Broadcast Delivery system for Broadband media content

Josu Gorostegui[1], Angel Martin[1], Mikel Zorrilla[1], Iñaki Alvaro[2] and Jon Montalban[3]

Abstract— **The pace of technology adoption in the broadcast industry is moving forward slower than in broadband media services because of different aspects. While the broadband penetration rate is growing sharply, the required investment to embrace the broadband content catalogue into the actual broadcast solutions is a major challenge. Nowadays, broadband media services offers more content by means of Internet as the main distribution system for media exchange. Due to the success of Over-the-top (OTT) services, there is no doubt that a global transition is about to come. Nonetheless, the broadcast transition period and required investments are considerably higher than in the broadband market. The principal aim of this paper is to assess the key points to take into account to assure the compatibility of OTT content in a broadcast environment. This paper dives into the implementation considerations to make broadband purpose video processing frameworks ready for broadcast pipelines. Market solutions usually perform transcoding for legacy compatibility needing a big processing capacity while losing fidelity during recompression. This approach will generate on the fly live content usable in broadcast contexts and technical environments while saving storage, mantaining the original encoded signal when possible. The approach is a reliable and cost-effective media delivery method optimized for live HTTP-based Adaptive Streaming media and real time broadcast media delivery with muxing correction. In order to show completeness and validate the presented aspects this paper describes the performed implementation. For this purpose, professional tools of broadcast validation and reference broadband sequences have been used.**

Keywords: DTV, Broadband, Broadcast, Transmuxing, DVB, MPEG Transport Stream, HTTP Adaptive Streaming (HAS), HLS, MPEG DASH, Buffer model, Interoperability, Technology adoption.

I. INTRODUCTION

IN the last two decades, the broadband Internet services has undergone an overall increase in the number of accesses per inhabitant in Europe. Due to this fact, the supply of content from Internet has increased exponentially [1][2]. Therefore, users are aiming for a wider content offer, as broadband operators also aim for growth in broadband subscribers base. As a consequence of the customer base growth, enriched and personalized content, such as multi-view TV, different language selection, and other features has grown. Apart from the previously mentioned aspects, broadcast technologies provide, generally, richer performance than broadband distribution technologies in terms of latency, stability and implacability. Notwithstanding, the broadband technology benefits from the bidirectional communication capabilities, contrasting with the essence of the broadcast delivery methods. This enhances the development of customized and interactive media services through users' requests.

However, in certain environments the cost-benefit ratio of upgrading the whole infrastructure and equipment is not affordable. Additionally, the transition period in broadcast requires more time than what markets demand. It is difficult to gauge the market adoption period exactly as it relies on several factors, but presumably it could be around 20 years, as happened from analog to digital [3].

On the grounds of these issues the following research designs a new transition, based on the existing ICT infrastructure that addresses not only the previously mentioned economic repercussions, but also those concerning practicability. The design of the transition tackles the problems which arise when trying to provide interoperability between broadband (OTT/IPTV) and broadcast environment contents.

While Digital Video Broadcasting (DVB) complies with the MPEG-2 standard, which has been designed to be used in noisy/lossy environments, broadband market systems are driven by different Quality of Service (QoS) approaches. This occurs because DVB has been designed to be used in environments where the errors are likely, whereas the broadband works over Internet and was originally created for data exchange rather than for audiovisual content transmission.

The main contribution of this paper is to analyze key points to make broadband content compatible with broadcast specifications ready to be delivered over broadcast infrastructures. As mentioned above, the requirements for broadcast are tight and strict and therefore the content designed for broadband often does not meet these requirements. This paper goes deeper designing and implementing the required workflow over a representative open source framework employing a computing model ready for broadband media processing, working in push and pull mode instead of free running. This approach would maximize the potential for reuse of all types of digitally produced content generating on the fly live content usable in broadcast contexts and technical environments while saving storage.

The remainder of this paper is organized as follows. Section II presents the background of the MPEG-2 transport stream (TS) standard, part of the MPEG-2 standards family, the most widely deployed standard for multimedia services in

[1]J. Gorostegui, A. Martin and M. Zorrilla are with the Department of Digital Tv & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net, Donostia-San Sebastin, 20009 Spain. Email: jgorostegui@vicomtech.org; amartin@vicomtech.org; mzorrilla@vicomtech.org

[2]I. Alvaro is with the Department of Multimedia R&D, Ikusi, Donostia-San Sebastin, 20014 Spain. Email: inaki.alvaro@ikusi.com

[3]J. Montalban is with the Electronic Technology Department, UPV/EHU, Donostia-San Sebastin, 20018 Spain. Email: jon.montalban@ehu.eus

the broadcast industry. Section III describes the most relevant issues to achieve the transition. In Section IV there is an overall overview of the implemented solution. To conclude, Section V shows the results of the approach and Section VI summarizes the conclusions and future work.

II. BACKGROUND

These days the delivery of every DTV system adopts MPEG-2 standard, using MPEG-2 transport stream (TS) (MPEG2-TS) to this purpose [4]. This format provides efficient mechanisms to multiplex different audio and video data streams into one delivery stream. These standards are widely used in the broadcast industry, as it is with Advanced Television Systems Committee (ATSC) [5] and Digital Video Broadcasting (DVB) [6].

In the context of broadband delivery technologies, the means used in the industry are HTTP Adaptive Streaming (HAS) techniques like HTTP Live Streaming (HLS) [7], Adobe HDS [8], Microsoft Smooth Streaming and Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [9]. HAS imitates traditional streaming via short downloads in HTTP client which downloads small video chunks. HAS is a pull-based protocol [10] that easily traverses middleboxes, such as firewalls and NAT devices. At the same time, it keeps minimal state information on the server side, making servers more scalable than conventional push-based streaming servers. Last but not least, concerning existing HTTP caching infrastructure, the protocol stack of HAS is not different than any other HTTP application. This allows distributed Content Delivery Networks (CDNs) to enhance the scalability of content distribution where individual segments of any content are cacheable as regular Web objects.

MPEG has developed a number of content delivery standards to support real-time delivery formats. These standards intended for use in broadcast environments are now mainstream in broadband networks. Standards like MPEG-1, MPEG-2, MPEG-4, ISO BMFF (Base Media File Format), MPEG-DASH and DVB, which are the widespread solutions to deliver media content in current broadband and broadcast networks. The current broadband contents, provided by means of CDNs empowering OTT services, are Variable bitrate (VBR) [11] in order to optimize the volume of traffic delivered by the network and the downloading rate of media across the Internet networks. In order to avoid the problem, hardware transcoding is widely used to convert Constant bitrate (CBR) content into VBR [12], providing the transport of different media qualities across the network and providing a channel capacity optimization according to the different hardware possibilities available today. In the case of the broadcast environment, CBR content delivery is required, adding stuffing bytes after the payload of media content. Generally speaking, transmuxing has the main advantage of maintaining quality, taking advantage that most extended video codec is H.264 (79% at 2017) [13], which is also used in broadcast.

In the same way that broadband content grows, encrypted content also does. To improve interoperability with protected content on the Internet, it is necessary to consider the encryption schemes used for example in HLS (AES-CBC), or MPEG DASH CENC[14]. Internet traffic is encrypted over 70% [15] and is expected to grow in the following years. Broadcast encryption could be maintained using other solutions if needed.

These specifications are a response from the industry to the tendency in the consumption environment of multimedia to enjoy through several types of devices. Concerning such heterogeneous ecosystem of specifications and delivery protocols tailored to the different devices used to play contents, it is important to make content production costs and technical delivery efficient. It is required to minimize overheads coming from the adoption of such wide spectrum of formats corresponding to specific contexts. In this line, forwards and backwards compatibility of the compression system is a major requirement during the specification of new formats and the development of compliant systems [16]. The forwards and backwards compatibility, cross compatibility, upgrade and downgrade compatibility usually means a translation of codecs, muxers and delivery protocols. The way different research activities and market solutions fix these situations is usually related to transcoding, moving from the stream to the raw audiovisual signals to generate an appropriate stream. Nevertheless, this process needs a big processing capacity while impacts on the fidelity of the generated content (PSNR) due to severe re-loss during recompression [17].

The major added value from the approach described in this paper lays on the design and the implementation which try to maintain the original encoded signal when possible. This benefit is twofold. On the one hand, the costly task for decoding and encoding the audiovisual flows is saved. On the other hand, the original fidelity of the compressed content is kept.

The first aspect is becoming critical due to the need of hardware embedded solutions able to process concurrent flows, exploiting built-in multimedia features. Complex software appliances typically consist of multiple software processes running concurrently to exploit the available computational resources in the hardware. However, the computational complexity of these software processes is often variable and the processes can interfere with each other [18]. This can be an issue for real-time applications with a fixed timeline like low delay video encoding. Keeping re-encoding out when possible, minimizes hardware processing stress.

The second aspect is related to preserve the original audiovisual signals, turning the processing idempotent or loss-free like other works [19]. In our case, when broadband codecs are compatible with broadcast specifications only remuxing is done. This concerns to video, audio and subtitles/captioning. This feature is mandatory for some business models where near-duplicate video copies accuracy performance needs to be achieved for duplication-aware storage, pirate video detection or polluted video tag detection [20]. Video watermarking to protect the video owner's copyright, video content integrity and authenticity [21] and avoid severe damage on the profit of producers, is another feature preserved by our solution. Here, the preserved media integrity also fosters the accuracy of audience share systems.

III. TRANSITION FROM BROADBAND TO BROADCAST

The following section explains the transition process, performing a comprehensive analysis of the MPEG standard, more precisely on MPEG-2 Transport Stream (TS), which is the media delivery used in DVB systems. Therefore, the evaluation dives into certain aspects and provides methods for synchronizing the video and audio, while ensures that the decoder respects the buffer model. This means a highly accurate handling of media timestamps and the muxing clock. Additionally, several relevant factors will be assessed: the VBR and CBR rate control schemes analysis, the required minimum PCR accuracy study, the computation model, the lightweight processing load, as well as other ones required to comply with the broadcast specifications.

As mentioned above, MPEG2-TS is used in DVB, Advanced Television Systems Committee (ATSC) and Internet Protocol TV (IPTV), although regarding to the broadband standards is also used in HLS to transport media stream in a container form encapsulating packetized elementary streams into a fixed size (188 bytes) packet structure and interleaved to form a distribution of multiple streams of audiovisual content. Fig. 1 shows the different types of elementary streams (e.g. video, audio), associated through the Packet identifier (PID) value to a specific Packetized elementary stream (PES). In the case of DVB, different streams are combined, forming a multiple program TS. To co-ordinate and control all the information Program-specific information (PSI) is also required, along with other signaling tables like Program Association Table (PAT), Conditional Access Table (CAT), Program Map Table (PMT) and so on.

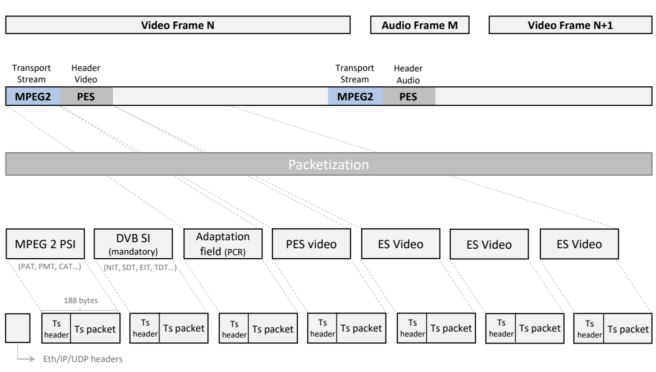

Fig. 1. MPEG2-TS muxing schema.

The MPEG-2 Systems are standardized and specified on [4], DVB project and standardized by ETSI, which describes the different multiplexing buffering methods and sync implementations, summarizing different requeriments and guidelines for each element.

The fundamental requirements for MPEG and DVB motivate the major design decisions our system. The main issues to be met by a MPEG2-TS muxer to generate DVB broadcast compliant contents pivot around three main aspects: first, the generation of a PCR clock; second, the translation of a VBR stream into a CBR one to track the highly tight restrictions in terms of packets in between contiguous PCR packets; third, a multi-stream re-arrangement of video and audio flows in order to provide favorable buffering conditions for thin devices.

All the measuring specifications are categorized in the technical report [22], and represents the basis for majority of hardware based MPEG stream analyzers. The Transport Stream integrity is checked with the MPEG-2 Conformance test defined in [22]. They are grouped into three categories according to their importance for monitoring purposes. In the case of HLS, which is the dominant standard for Adaptive Bitrate (ABR) streaming [13] and has MPEG-2 Transport Stream media segments, the first priority group of errors has implications in the decodability on the TS stream. The transport stream media segments inside the HAS playlist must be error free regarding the parameters in first column of Table I. The parameters in the second column of the table are bounded to DVB and broadcast environment. So they are more unlikely to be satisfied by broadband and Internet streaming services. These errors are more stringent in Digital TV broadcast, causing frozen frames and loss of lip sync.

TABLE I
LIST OF PARAMETERS FOR EVALUATION ACCORDING TO [22]

First priority	Second priority
TS_sync_loss	Transport_error
Sync_byte_error	CRC_error
PAT_error	PCR_error
Continuity_count_error	PCR_accuracy_error
PMT_error	PTS_error
PMT_error_2	CAT_error
PID_error	

A. PCR regeneration

One of the key aspects to address the correct playback in DVB environments is the Program Clock Reference (PCR). The PCR accuracy is required to occur at maximum of 100ms intervals in order to decode the stream correctly and play without artifacts in TV. These errors are inside second priority *PCR* errors. The PCR time stamps are necessary to synchronize the decoder System Time Clock (STC) with the encoder STC. The PCR conveys the time stamp information, which is sampled at 27 Mhz. A typical scenario of hardware muxing normally implies hardware based clock generation with an accuracy independent to the processing load. Nowadays, this resolution has been improved in all Linux based OS [23] under 1ms, facilitating the software clock generation.

The Video and Audio Decode Time Stamp (DTS) and Presentation Time Stamp (PTS) are used to create the PCR according to detected input bitrate in the source. The HAS input is analyzed during a fixed period of time, and the 27 MHz clock PCR is adjusted to media sampling frequency. The PCR is monotonically crescent in the same way as DTS and PTS, so the PCR can be determined from the video and audio DTS although the first is in 27 MHz base clock, while others are in 90 KHz clock. Therefore, clock conversion must be made to create PCR and clock drift happens as the incoming video source is in VBR.

Furthermore, padding null packets are employed to fix a constant rate rate between PCR, throughput bytes and PTS/DTS steps, according to the specification. To achieve the minimum PCR accuracy and other second priority errors, stuffing bytes are added, obtaining an output bitrate illustrated in Fig. 2, with the PCR interarrival time fixed through the previous calculation of the input bitrate.

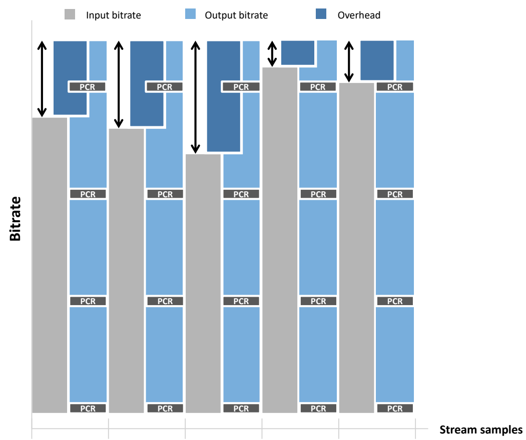

Fig. 2. Output bandwidth comparison with the input bandwidth.

B. Drift fix

The typical existing transmuxing and software-based transcoding architectures can suffer from clock drift [24],[25]. Since the video and audio processing time is not equal for frame types, the frame parsing time drift entails processing time drift in the same way as in the transcoding software schemes. An essential parameter for the definition of the pipeline is the calculation of the PCR rate. To this end, it is necessary to parse the manifest file which can provide the required bandwidth without any assessment. If the manifest does not provide this measure, the average bitrate of the media, the calculation have to fix a PCR rate while adding an overhead to avoid the undesired peaks that causes degradation in video quality if are not taken into account. It should be noted that the drift is caused by the *variability* of the bitrate in the broadband streams. It has also been found that the rate quality or the PTS/DTS and thus PCR bases changes during the same stream when different type of events happens (ads, change of the program,...), so this drift fix also prevents this type of error compensation required from the moment the media is maintained and not transcoded.

C. Buffer model

In order to approximate the buffer model defined in the standard [4], and following the principles of the leak method, the data rate calculated in the first step is used to respect the rate determined to the output, prioritizing this information over the data and video type specifications. The leaky bucket model, applied in the same manner to the broadband and broadcasts contexts, is a way to model the buffering requirements in order to playback any media stream smoothly. To this end, the buffering done in video and audio type streams are fixed with a independent delay. This is done due to the rearrangements that are made in the video frames (type P or B frames) and the absence of these patterns in the audio.

In this regard, the Transport System Target Decoder (T-STD) buffer model described in [4] references a conceptual decoder, which specifies that main buffer overflow, underflow, empty, delay and so on does not occur while the input stream is moving data to output stream on the basis of the timestamps. The Fig. 3, shows the ideal buffer operation, representing the DTS evolution over the time and how the buffer is unloaded every buffer window of time.

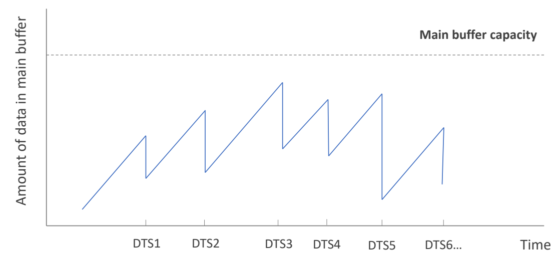

Fig. 3. Buffer model operation.

IV. IMPLEMENTATION

The implementation is done on top of GStreamer[1], an open source multimedia framework. In this way the solution take benefit from the wide set of extensible plugins which support different codecs, muxers and delivery protocols to be able to transform any current or future broadband stream.

The solution has 2 main blocks. The *muxer* concurrent pipelines that transform broadband streams into DVB compliant ones, and the *manager* which bootstraps and monitors the running pipelines.

Regarding the *manager*, first, it is in charge of discovering the input protocols and formats inspecting target broadband streams. In this manner, it shapes the concurrent pipelines to transform incoming streams into broadcast compliant ones. Second, as stated in the Section III, accurate incoming bitrate estimation is the main variable which tunes parameters for a DVB compliant format. In order to avoid overflows with live segments and the bitrate variability that occurs in the change of media contents, customized size overhead is introduced for all muxed streams. Overestimated incoming bitrate is inefficient in terms of bandwidth of the throughput, while underestimated scenarios breaks the essential lineal relation between number of bytes and PTS/DTS steps inside continuous PCR. While processing VBR contents an unforeseen

[1]https://gstreamer.freedesktop.org

peak of traffic can produce severe degradation of scheduled CBR. In that case, two options are applicable: update the CBR throughput from that moment raising CBR throughput indefinitely; or, the one implemented, to buffer the overhead sending frozen PTS/DTS values, till the peak is absorbed, while the PCR, which is related to the number of throughput bytes, increments. Both ones produces discontinuities that impact on the TV with artifacts. Third, the *manager* bootstraps the *muxer* pipeline and monitor potential issues such as Internet connection retries, performance degradation linked to logs, etc. This workflow is shown in Algorithm 1.

Algorithm 1 Manager algorithm

```
function AnalyseStream(URL, time, target)
Input: URL                                  ▷ manifest Url
Input: time             ▷ analysis time for bitrate calculation
Input: target           ▷ target resolution from master manifests
Output: pipeline        ▷ selected pipeline according to input type
    targetURL ← urlParse(URL, target)   ▷ option from master manifests
    bitrate ← fetchStream(targetURL, time)   ▷ average bitrate
    pipeline ← buildPipeline(targetURL, bitrate)   ▷ constructor

procedure MonitorPipeline(pipeline)   ▷ monitor&events trigger
    while playing do
        pipelineControl(pipeline)        ▷ cpu and memory status
        playlistStatus(targetURL)        ▷ track feed loss
        if !ok then
            event(message)               ▷ trigger event
            rebootPipeline(pipeline)     ▷ pipeline reboot

function Stream2DVB(URL, time, target)   ▷ concurrent stream
    pipeline ← AnalyseStream(URL, time, target)
    launchPipeline(pipeline)             ▷ pipeline startup
    MonitorPipeline(pipeline)            ▷ monitor and events trigger
```

Concerning the concurrent pipelines, the Fig. 4 shows the *muxer* pipeline runtime. The pipeline includes the source type and the appropriate demuxer and parsers depending on the type of content (e.g. HLS, MPEG-DASH).

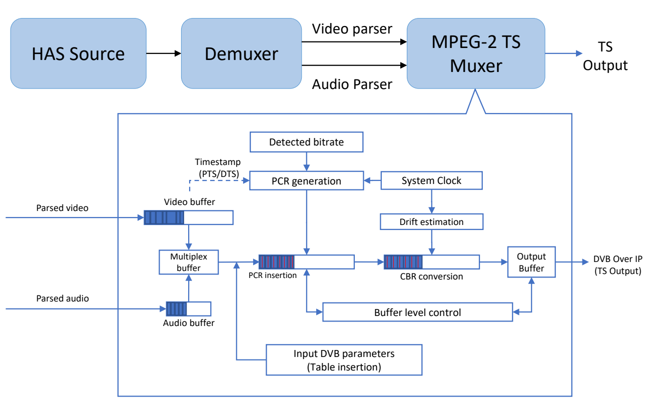

Fig. 4. Transmuxer pipeline and system architecture for the broadcast compatible output.

Going deeper, the MPEG2-TS muxer generates the right PCR value according to the media flows timestamps, corrects long-term deviations, turns the stream into CBR by means of null packets, buffers the different media flows to realign them, inserts PCR packets following an exact packet interval rate, and finally makes TS packets packaging in payloads of 1316 bytes. Main workflow features are described in the Algorithm 2.

Algorithm 2 Muxer algorithm

```
function pushPackets(buffer, throughput, PCR)   ▷ output packets
Input: buffer                        ▷ buffered data flow
Input: throughput                    ▷ accumulated throughput
Input: cbrThroughput                 ▷ scheduled CBR throughput
    while !empty(buffer) do          ▷ more data
        if throughput==cbrThroughput then
            return      ▷ potential overflow is postponed/buffered
        else
            if PCR-DTS respect bufferModelOffset then
                throughput+=outPacket(buffer)   ▷ within 1316 byte payloads

procedure InputBuffer(buffer, PTS, DTS)   ▷ incoming flow
Input: buffer        ▷ full video frame or media data for a PTS/DTS
Input: PTS                           ▷ PTS of the buffer
Input: DTS                           ▷ DTS of the buffer
    cbrThroughput ← cbr(PTS, DTS)    ▷ target CBR throughput
    if !PCR then
        PCR ← instant(PTS, DTS)      ▷ initial PCR
    else
        PCR ← deviation(PCR, throughput, cbrThroughput)
                                     ▷ PCR drift correction
    if isVideoBuffer(buffer) then    ▷ new video PTS/DTS
        if throughput != cbrThroughput then
            padding ← cbrThroughput-throughput   ▷ bytes to CBR level
            pushPackets(newBuffer(padding), throughput)
                                     ▷ output null packets
        throughput+=outPCR(PCR)      ▷ PCR packet within 1316 byte payloads
    pushPackets(buffer, throughput, cbrThroughput)
                                     ▷ output packets
```

It is important to highlight that the processing pipeline works in a push mode buffering, which means that the processing callback runs every time a new buffer comes. So it is necessary to buffer the input data to perform a inter-buffer processing. This has direct implications in the shape of the throughput that must be flat in terms or packets inter-arrival, removing bursts. To this end, the data is queued inside operational thresholds, specified by the *manager*. Dataflow maintained within the boundaries during the pipeline execution helps to isolate processes in different threads to avoid resource collision.

Last but not least, the support of encrypted content HLS AES-128[4] and MPEG-DASH CENC[14], is brought by OpenSSL hardware encryption/decryption lib. [2]

V. RESULTS AND VALIDATION

The implementation described in the previous section is tested with the industry standard MPEG analysis instruments in order to assure that the interoperability and MPEG stream

[2]OpenSSL Cryptography and SSL/TLS Toolkit https://www.openssl.org/

analysis tests are satisfied, evaluating the delivered overall quality of experience.

A. Tools and Setup

Professional hardware [3] and software [4] are used for the analysis and measurements of the results. Generated output has been modulated and displayed in a TV, adding all the necessary signaling and tables for correct tunning.

Furthermore, implementation of the software solution is made in low consumption ARM based motherboard running linux operating system. The ARM based platform also uses a linux based operating system, thus the plugins operation is assured. The server was running an Intel i7 Ivi Bridge Quad Core CPU with 16 GB of RAM, and transmuxer on a similar configured computer with Haswell CPU. The server was configured to receive stream from a stand-alone hardware encoder to avoid CPU overload cased by multiple encoding threads. On the other hand, the ARM based motherboard runs ARM Cortex-A9 CPU Quad core at 1GHz with 1GB of RAM.

B. Media sources

The tests have been performed in different scenarios, using mainstream CDNs and end-to-end closed systems with diverse sources, providing different types of video and audio codecs, resolution and input bandwidths.

The Table II shows the distinct sources, which have been tested with different conditions, addressing the desirable stability that is mainstream in DVB systems. The first 4 sources from the table have been used, as HLS it is made up of several Transport Stream segments. Linking them together and adding to a large enough unit test the MPEG-2 compliance analyzers are used to compare results between the HLS feed and the transxmuer output.

TABLE II

SUMMARY OF TESTS SOURCES USED

	Video Resolution	Bitrate(Kbps)	Video Codec	Audio Codec
Source HLS 1 (online & live)[5]	720p	1800	H.264	AAC
Source HLS 2 (online & live)	720p	1950	H.264	AC3
Source HLS 3 (online & live)	720p	2250	H.264	AAC
Source HLS 4 (offline & live)	1080i/1080p	4000-8000	H.264	AAC
Source DASH 5 (online & live)[6]	720p	1825	H.264	AAC
Source DASH 6 (online & live)	720p	2000	H.264	AAC
Source MSS 7 [7] (online & live)	720p	1950	H.264	AAC
Source MSS 8 (online & no-live)	720p	2275	H.264	AAC

[3]Tektronix Transport Stream Compliance Analyzer
[4]Elecard Stream Analyzer and DiviSuite
[5]Akamai (Mediaset sources, CBS, ...)

C. Results

Examining the PCR, PTS, and DTS values over time of different HLS outputs directly, showed on the Fig. 5 as an example, the VBR output shows different slopes and irregular behavior over the time. On the other hand, the Fig. 6 shows a fixed constant evolution of the identified packet types in the graph, which also address the difference between CBR and VBR.

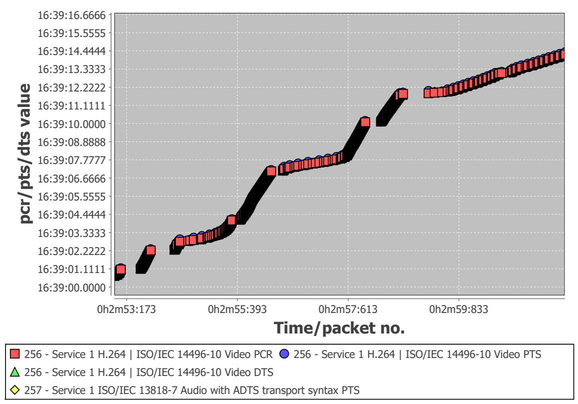

Fig. 5. PCR/PTS/DTS graph on VBR fragment.

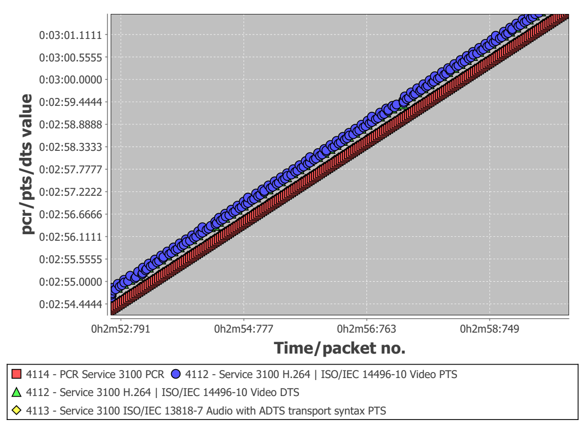

Fig. 6. PCR/PTS/DTS graph on CBR fragment.

Therefore, it can also be observed that comparing Fig. 6 (a broadband stream transformed into DVB compliant) to Fig. 7 (a native DVB signal), in both of them the PCR evolution keeps in a constant rate over time, which validates the muxing output as it has the same patterns as in the DVB systems. As it can be seen in both examples, compliant offset value between PCR and video DTS and PTS are shown.

Taking the TS rate into account and measuring in terms of byte index, the PCR accuracy is measured, calculating the PCR value difference from what is expected. It must be within $\pm 500ns$ for the color subcarrier to be synthesized from the system clock. The muxer complies the different

[6]MPEG-DASH Compliant Dynamic MPD segment templates
[7]Microsoft Smooth Streaming Live and PlayReady Test Server Smooth Streaming assets

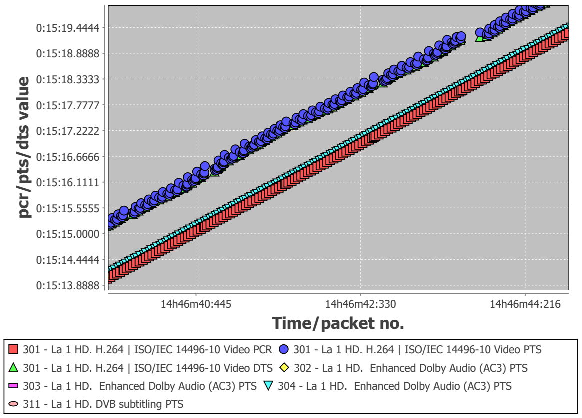

Fig. 7. PCR/PTS/DTS graph directly recorded from DVB source.

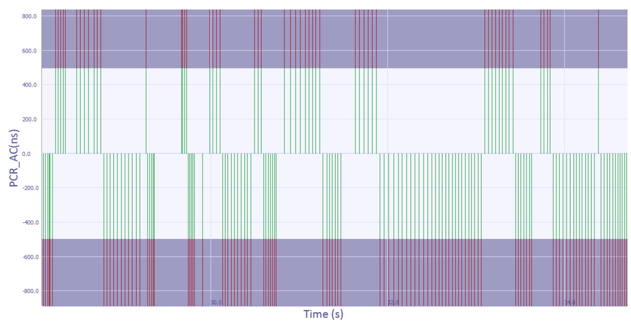

Fig. 8. PCR accuracy from a Internet source source.

demarcations (MGF1, MGF2 and MGF3) specified in Table 5.1 in the document [22], assuring that all the frequency components related to the PCR measurements are satisfied, delimiting the range of drift rate and jitter frequencies.

Fig. 8 shows how the HLS VBR sources from Internet do not fulfill the minimum required accuracy, and CBR source is required. Fig. 9 shows the results of PCR accuracy analysis extracted from the output of the implemented *muxer*. In this case, the MGF3 filter profile has been used for the tests.

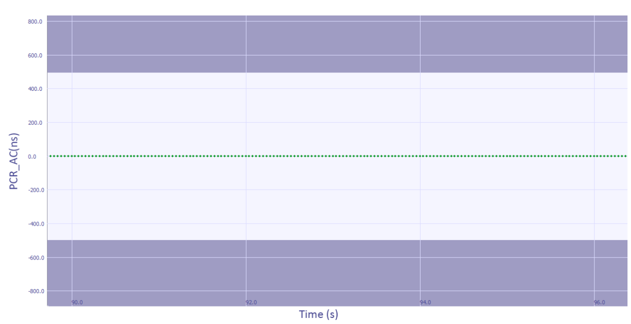

Fig. 9. PCR accuracy after transmuxing.

Another measuring test for correct display of the video is the PCR arrival interval, which the DVB recommends being not greater than 40 ms. However, values outside the range of 0-100 ms shows irrecoverable errors and display artifacts or glitches. A DVB compliant modulator or multiplexer should recover PCR inaccuracy below the 100 ms.

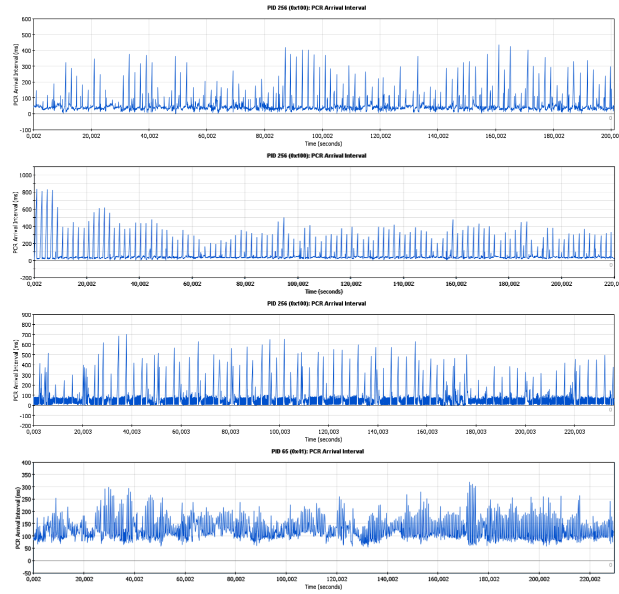

Fig. 10. PCR Arrival Interval of the HLS sources from Table II.

Fig. 10 shows how the time interval behaves between two consecutive PCR. This varies sharply for broadband sources, possibly causing a clock jitter or drift that could end in the receiver/decoder out of lock. The transmuxer output stays in the range below the 40 ms (Fig. 11), showing lower results depending on how much representative is the bitrate calculation made in the first step of the *manager*.

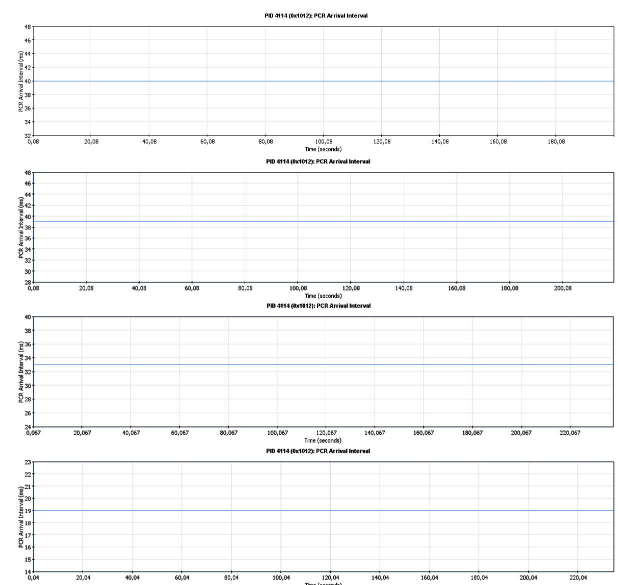

Fig. 11. PCR Arrival Interval of the HLS sources from Table II after transmuxing.

The CPU consumption has been compared in x64 system, comparing the transmuxing low CPU consumption with the CPU encoding (H.264/AAC), enabling the low latency capabilities on the encoder to achieve the best performance in detriment of the visual quality. Transmuxing does not manipulate the sources, so the quality remains exactly the same as in the source. The encoding done in the tests was with a 2000 Kbps bitrate.

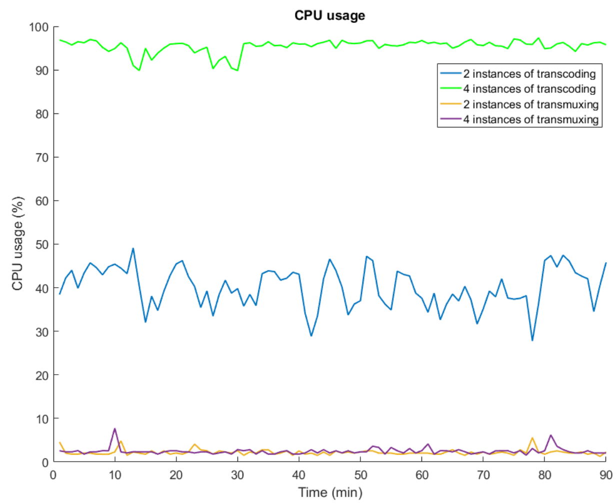

Fig. 12. Transmuxing and transcoding CPU consumption comparison in x86 systems.

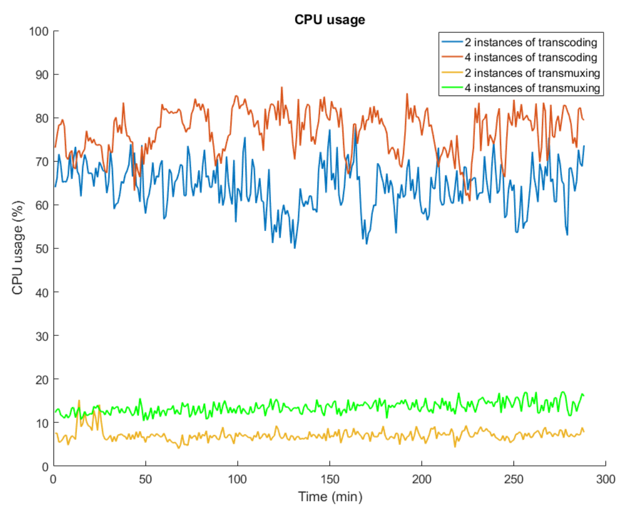

Fig. 13. Transmuxing and hardware transcoding CPU consumption comparison in ARM systems with hardware video acceleration capabilities.

All the figures are representative samples of achieved performance from above 1000 hours of processed content to identify performance degeneration issues.

The encryption does not add overhead to the CPU consumption, staying in the same levels when processing content encrypted with HLS AES-128[4] and MPEG-DASH CENC[14], since hardware encryption/decryption libs are used. This grants the support to different DRM systems, remaining the same pipeline structure and decrypting the media content before demuxing with the HTTP client manager.

VI. CONCLUSIONS

In this paper, we have described a method for transmuxing OTT content, presenting the constraints of the DVB environments compared to the broadband related ones. It includes a study of different aspects to make transmuxing in a software based approach, which makes compatible with multiple technologies, standards and protocols for media delivery that are changing over time.

The proposed system can process up to eight MPEG streams, which limit is fixed by the I/O interfaces and not the CPU usage, as metrics has shown in Section V. Furthermore the system allows any decodable input type from Internet to be muxed into standard codecs, making DVB fully compliant Transport Stream. The comparison with transcoding was also significantly interesting, as the vast amount of CPU resources or need of specific hardware to make use of transcoding always has a minimal impact on quality loss compared to the source that transmuxing does not have.

Live and on-demand streaming experiences and formats have been validated, synthesized and implemented over ARM based architecture, doing stress tests alongside with x86 platforms and making objective (MPEG-2 compliance tests) and subjective (lip-sync and fluid playback) tests.

APPENDIX I

ACRONYMS

Acronyms		Meaning
ABR	=	Adaptive Bitrate
ATSC	=	Advanced Television Systems Committee
CAT	=	Conditional Access Table
CBR	=	Constant bitrate
DTS	=	Decode Time Stamp
DVB	=	Digital Video Broadcasting
HAS	=	HTTP Adaptive Streaming
HLS	=	HTTP Live Streaming
IPTV	=	Internet Protocol TV
MPEG-DASH	=	Dynamic Adaptive Streaming over HTTP
MPEG2-TS	=	MPEG-2 transport stream (TS)
OTT	=	Over-the-top
PAT	=	Program Association Table
PCR	=	Program Clock Reference
PES	=	Packetized elementary stream
PID	=	Packet identifier
PMT	=	Program Map Table
PSI	=	Program-specific information
PTS	=	Presentation Time Stamp
STC	=	System Time Clock
T-STD	=	Transport System Target Decoder
VBR	=	Variable bitrate

REFERENCES

[1] N. Bhuvaneshwari and A. Aryaputra. IPTV — the next generation television. In *Proc. IEEE 2nd Global Conf. Consumer Electronics (GCCE)*, pages 509–513, October 2013.

[2] Lorin Hitt and Prasanna Tambe. Broadband adoption and content consumption. *Information Economics and Policy*, 19(3):362–378, 2007.

[3] L. Claudy. The broadcast empire strikes back. *IEEE Spectrum*, 49(12):52–58, December 2012.

[4] ISO/IEC. *ISO/IEC 13818-1:2000, Information Technology - Generic Coding of Moving Pictures and Associated Audio Information.* ISO/IEC, 2013.

[5] Inc. Advanced Television Systems Committee. *A/53: ATSC Digital Television Standard, Parts 1-6.* Advanced Television Systems Committee, Inc., 2007.

[6] ETSI EN 300 744. *Digital video broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*, pages 2004–11, 2004.

[7] Roger Pantos and William May. HTTP Live Streaming. 2016.

[8] Adobe HTTP Dynamic Streaming. 2016.

[9] ISO/IEC. *ISO/IEC 23009-1:2014, Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats*. ISO/IEC, 2014.

[10] A. Begen, T. Akgul, and M. Baugher. Watching video over the web: Part 1: Streaming protocols. *IEEE Internet Computing*, 15(2):54–63, March 2011.

[11] Anthony Vetro, Charilaos Christopoulos, and Huifang Sun. Video transcoding architectures and techniques: an overview. *IEEE Signal processing magazine*, 20(2):18–29, 2003.

[12] Jun Xin, Chia-Wen Lin, and Ming-Ting Sun. Digital video transcoding. *Proceedings of the IEEE*, 93(1):84–97, 2005.

[13] 2017 Global Media Format Report. Technical report, encoding.com, 2017.

[14] ISO/IEC. *ISO/IEC 23001-7:2016, Information technology MPEG systems technologies Part 7: Common encryption in ISO base media file format files*. ISO/IEC, 2016.

[15] Sandvine. Global Internet Phenomena Report 2016. *Sandvine Intelligent Broadband Networks*, 2016.

[16] Lajos L Hanzo, Peter Cherriman, and Jurgen Streit. *Video compression and communications: from basics to H. 261, H. 263, H. 264, MPEG4 for DVB and HSDPA-style adaptive turbo-transceivers*. John Wiley & Sons, 2007.

[17] Huifang Sun, W. Kwok, and J. W. Zdepski. Architectures for mpeg compressed bitstream scaling. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(2):191–199, Apr 1996.

[18] T. Vermeir, J. Slowack, G. V. Wallendael, P. Lambert, and R. V. d. Walle. Low delay complexity constrained encoding. In *2016 Data Compression Conference (DCC)*, pages 635–635, March 2016.

[19] Tao Lin. Achieving re-loss-free video coding. *IEEE Signal Processing Letters*, 16(4):323–326, 2009.

[20] Y. Chen, W. He, Y. Hua, and W. Wang. Compoundeyes: Near-duplicate detection in large scale online video systems in the cloud. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[21] Xi Chen, Ping Zhou, Xinxing Jing, and Juntao Shi. The video water marking scheme based on mpeg-4 motion vector. In *2010 International Conference on Intelligent Computing and Integrated Systems*, pages 189–192, Oct 2010.

[22] ETSI TR 101 290. *Digital video broadcasting (DVB); Measurement guidelines for DVB systems*, 2014.

[23] Dan Tsafrir, Yoav Etsion, Dror G Feitelson, and Scott Kirkpatrick. System noise, os clock ticks, and fine-grained parallel applications. In *Proceedings of the 19th annual international conference on Supercomputing*, pages 303–312. ACM, 2005.

[24] Mario Montagud, Fernando Boronat, Hans Stokking, and Ray van Brandenburg. Inter-destination multimedia synchronization: schemes, use cases and standardization. *Multimedia systems*, 18(6):459–482, 2012.

[25] Andrew Mason and Richard Salmon. Factors affecting perception of audio-video synchronization in television. In *Audio Engineering Society Convention 125*. Audio Engineering Society, 2008.

Josu Gorostegui Josu Gorostegui is with the Department of Digital Media & Broadcasting Technologies, Vicomtech-IK4. He received his Telecommunication Engineering degree in 2015 from the School of Engineering of the University of Navarra in San Sebastin, Spain. He started developing video monitoring components in HDR using GPGPU technologies. Currently he is on Vicomtech-IK4, where he designs and develops projects around Media and Broadcasting Technologies. His work strech over all aspects of multimedia, with a strong focus on the fields of coding, transmission and synchronization of multiple multimedia flows in different protocols. In addition, he actively participates in widely used open source frameworks.

Angel Martin Angel Martin is with the Department of Digital Media, Vicomtech-IK4. He received his engineering degree (2003) from University Carlos III. He is working in his PhD. degree in the field of video streaming. He collaborated with Prodys developing a standard MPEG-4 AVC/H.264 codec for DSP (2003-2005). He started to work on Telefonica going deeper into image processing area in terms of 3D video and multiview coding (2005-2008). He worked in Innovalia as R&D Project consultant related with smart environments and ubiquitous and pervasive computing (2008-2010). Currently he is on Vicomtech-IK4 managing and developing R&D projects around multimedia content services.

Mikel Zorrilla Dr. Mikel Zorrilla is with the Department of Digital Media, Vicomtech-IK4 since July 2007. He received his Telecommunication Engineering degree in June 2007 from the University of Mondragon (Spain), an advanced degree in Computer Science in June 2012 from the University of Basque Country (Spain), and the PhD degree in September 2016 from University of the Basque Country (Spain) entitled Interoperable Technologies for Multi-Device Media Services. Before joining Vicomtech-IK4, he has held positions at Ikerlan S. Coop. as assistant researcher (2002-2007) in the field of Transport of Multimedia Traffic. Currently, he is the head of the Digital Media department. He has been the technical and scientific manager of MediaScape European project (www.mediascapeproject.eu) and had participates in several European and local research projects, such as the Hbb4All European project (www.hbb4all.eu), where accessibility for media services from Connected TVs was addressed. In 2014 he has been an associate professor in the double degree MBA (Master in Business Administration) & Computer Engineering in Deusto Business School in the field of Media Technologies.

Iñaki Alvaro Iñaki Alvaro was born in San Sebastian, Spain, in 1966. He received his degree in Computer Science Engineering from the Universidad del Pas Vasco U.P.V (Spain) Euskal Herriko Unibertsitatea E.H.U (Spain) in 1989, and a postgraduate degree in Science and Technology of Computers in 1991, in the same university. He initiated his professional career in the R&D Multimedia department at IKUSI-Angel Iglesias in 1992. He is currently R&D Architect and Project Manager at IKUSI-Angel Iglesias. He has been involved during these years in the development of several products for the Digital TV, Video and Audio processing, IPTV markets. Nowadays, his research activity lies in the emerging applications and technologies used by OTT (Over the Top) solutions, to distribute video and audio through Internet.

Jon Montalban Jon Montalban received the M.S. Degree and PhD in Telecommunications Engineering from the University of the Basque Country (Spain), in 2009 and 2014, respectively. Since 2009 he is part of the TSR (Radiocommunications and Signal Processing) research group at the University of the Basque Country, where he is currently a postdoctoral researcher involved in several projects in the Digital Terrestrial Television broadcasting area. His current research interests include digital communications and digital signal processing for mobile reception of broadband wireless communications systems in 5G.