Frame accurate media synchronization of heterogeneous media sources in an HBB context

Arjen Veenhuizen and Ray van Brandenburg
Media and Network Services
TNO
Delft, The Netherlands
<arjen.veenhuizen>, <ray.vanbrandenburg>@tno.nl

***Abstract*— Next generation HbbTV applications promise exciting new possibilities with respect to heterogeneous content sources synchronization, like multi-angle picture-in-picture video rendering and . Frame accurate synchronization is required in order to achieve these new possibilities. In this study, as part of the ongoing European FP7 Next Generation Hybrid Broadcast Broadband (Hbb-Next) project, we investigated the use of content agnostic timestamp information for media synchronization. We demonstrate that frame accurate synchronization on a single device can be achieved between local transport streams and a remote MPEG-DASH stream using a customized GStreamer based test bed. In future research, we will extend our test bed for inter device and inter destination synchronization.**

Keywords—HBB-Next, Hybrid Broadband Broadcast, multi source media synchronization.

I. Introduction

Compared to a decade ago, distribution and consumption of multimedia content has changed radically. The worlds of broadcasters and broadband suppliers are converging to one hybrid platform. This transition has sparked the development of for example the Hybrid Broadcast Broadband TV (HbbTV) platform [1] and OpenHBB platform [2]. These platforms have mostly linked Internet and broadcast by offering on-demand content in addition to current linear broadcast services. Limitations of these platforms in the field of interoperability and lack of support for (synchronized) heterogeneous media have been discussed in literature [3, 4].

The European FP7 Next Generation Hybrid Broadcast Broadband, or HBB-Next, project [5], aims to develop technical solutions that allow for innovative new services to be delivered over hybrid-broadcast-broadband platforms. Topics being researched in HBB-Next include group recommendation systems, synchronization of heterogeneous media on a single or multiple devices and deployment of device and platform independent applications.

Frame accurate synchronization of heterogeneous sources within an HbbTV context allows for new possibilities like third party audio channel delivery and synchronized multi-angle picture in picture video feeds. This report discusses the preliminary results of our ongoing study on frame accurate inter media synchronization originating from heterogeneous sources within an HBB context on a single device focusing on both video and audio content.

Part of the research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287848.

II. Scenarios

When it comes to synchronization of multiple sources, three cases can be distinguished [6]:

- inter-media synchronisation (synchronizing play out of different media on a single device)
- inter-device synchronisation (synchronizing play out of different media on multiple co-located devices)
- inter-destination synchronisation (synchronizing play out of different media at geographically distributed locations and devices).

In this study, we focus on the first case: frame accurate synchronized play out of heterogeneous media on a single device. Future work will be directed towards the latter two cases and built on the results obtained so far.

In this first case, two or more media streams are delivered to the end-user and synchronized at the local device which receives these streams. These media streams can contain any kind of media, e.g. a broadcast DVB satellite feed, an IP based video feed, localized subtitles, a hearing impaired audio channel, (targeted) commercials, an accompanying second screen application, a website, recommended content, etc. Frame accurate synchronization will be required in case of for example picture-in-picture (e.g. a camera feed of the same scene from another angle) or tiled streaming (e.g. ultra high resolution video distribution where different spatial areas of the same video are delivered as different streams) [7].

When dealing with multiple streams, generally one stream is regarded to be the primary media stream. In most use cases, this will be the stream that is being sent over the broadcast network, although technically it might just as well be sent over the broadband network. The other streams are regarded as secondary streams and contain additional content to the primary stream. The secondary streams can either be sent over

ISBN - 978-90-5986-410-8

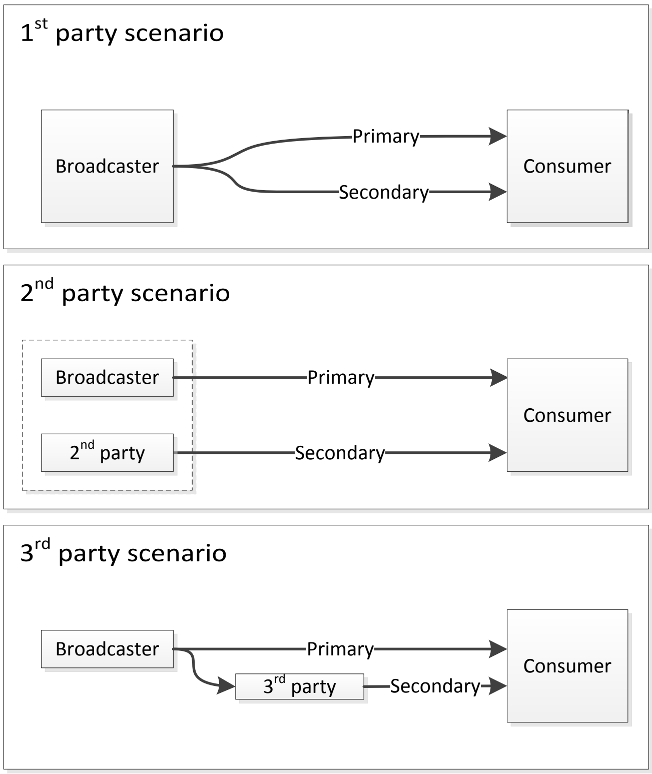

Figure 1. Delivery of primary and secondary media streams via different providers and hybrid networks.

the same network as the primary stream, but can also be sent over another type of network (e.g. over a broadband network).

Apart from possibly being sent over heterogeneous networks, these primary and secondary streams do not necessarily originate from the same content creator/owner. With respect to synchronization of heterogeneous media sources, three scenarios can be distinguished (see Fig.1).

- 1st party secondary stream provider. The provider of the primary stream also provides secondary streams. For example, the broadcaster provides both the DVB satellite video feed (via broadcast) and the hearing impaired audio feed (via broadband).
- 2nd party secondary stream provider. The providers of the primary and secondary feeds are somehow related to each other which means they can share clock information with each other.
- 3rd party secondary stream provider. The providers of the primary and secondary streams are not related to each other. For example, a third party is providing a commentary track.

In a broadcast delivery network, media streams can be remultiplexed multiple times. Remultiplexing of transport streams can result in a loss of absolute synchronization information embedded within the media streams. It is assumed that in the first two scenarios, the content creators have influence on the broadcast network, e.g. remultiplexing of media streams is performed within their bounds. In case of the third scenario, it is assumed that this is not the case.

Each scenario imposes different constraints on the possibilities and limitations for content synchronization. The fact that each stream can contain any sort of media further complicates the case.

The following technical challenges are acknowledged with respect to inter media synchronization [6]:

- Having different reference clocks. This especially applies to the third scenario.
- Using different transport protocols. Different protocols have different timing models and feedback mechanisms.
- Having different transmission delays. Media streams from different sources and/or over different transport channels have different transmission delays and jitter.
- Broadcast with internet multicast on on-demand delivery. The content delivery nature may be different. Whereas one stream is delivered continuously as DVB broadcast or IP multicast another may be transmitted on demand. Each streaming technology has its own characteristics with regard to delay and timing mechanisms.

III. RELATED WORK

The topic of inter media synchronization has been discussed abundantly in literature, each addressing one or multiple of the issues summarized in the previous section.

In [8], Ehley et al. propose an algorithm for inter-media synchronization based on controlling audio/video playback according to the Program Clock Reference (PCR) and associated Presentation Time Stamp (PTS). At the beginning of play-back, the algorithm checks whether the audio and video streams start at the same time. If this is not the case, the control algorithm delays the slave stream with a greater PTS until it is in sync with the master stream. As discussed in [9], master and slave can be switched if the slave stream starts to run in advance compared to the master stream. PTS values forms the basis of inter media synchronization in for example the Transport Stream specification [10, 11].

Additionally, usage of PTS values to synchronize heterogeneous media sources has been described in [12, 13]. In this case, the various sources use a synchronized wall clock (by means of for example GPS, NTP or Precision Time Protocol (PTP)) or clock recovery, and each media source inserts PTS values into the media stream in a synchronized manner. An important downside of this approach is that remultiplexing operations, which are part of many broadcast and some broadband networks, typically regenerate PCR values making it difficult to maintain clock synchrony between different sources. Furthermore, the PCR is attached to the service and contains no reference to the temporal position within the current event or stream. A timing reference is thus required to solve this issue [14].

Beloqui et al. [15] also envisaged the hybrid broadcast broadband delivery of audiovisual contents, including a synchronized play-out of multiple media streams delivered via

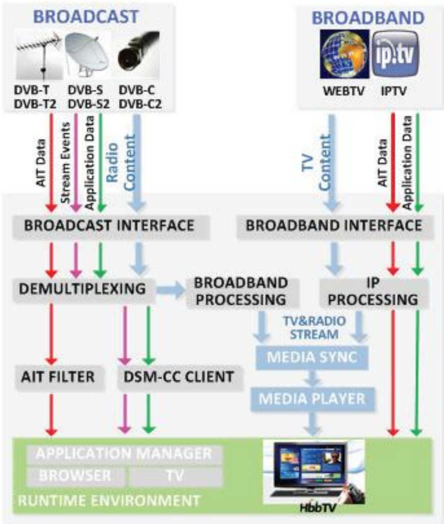

Figure 2. Hybrid Broadband Broadcast synchronization by employing a synchronization module [15].

HBBTV system consisting of a DVB stream and one or multiple MPEG2-TS [10] streams. To accomplish the effective synchronization between these broadcast and broadband streams the authors introduce a "media sync module" where the input streams will be synchronized. Fig. 2 gives an overview of this architecture.

First, some relation needs to be established between the clocks in the different streams at the source. The authors propose the use of NTP, GPS or Precision Time Protocol (PTP) depending on the situation.

Next, the delivery of a connection between the initial timestamp with the NTP clock over IP networks is performed via the Real-Time Control Protocol (RTCP) which provides the RTP timestamp, related to the MPEG2-TS payload timestamps, and their related NTP times. If an RTCP packet is sent by the source at the beginning of the media session the relationship is established and sent to the receiver.

The DVB system does not use RTCP and, thus, another method is needed to signal timing events. The authors propose to send the relationship between MPEG2-TS timestamps from the related program with the wall-clock time using the Event Information Table (EIT). The EIT transmits, among other information, the transport stream ID, event ID, start time and duration. It is proposed to add an extra field (PTS_timestamp) indicating the timestamps of the initial PTS event called, as in the MPEG2-TS.

At the client side, the sync module realigns the received media sources and renders the streams in sync. The authors did not implement this approach, so it is unclear to what accuracy this approach can provide synchrony.

The authors Rauschenbach and Putz [16] describe the SAVANT (Synchronized and scalable AV content Across NeTworks) project. In this project the best-suited transmission mechanism is chosen automatically for each media item depending on content properties, network parameters and number of subscribers. The system provides end-to-end support

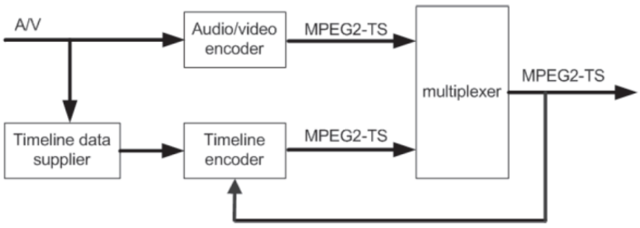

Figure 3. Timeline component insertion in MPEG2-TS [14].

to present two media streams synchronously at the receiver side, even if they are transmitted via different networks.

At the content provider side, the timestamps for the main content consist of NPT (Normal Play Time) descriptors, which are inserted into the MPEG2-TS using an NPT generator. The NPT value represents a reference to the system time clock (STC) value of the TS. The generation of the NPT descriptors and RTP timestamps is triggered to start simultaneously. At the start of the main program their values are reset and increased periodically and monotonically. In this way, a common time base between the multiple media sources is realized.

At the client side, a local clock is generated based on and synchronized to the extracted NPT values. The RTP Proxy presents an RTP packet to the player when the local clock reaches the value of the timestamp in that media packet. In order to do so, the client can delay the RTP stream if needed (due to for example network delays or jitter), and resynchronize it with the main program.

This approach assumes co-location of the different media sources in order to share the common clock and synchronization signal (e.g. scenario 1). Additionally, DVB now regards the use of NPT to be obsolete [17] and RFC 3550 [18] specifies that the initial value (in this case set to 0) to be randomly generated. Therefore, the applicability of this method in the current situation is limited.

Christopher Howson et al [14] describe the event timeline as a solution for frame accurate synchronization of hybrid media components used to compose personalized second screen TV services. The event timeline is a system for synchronization of TV service components delivered over different networks without being dependent on content type, transport protocol or timing model. The timeline information is precisely synchronized to the other existing protocol synchronization mechanism (e.g. PCR/PTS). The timeline information is inserted as an additional component of a transport protocol, as can be seen in Fig. 3.

An evaluation of this framework has shown that lip sync accurate synchronization between an audio multicast stream and broadcast DVB-T video stream can be achieved [14]. According to the authors, additional research is necessary to show whether this methodology is sufficiently accurate to synchronize two video streams frame accurate since this requires even more stringent synchronization than lip sync synchronization .

In order to achieve frame accurate media synchronization, we investigate the possibilities of utilizing PCR/PTS and timelines. The timeline approach and PCR/PTS both allow for synchronization in all three described scenarios while the methodology as developed within the SAVANT project is only applicable to scenario 1.

IV. Analysis

A. PCR/PTS approach

Program Clock Reference (PCR) based synchronization utilizes timing information which is already present within the MPEG-2 Transport Stream. The PCR, an accurate 27 MHz clock (tolerance ± 500 ns [10]), is used by decoders to reconstruct the encoder clock. Presentation Time Stamps (PTS), media unit timing information relative to the PCR, are used to time the output of the media unit precisely.

The PCR and PTS can be utilized for inter media synchronization originating from heterogeneous sources as long as MPEG-2 TS is used and a synchronized PCR is carried. This approach has two basic requirements:

- The PCR and PTS/DTS values must not be changed during transmission.
- The encoder clocks must be synchronized to the same PCR values.

The working principle is as follows: the receiver receives different media streams via heterogeneous transmission media (IP/DVB) and buffers the received streams until it is able to align the PCR (and thus PTS) of the different streams. This requires an adjustment of the input buffers according to the expected or actual reception delay since the different delivery paths may have different latencies. The media data is then processed and, according to the reconstructed time lines, presented to the user in a frame accurate manner.

The presented approach can be applied to any MPEG2-TS since PCR (and PTS) are defined in the MPEG2-TS specification [10]. No additional elements need to be introduced to implement this synchronisation mechanism.

B. Timeline approach.

Combining DVB broadcast with on-demand IP video or IPTV may enrich the user's TV experience. Complementary services, like picture-in-picture streaming of a second, synchronized, video stream coming from an IP-based content provider (1st, 2nd or 3rd party), would adapt their playout position to match the current playout position of the DVB service. In order to do so, a common clock providing absolute timing information about the different media sources is required. This absolute clock can be provided by inserting a common timeline into each media stream. This timeline relates the timing information embedded in each stream to a common clock.

Each media provider (broadcaster and secondary service provider) needs to include this common timeline into their stream, or, possibly, over the top via for example a broadband connection. The media player which renders all streams must extract the embedded timeline, and adjust media presentation timestamps accordingly.

Embedding timestamps in media streams has been covered in literature [14, 19] and has been standardized in ETSI TS 102 823 for MPEG2 transport streams [20]. According to this specification, a broadcast timeline is encapsulated in a Packetized Elementary Stream (PES). The PES is advertised in the PMT as private data. The timeline PES payload carries a broadcast timeline and a TV-Anytime (TV-A) description in the auxiliary data descriptor. The purpose for the TVA descriptor is to offer the possibility to link metadata to the timeline using a TVA id. Fig. 4 shows the process for generating and inserting the timeline. For each I-frame found, a timeline PES packet is created, using the PTS value from the I-frame's PES header. The broadcast_timeline descriptor is configured as direct timeline using absolute time values. In theory, also a relative tick format can be used, but this is not considered for the current approach. As tick_format 1000ticks/s was chosen. The timeline generation starts at the beginning of a show meaning that the absolute_ticks in the broadcast timeline descriptor starts with the value of zero. Once the timeline is generated, it is multiplexed to the MPEG2 TS. Since the timeline's PES headers carry the same PTS values as the video I-frame's PES header it is synchronized to the video stream.

C. Rationale for different approaches

The main difference between the PCR/PTS approach and the Timeline approach is that they are based on a relative and absolute timing paradigm respectively.

The timeline data is encapsulated into a separate elementary stream and adds absolute timing information to the media

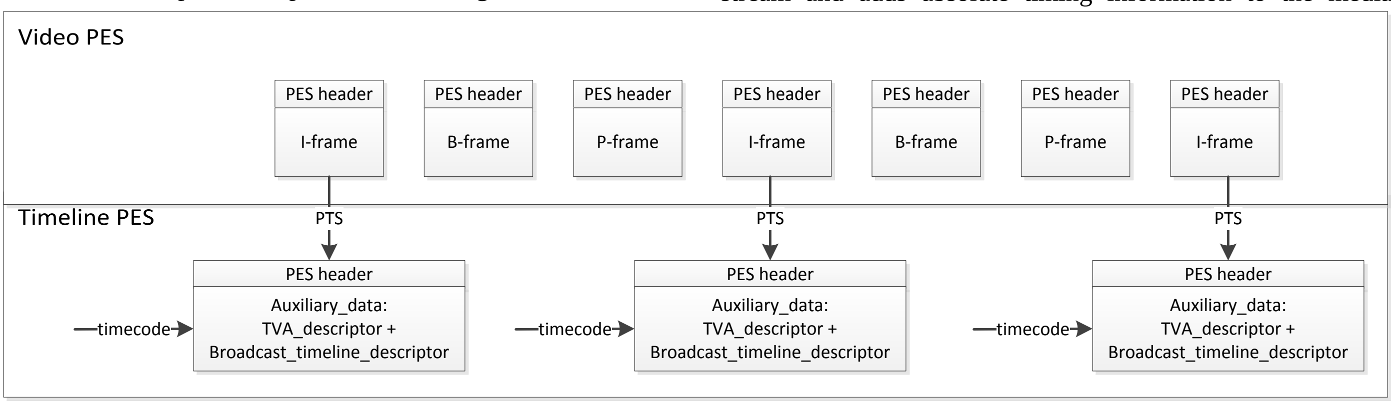

Figure 4. An overview of the timeline insertion process into an MPEG2-TS

ISBN - 978-90-5986-410-8

stream, allowing for the synchronization of metadata and on-demand media streams which normally lack synchronization information. The timeline is therefore codec- and transport protocol agnostic, but requires customizations at both the encoder and decoder side of the play out network (to inject and extract the timeline data respectively).

The PCR/PTS approach requires modification of the already embedded PCR/PTS values in a transport stream but can only provide relative timing information. This requires no additional resources and less processing complexity since it is basic part of the MPEG2-TS specification [10]. Frame-accurate inter-media synchronization of media with the same container configuration is thus relatively straightforward on both the encoder and decoder part of the network.

When taking the three scenarios as discussed in Section II into consideration, both approaches have their advantages and disadvantages. In case of the first party scenario, the PCR/PTS solution is regarded to be applicable since a common clock is available, assuming that the PCR/PTS values are not changed due to remultiplexing in the broadcast network. The same holds for the second party scenario, when clock synchrony at multiple locations is achieved by utilizing for example GPS clock signals or the NTP [21] or PTP [22] protocol. An important advantage of the PCR-based approach is that PCR/PTS synchronization is widely utilized in the broadcaster world. An important limitation is that this synchronization approach is only applicable for streaming media being transported in an MPEG2-TS container. In addition, this approach assumes no re-multiplexing occurs in the broadcast network.

The timeline approach is applicable to both scenarios. Especially the third scenario will benefit from its advantage in that it allows for synchronization of any content type as long as the timeline information is embedded or provided (using for example a manifest file). Furthermore, it is not prone to any PCR/PTS adjustments due to remultiplexing operations in the delivery network.

In this study we implement the timeline approach and demonstrate frame accurate synchronization of multiple heterogeneous media sources as discussed in the next section.

V. PROOF OF CONCEPT

In order to test the proposed solution, a test bed has been developed that demonstrates frame-accurate synchronization of heterogeneous media on a single device using a timeline-based synchronization mechanism. This test bed uses the Linux GStreamer framework [23], a element-based multimedia rendering and processing framework. A number of related and synchronized media content files in different file formats have been created: two video files one containing a synchronized audio track) and a picture-in-picture video stream. Each media stream has synchronized timeline information embedded within it.

By default, the GStreamer framework does not support timeline synchronization. Therefore, the standard transport stream demultiplexer has been extended in order to support the extraction of the embedded timeline data. Additionally, a custom element has been developed which allows for synchronization of multiple media streams by modifying GStreamer internal timestamp values according to the timeline data which is signaled on the GStreamer communication bus using messaging events [24].

A. How it works

Within the GStreamer platform, a string of elements allows for the parallel rendering of different media sources like a DVB stream, HTTP Live Streaming stream, MPEG-DASH stream or local file. Each element is added to a single pipeline which provides a common clock to all elements. This new element, in combination with a PTS for each media frame derived from each streams, allows for precise timing of media presentation.

To demonstrate the effective frame-accurate synchronisation of different media files with unrelated PCR/PTS values but related timelines, a pipeline has been constructed using two file sources and one MPEG-DASG source. The use of an MPEG-DASH source requires the use of a buffer to account for network delay and jitter, and to prevent buffer overflows and underflows (e.g. frame drops).

The synchronizer element is included for each media stream. Synchronization functionality can be enabled and disabled and additional offset (in number of frames) can be added as well.

B. Evaluation

The pipeline is ran on a laptop. Local files are stored on the hard drive of that laptop while the remote DASH stream is stored on a remote network server. The resulting video is rendered on an HDTV connected to the laptop via the HDMI interface.

Because it is difficult to show frame accurate synchronization in a still picture, one video has been cut in a left and right part. PCR/PTS values have been changed manually within these two files, while synchronized timeline information was added. Playing both files in the same renderer window next to each other will immediately show if both streams are synchronized.

When the pipeline is started, first some initial buffering is noticed. This is to be expected since one internet stream is requested. The local files are paused until sufficient data is retrieved from the remote host to start rendering all streams. By default, the synchronization mechanism is enabled and the synchronization element checks each passing media unit for its timeline value and adjusts the PTS accordingly. This results in frame accurate synchronized playback of multiple heterogeneous media sources. As can be seen in Fig. 5, the left and right part of the main video are synchronized, e.g. no visual distortion can be noticed. Additionally, multiple test subjects did not notice any lip-sync issues.

When disabling the synchronization module for one of the streams, desynchronisation can be noticed immediately. In Fig. 6, the synchronization module has been disabled for the right main video. Visual distortion can be noticed at the line of intersection of both files in the middle of Fig. 6. The picture in

Figure 5. HBB-Next multi source synchronisation test bed on a single device using two local files, a DASH video stream and HLS audio stream stored on a remote server. Audio/video content provided by RBB [25].

Figure 6. The effect of desynchronisation between two video sources by one frame is clearly visible in the middle of the main video.Audio/video content provided by RBB [25].

picture video feed are still synchronized to the left part of the main video.

VI. Conclusion and Discussion

In this study, we investigated frame accurate synchronization of heterogeneous media sources in an HBB context. By using the content agnostic concept of timeline synchronization, we were able to demonstrate frame accurate synchronization of multiple audio and video feeds, stored both locally and remotely using the MPEG-DASH streaming technology. The synchronization algorithm has been implemented in the open source GStreamer framework as an additional element and custom transport stream demultiplexer.

Within the context of HbbTV, two different media synchronization approaches have been studied: 1. synchronization based on common PCR clocks and 2. synchronization based on a common timeline. The main advantage of using a common PCR clock is that it is already widely used for local inter-media synchronization. It therefore does not require any modification at the provider or client side, other than that a common clock is available at the provider side. Important downside is that the PCR/PTS values ten to break due to remultiplexing operations between provider and client, breaking the synchronization in case the media streams arrive via different routes. Furthermore, a synchronized encoder clock is required.

On the other hand, timeline data is encapsulated into a separate elementary stream and adds absolute timing information to the media streams, allowing for the synchronization of metadata and on-demand media streams which normally lack synchronization information. This does, however, require modifications at both the provider and client side. Important advantage is that the timeline synchronization data does not break because of remultiplexing operations in the delivery path between provider and client.

Three content creation scenarios have been analyzed. Synchronization of heterogeneous media sources can be achieved with PCR/PTS synchronization when all content is created by a one party (e.g. the 1st party scenario), provided that a common clock is available. The same applies for the 2nd party scenario, having a high resolution common clock on distributed locations using for example NTP, GPS or PTP. When synchronization of multiple media sources from third parties is required, the use of timeline data is recommended. Being media container and transport protocol independent, it can provide clock synchrony for any content. However, it requires customizations at both the encoder and decoder side of the play out network.

A. Future work

This study has shown frame accurate synchronization of heterogeneous sources on a single device. In our future study, we intend to investigate frame accurate synchronization of heterogeneous sources in a multi device (e.g. second screen) environment and inter destination environment. Furthermore, we are looking into embedding timeline information in non-audio/video content in real time. Currently, we tested linear play back of media sources. In a future study we plan to investigate trick play and synchronized video on demand services as well with different encoding schemes like H.264 and containers like MP4. Additionally, we plan to release the relevant source code to the GStreamer community.

References

[1] Hybrid Broadcast Broadband, ETSI TS 102 796 V1.1.1, June 2010

[2] http://openhbb.eu/, accessed on 20 September 2012

[3] P. Altendorf, M. Riethmayer, M. Probst and R. Zimmermann, “Scenarios for HBB, what NoTube can bring to HbbTV (and Co.)”, Online, 30 January 2012

[4] Jean-Claude Dufourd, Stéphane Thomas, and Cyril Concolato. “Recording and delivery of HbbTV applications. In Proceddings of the 9th international interactive conference on Interactive television (EuroITV '11)”. ACM, New York, NY, USA, 51-54.

[5] http://www.hbb-next.eu/index.php, accessed on 20 September 2012

[6] J. Müller, M. Probst, J. Renz, S. Agarwal, C. Köhnen, C. Köbel, A. Veenhuizen and R. van Brandenburg, “Deliverable 4.1, Analysis: cloud-based services and service/content synchronisation.”, online, 30 March 2012

[7] http://www.fascinate-project.eu/, accessed on 20 September 2012.

[8] L. Ehley; B. Furht; M. Ilyas, “Evaluation of Multimedia Synchronization Techniques”, ICMCS’94, pp 514-519, 1994.

[9] Y. Ishibashi; S. Tasaka, “A Comparative Survey of Synchronization Algorithms for Continuous Media in Network Environments”, IEEE Conference on Local Computer Networks (LCN) 2000, Nov. 2000.

[10] “Information technology — Generic coding of moving pictures and associated audio information: Systems”, ISO/IEC 13818-1, Second edition, December 2012

[11] "Series H:Audiovisual and multimedia systems: Infrastructure of audiovisual services – Transmission multiplexing and synchronization.", ITU-T Rec. H.222.0, May 2006

[12] Matsumura, K., Kanatsugu, Y., Katoh, H., "Toward the Construction of Hybridcast", ATSC 2010 Symposium on Next Generation Broadcast Television, 2010, September 1st 2010.

[13] M. Armstrong, J. Barrett and M. Evans, "Enabling and enriching broadcast services by combining IP and broadcast delivery", BBC Research White Paper WHP 185, Sep 2010.

[14] C. Howson, E. Gautier, P. Gilberton, A. Laurent and Y. Legallais, "Second Screen TV Synchronization", IEEE International Conference on Consumer Electronics – Berlin, September 2011

[15] 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, 13-16 Jan. 2012

[16] U. Rauschenbach , W. Putz , P. Wolf , R. Mies and G. Stoll, "A Scalable Interactive TV Service Supporting Synchronized Delivery over Broadcast and Broadband Networks", Proc. IBC 2004 conference, 2004

[17] "Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in Hybrid broadcast/broadband environments", ETSI TS 102 809 V1.1.1, January 2010

[18] H. Schultzrinne, S Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003

[19] C. Köhnen, C. Köbel and N. Hellhund, "A DVB/IP Streaming Testbed for Hybrid Digital Media Content Synchronization", in the proceedings of the 2nd IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin 2012), September 2012

[20] "Digital Video Broadcasting (DVB); Specification for the carriage of synchronized auxiliary data in DVB transport streams", ETSI TS 102 823 V1.1.1, November 2005

[21] D. Mills, U. Delaware, J. martin, J. Burbank and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010

[22] "A precision clock synchronization protocol for networked measurement and control systems.", IEEE 1588 Standard, 2002

[23] http://gstreamer.freedesktop.org/, accessed on 20 September 2012.

[24] http://gstreamer.freedesktop.org/data/doc/gstreamer/head/pwg/html/section-events-definitions.html, accessed on 20 Spetember 2012

[25] C. Köhnen et al, "Deliverable 4.2, Design and protocol: middleware components content synchronization / cloud service offloading.", unpublished.

ISBN - 978-90-5986-410-8