note the RTP timestamp related to the NTP value of the wall-clock time.

For every MP3 frame subsequently received by the client, its time equivalent will be calculated, using formula 1 or 2 depending on the MPEG Audio Layer, to estimate the time since the beginning of the game. This difference in time is used to ensure than MP3 frame is inserted correctly into the video stream.

A high level diagram of the threads in prototype and how values are obtained and calculated can be found in Fig. 9

TABLE II
DESCRIPTION VALUES USED IN FORMULAE 5 AND 6 AND FIG. 10 [9]

Variable	Description
i	i”<i<i’ index
i”	Byte index of the last bit of the most recent PCR field
i’	Byte index of the last bit of the immediately following PCR
k	RTPP(n+1) first PCR index
R(i)	Transport rate i^{th} byte
PCR(k-1)	PCR encoded time at 27MHz
PCR(k)	First PCR time encoded
p	Number of bytes previous to the received PCR(k)
RTP(n)	Target Transmission Time
RTPP(n)	RTP packet(n+1) payload first byte
t(n+1)	Time which the first byte of RTP(n+1) payload enters the STD

and the description of all variable names are depicted in Table I. There are two steps in the process, firstly values extracted when the first RTP packet arrives to the client and secondly when the client receives the first RTCP packet. In both stages the system stores the needed values to further initial synchronisation.

When the first RTP packet arrives to the client-side in the RTP Thread we store the RTP_0. When the first RTCP packet arrives then we store $RTCP_{ntpIni}$ and $RTCP_{rtpIni}$.

Knowing $RTCP_{ntpIni}$ and $RTCP_{rtpIni}$ from the RTCP Thread and RTP_0 we calculate $RTCP_{ntp0}$. Finally knowing the difference between $RTCP_{ntpGameStart}$ and $RTCP_{ntp0}$ we calculate the time remaining to the beginning of the game.

Every time a MP3 frame is received at client-side we calculate the time equivalent and is added to $Time_{MP3}$. Only when the value of this variable is bigger than the time to beginning of the game then the MP3 frames are stored in the audio buffer.

ISBN - 978-90-5986-410-8

B. Continuous Synchronisation

In addition to the initial synch issue, whereby streams are initially aligned, we need to ensure that any clock skew between the different source media streams does not result in significant cumulative misalignment. E.g., a relative skew of 50ppm will result in a cumulative offset in alignment of 3 msec per minute for streams that are initially synchronised. In context of lip-synch requirements of 20-100 msec, this is quite significant. It is thus critical to determine the rate with which source clocks differ, i.e., relative skew so that continuous synchronisation can be maintained.

As part of the future work we plan to accomplish such clock skew detection in the MP2T stream via the formulae outline in ETSI document [9]. Fig. 10 shows the timing of values used whereas Table II describes the meaning of variables used. In future work we aim to calculate clock skew between the MP2T stream and the Internet Radio MP3 stream via the mapping relationship between the PCRs values and the RTP timestamps in Formula 7.

The overall objective is to use a common NTP time reference to establish the relative skew between the MP3 encoder and the MP2T PCR encoder. The MP3 skew relative to true time can be determined by comparing the elapsed NTP time from two consecutive RTCP packets from the MP3 stream and the elapsed time as represented by the number of MP3 bytes at the particular MP3 encoding rate.

The skew of the MP2T stream relative to true time can be determined using a similar approach using the NTP to PCR mapping formulae. Once both values of skew are known, the relative skew between streams can be determined and a skew compensation mechanism can be deployed. This can be either through MP3 stream silence addition or deletion. This approach is similar to that deployed for Voice over IP (VoIP) by [27]

The Transport Rate, R(i), between two consecutive PCR values is calculated in [9] as:

$$R\left(i\right) = \frac{\left(i' - i''\right) \cdot 27MHz}{PCR\left(k\right) - PCR\left(k-1\right)} \tag{5}$$

The time at which RTPP(n+1) arrives to the System Target Decoder (STD) is calculate in [9] as follows:

$$t\left(n+1\right) = \frac{PCR\left(k\right)}{27MHz} - \frac{p}{R\left(k\right)} \tag{6}$$

As a result the mapping formula in [9] can be applied using the RTP timestamps values.

$$PCR\left(k\right) \cong RTP\left(n\right) + 90kHz \cdot \frac{p+1}{R\left(i\right)} \tag{7}$$

Initial results for formula outlined in [9] indicate a mapping relationship between RTP timestamps and PCR values within a MP2T video stream. This formula applies as far we can appreciate as long as PCR and RTP timestamp increment is constant and PCR are send every a fix number of packets, therefore transport rate also becomes constant. For example a PCR increment of 1080000 when is sent every seven MP2T packets indicates a Transport rate of 32900bps. Thus the increment between two consecutive RTP timestamps is 1080000.

We can observe a mapping relationship between RTP timestamps and PCR values but we need to establish how the formula behaves within a variable transport rates and variable PCR increments. Further study shall be perform to see how formula behaves when not constant values are applied.

Future work will involve the use of MPEG-DASH specification for the delivery of the MP3 file. Further study of time and timing within the standard shall be performed and applied to develop a clock skew detection mechanism for MPEG-DASH Internet Radio delivery.

VII. Conclusion

Multimedia synchronisation can provide useful and exciting features to HbbTV particularly in live sports transmission scenario where timing is a key issue for viewers. Media synchronisation can offer interactive personalised services to IPTV users which in our prototypes allows them to watch a sports event and select any content related audio from Internet Radio.

The media integration needs to be tightly synchronised. Initial and continuous synchronisation challenges need to be addressed. We have achieved an initial synchronisation system using RTP/RTCP and its relation to RTP timestamp with the NTP wall-clock time and MP2T clock references, PCR.

Future work involves the clock skew detection at client-side of MP2T and MP3 streams so we can apply correction techniques to provide users with continuous synchronised play-out of a sports event. The final challenge will be to ensure intra-media synch, using either PLL methods or alternatives using NTP or equivalent. The solution to this latter challenge will use the same technologies deployed so far for initial synch and will be based on previous work by one ot the authors, as described in [27].

Acknowledgement

Research funded by the Irish Research Council (IRC) with SolanoTech.

References

[1] F. Boronat, J. Lloret, M. Garcia. Multimedia group and inter-stream synchronisation techniques: A comparative study, in: Elsevier, Information Systems 34 (2009), pp.108-131

[2] Beloqui Yuste, L.; and Melvin, H.,"Enhanced IPTV services through time synchronisation" *IEEE 14th Int'l Symp. on* Consumer Electronics, pp. 1-6, 2010.

[3] Beloqui Yuste, L.; Saleh Al-Majeed, S.M.; Melvin, H.; Fleury, M.; , "Effective synchronisation of Hybrid Broadcast and Broadband TV" Consumer Electronics (ICCE), 2012 IEEE International Conference on , vol., no., pp.160-161, 13-16 Jan. 2012

[4] Internet Engineering Task Force. RFC5905, Network Time Protocol 4: Protocol and Algorithms Specification. June 2010.

[5] RTP Clock Source Signalling. Draft-ietf-avtcore-clksrc-00. http://datatracket.ietf.org/doc/draft-ietf-avtcore-clockr/

[6] ETSI TS 102 796 v1.1.1. Hybrid Broadcast and Broadband TV (2010-06).

[7] ETSI 102 033 v1.1.1. Digital Video Broadcasting (DVB); Architectural Framework for the Delivery of DVB-services over IP-based Networks. (2002-02).
[8] ETSI TS 101 154 v1.9.1. Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream (2011-06).
[9] ETSI 102 034 v1.4.1. Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks (2012-02).
[10] Internet Engineering Task Force. RFC768, User Datagram Protocol. August 1980.
[11] Basso, A., CASH, G. L., Civanlar, M. R.: Real-Time MPEG-2 delivery based on RTP: Implementation Issues, Signal Processing: Image Communication, No. 15, 1999, pp. 165178.
[12] Macaulay, A., Felts, B., Fisher, Y.: White Paper IP Streaming of MPEG-4: Native RTP vs. MPEG-2 Transport Stream. Envivio, October 2005.
[13] Internet Engineering Task Force.RFC793, Transmission Control Protocol. September 1981.
[14] Internet Engineering Task Force. RFC2616, Hypertext Transfer Protocol HTTP/1.1. June 1999.
[15] A.C. Begen, T. Akgul, and M. Baugher, Watching Video over the Web, Part 1: Streaming Protocols Internet Computing IEEE, vol. 15, no. 2, pp. 5463, March-April 2011.
[16] A.C. Begen, T. Akgul, and M. Baugher, Watching Video over the Web, Part 2: Applications, Standardization, and Open Issues Internet Computing, IEEE, vol. 15, no. 3, pp. 5963, May-June 2011.
[17] ISO/IEC 13818-1. Information Technology. Generic Coding of Moving Pictures and Associated Audio: Systems (2000E).
[18] ISO/IEC 13818-3. Information Technology. Generic Coding of Moving Pictures and Associated Audio: Audio (1998E).
[19] Internet Engineering Task Force. RFC3550, RTP: A Transport Protocol for Real-Time Applications. July 2003.
[20] Internet Engineering Task Force. RFC2250, RTP payload Format for MPEG1/MPEG2 video. January 1998.
[21] Schulzrinne, Henning. jlibrtp - Java RTP Library. December 2006. Date of access: 24th June 2012. https://wiki.cs.columbia.edu:8443/display/ hgs/jlibrtp+-+Java++RTP+Library
[22] VideoLAN Organization. VLC Media Player. Date of access: 24th June 2012. www.videolan.org/vlc/index.html.
[23] Beloqui Yuste,L; Melvin, H.;, "Client-side Multisource Media Streams Multiplexing for HbbTV" Consumer Electronics (ICCE) - Berlin, 2012 IEEE International Conference on , vol., no., pp., 3-5 Sep. 2012.
[24] EBU Recommenation R130 (Unidirectional transport of constant bit rate MPEG-2 TS on IP Network). Geneva March 2010.
[25] Shannon, J.; Melvin, H,; Ruzzelli, A. "Dynamic Flooding Time Synchronisation Protocol for WSNs" GLOBECOM. California 2012
[26] ETSI TR 101 211 v1.1.1. Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI) (2009-06).
[27] Melvin, H.; Murphy, L.; "An integrated NTP-RTCP solution to audio skew detection and compensation for VoIP applications" Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on, vol 2, no., pp. II-537-40 vol 2, 6-9 July 2003.

ISBN - 978-90-5986-410-8