

Document downloaded from:

<http://hdl.handle.net/10251/154815>

This paper must be cited as:

Marfil-Reguero, D.; Boronat, F.; Montagud, M.; Sapena Piera, A. (2019). IDMS solution for hybrid broadcast broadband delivery within the context of HbbTV standard. IEEE Transactions on Broadcasting. 65(4):645-663. <https://doi.org/10.1109/TBC.2018.2878285>



The final publication is available at

<https://doi.org/10.1109/TBC.2018.2878285>

Copyright Institute of Electrical and Electronics Engineers

Additional Information

"© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works."

IDMS solution for hybrid broadcast broadband delivery within the context of HbbTV standard

Dani Marfil, Fernando Boronat, *Senior, IEEE*, Mario Montagud, Almanzor Sapena

Abstract— Inter-Destination Media Synchronization (IDMS) is a key requirement to enable successful networked shared media experiences between remote users. This paper presents an adaptive, accurate and standard-compliant IDMS solution for hybrid broadcast and broadband delivery. Apart from providing multi- and cross-technology support, the presented IDMS solution is able to accomplish synchronization when different formats/versions of the same, or even related, contents are being played out in a shared session. It is also able to independently manage the playout processes of different groups of users. The IDMS solution has been integrated within an end-to-end platform, which is compatible with the Hybrid Broadcast Broadband TV (HbbTV) standard. It has been applied to Digital Video Broadcasting – Terrestrial (DVB-T) technology and tested for a Social TV scenario, by also including an ad-hoc chat tool as an interaction channel. The results of the conducted (objective and subjective) evaluations prove the satisfactory behavior and performance of the IDMS solution and platform as well as in terms of the perceived Quality of Experience (QoE).

Index Terms— Broadcast, DVB, HbbTV, IDMS, Media Synchronization, QoE, Shared Media Experiences, Social TV.

I. INTRODUCTION

MEDIA delivery and consumption are becoming increasingly heterogeneous and ubiquitous. On the one hand, media services have been traditionally conceived for being used through specific types of delivery technologies and consumption devices. Probably, the most common example is the reception of broadcasted TV content via aerials and their playout on TV sets. However, the delivery of the same, or related, contents, possibly with different formats or versions, via diverse (broadcast and broadband) delivery technologies, and their consumption on different devices are becoming commonplace. This evolution allows leveraging the available resources at users' disposal for consuming media, providing more adaptive and ubiquitous media services.

This work has been funded, partially, by Generalitat Valenciana, Investigación competitiva proyectos, under the R&D program “Grants for research groups to be consolidated, AICO/2017”, in project with Ref. AICO/2017/059.

Authors are with the Universitat Politècnica de Valencia, Immersive and Interactive Media R&D Group, Communications Department, 46730, Grao de

On the other hand, we are witnessing a transition from physical togetherness towards networked togetherness around media content consumption. Traditionally, family members and friends have gathered at common locations for consuming media together. An example of a typical scenario is a group of friends watching a football match or a TV show at a friend's home.

Unfortunately, a myriad of practical issues prevents many times people from physically meeting up. People can move to different locations for studying, working or any other purpose. The recreation of such shared media experiences while being apart can be possible thanks to the latest advances in delivery technologies, the ever-increasing connected media ecosystem, and the proliferation and massive adoption of social networking (e.g., Social Media and conferencing services). For instance, the same friends in the above example, but being each one in remote locations, can now watch the football match not in an isolated manner, but being able to converse, discuss about it and cheer goals together (Figure 1).



Figure 1: Networked shared media experiences

The presented use case within the TV context is commonly known as Social TV. In addition, apart from Social TV, other forms of shared media experiences are gaining momentum, such as e-learning, tele-work and Multiplayer Online Games (MOG)¹.

However, many challenges must still be faced in order to successfully provide satisfactory networked shared media experiences, such as dynamic community building, media synchronization (sync hereafter), Quality of

Gandia, Valencia, SPAIN (e-mails: {damarre@dcom, fboronat@dcom, mamontor@, alsapie@mat}.upv.es).

¹ Global IP traffic will triple by 2021, mostly by Online Gaming and Video <https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>

Service/Experience (QoS/E), scalability, presence awareness, privacy concerns, and social networking integration [1]. This work focuses on a very relevant one: media sync and, concretely, on Inter-Destination Media Synchronization (IDMS). The need for IDMS to compensate for the existing delay differences between destinations in actual delivery networks and scenarios has been identified in previous works (e.g., [2, 3, 4, 5, 6]). Indeed, these delay differences may originate a temporal un-alignment between the involved playout processes in the shared session. It, in turn, can result in incoherent interactions between the users, and their frustration (e.g., a friend being aware of a goal via the chat channel, before watching and hearing the related actions on the local device). Up to 20 use cases in which IDMS becomes beneficial and/or essential are analyzed in [2].

Given the increased relevance of IDMS, a variety of solutions to accomplish it have been devised up to date (reviewed in Section 2). Notwithstanding the progress in this context, the existing IDMS solutions still present important limitations:

- Mostly based on the use of proprietary protocols and techniques. However, as discussed in [3, 7], the availability of at least standard-compliant, solutions provides many benefits and is much preferable.
- Exclusively focused on broadband environments. On the one hand, the use of broadcast technologies for media delivery and consumption is still dominant, so it is the most appropriate approach to reach a huge audience [8, 9]. On the other hand, previous studies [5, 10] have demonstrated that significant delay differences between destinations exist in broadcast scenarios.
- Conceived for being applied by using single broadband delivery technologies. However, in the current heterogeneous media ecosystem, contents can be delivered via a plethora of broadband and broadcast technologies.
- Provide support for scenarios in which the involved users are playing out the same media contents. A key benefit of this is to enable the selection of specific variants of these contents, based on the available resources and/or users' preferences.
- Mainly developed and tested in simulation frameworks or in simple real environments. However, few solutions have been integrated and evaluated in realistic, close to commercial, settings.

This work presents an adaptive, accurate and standard-compliant IDMS solution that overcomes all these limitations and exploits these emerging opportunities. In particular, the presented IDMS solution is able to synchronize the playout of media contents delivered via different hybrid (broadcast and/or broadband) technologies.

The IDMS solution has been developed and integrated within an end-to-end platform for the delivery and synchronization of hybrid contents, both on single- and multi-device scenarios [12]. This platform is compatible with the latest release of

HbbTV [11] (V2.0.1) and includes all the required steps along the end-to-end media chain, such as encoding, encapsulation, segmentation, storage, modulation, delivery, reception, tuning, processing and playout.

In order to evaluate the IDMS solution the Social TV use case has been selected. It is a highly valued use case, with high demand and commercial potential [6]. On the one hand, the results of the objective evaluation prove the consistent behavior and performance of the IDMS solution (and the platform) for the Social TV use case. On the other hand, the results of the subjective evaluation reveal very satisfactory Quality of Experience (QoE) levels.

The remainder of this paper is structured as follows. In Section II, the state of the art regarding related media synchronization solutions and Social TV studies is reviewed. In Section III, the designed IDMS solution is presented, with all its involved entities, functionalities and modules. In Section IV, the integration of the designed IDMS solution within the aforementioned HbbTV compliant platform is described. In Section V, the implementation of a Social TV use case within the platform is described. In Section VI, some results from the objective and subjective evaluations are provided. Finally, in Section VII the conclusions and some future work are discussed.

II. RELATED WORK

This section firstly reports on the magnitudes of delay differences that occur when delivering media contents, according to measurements conducted in previous studies. Then, it provides an overview of relevant IDMS and timestamp based solutions within the context of this work. Finally, related studies about Social or Interactive TV, for which IDMS becomes essential, are reviewed as well.

A. Delay Differences in Media Delivery

Different factors and system components along the end-to-end media delivery chain can contribute to the increase of delays and of their variability. An analysis of the sources of these delays and their impact can be found in [2] (for broadband delivery) and [5] (for broadcast delivery).

Likewise, the involved receivers would not simultaneously play out the same Media Units (MUs) of specific media components in a shared media session. All these delay differences, which can further increase when different sources or providers send the involved media components, reflect the need for different types of media synchronization (sync, hereafter) [6].

On the one hand, regarding broadband delivery, the International Telecommunications Union (ITU-T) G.1050 standard [13] reports on typical values of delays and jitter in Internet. It is indicated that network delays typically range between 20 and 500ms, while jitter values range between 0 and 500ms. In [14], it is reported that end-to-end delays when using different video-conferencing systems in a variety of scenarios

can range between a few tens of ms to more than 300ms. In [4], it is pointed out that end-to-end delay differences between receivers in an Internet Protocol Television (IPTV) scenario can be larger than 6s, with delays ranging between 250ms and 6500ms. In [10], it is also shown that end-to-end delay differences of up to 8s in worst case scenarios (i.e., when comparing the reception of the same content but in SD and HD) can occur.

On the other hand, as shown in Table 1, different broadcast technologies influence on the resulting end-to-end delays.

TABLE I. DELAY DIFFERENCES BETWEEN BROADCAST TECHNOLOGIES [10]

Technologies	~ Delay Differences
DVB-T vs analog	Up to 1.24s (variability of up to 0.7s)
DVB-T vs DVB-C	Up to 2.75s (variability of up to 0.2s)
DVB-T vs DVB-H	Up to 3.4s (variability of up to 0.1s)
DVB-T vs DVB-S	Up to 1.55s (variability of up to 0.3s)

Similarly, the study in [5] also provides measurements about the magnitudes of end-to-end delay differences for different TV setups in specific receivers. Delay differences between different TV broadcasts in a national scenario can accumulate up to almost 5s, while they can accumulate up to 6s in an international scenario. In addition, significant delay differences between receivers when using the same TV delivery technology, subscription type, setup combination and equipment were also detected in [5].

Delay differences when delivering media contents using broadcast technologies compared to when using broadband technologies have been also reported in [4] and [10]. The measurements in [10] reflect that delays when using broadband technologies can be up to 8s higher than when using broadcast ones. The measurements in [4] indicate that these differences can accumulate up to 72s.

B. Media Synchronization and IDMS Solutions

1) Overview of Media Synchronization

Many media sync solutions have been proposed for a variety of networked environments and applications [15]. [2] and [16] provide a taxonomy of existing media sync solutions, while [17] provides a historical review. Recently, an overview of standards for inter-media sync, Inter Device Sync (IDES) and IDMS is provided in [7]. Furthermore, the most relevant standard technologies and solutions for hybrid sync and IDMS, focusing on HbbTV [11], are reviewed in [12].

2) Timelines for Temporal Alignment of Media Streams

Typically, media delivery technologies rely on the insertion of *intrinsic* and relative timelines into the media streams, which are commonly obtained from local clocks (Figure 2). These timelines are useful to provide intra- and inter-media sync between components (e.g., audio and video) within the same stream, but their value has no signification outside the media included in these streams. The work in [18] reviews how clock references and timestamps are conveyed within different MPEG and DVB standards.

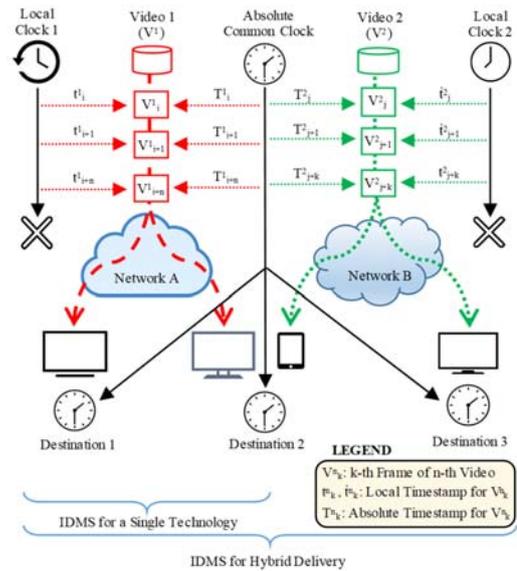


Figure 2: Global and Common Timelines in Media Sources

An important drawback of this approach is that these intrinsic and relative timelines can be overwritten by networking equipment (e.g., multiplexers, transcoders, splitters...) throughout the end-to-end delivery chain, without the content providers and clients being aware of this change. This may negatively affect the sync process. Moreover, it is very difficult to synchronize the playout of multiple media components generated by different sources or providers by using these kinds of timelines, as they are typically obtained from different unrelated local clocks, and a mapping between them may not be possible. The insertion of *extrinsic*, absolute and stable timelines related to pieces of media contents and/or to the intrinsic and relative timelines can solve the mentioned limitations (Figure 2). These absolute timelines can be obtained from the same or traceable (wall-)clock sources, such as, e.g., from Network Time Protocol (NTP) [19] servers. This approach is also the one followed when using RTP/RTCP streaming. Indeed, some other solutions based on this approach have been proposed. First, the European Telecommunications Standards Institute (ETSI) presented a timeline mechanism [20], which allows inserting an external and absolute timeline information into MPEG2-Transport Streams (MPEG2-TS).

A mechanism called TEMI was proposed by MPEG and DVB as an amendment to ISO/IEC 13818-1 [21]. TEMI mechanism enables the signalling and sync of external enhancements of programs carried over MPEG2-TS. In particular, TEMI provides the following main features [21]:

- Carrying timing information that can be used to synchronize external data.
- Signalling the location of external data that can be synchronized with the program.

These two types of metadata are commonly known as TEMI *timeline* and *location* descriptors, respectively. This solution has been adopted by HbbTV [11] so it will also play a key role in this work, as described in Section 3 onwards.

3) Existing IDMS Solutions

A variety of IDMS solutions have been devised up to date to meet specific requirements. Up to 20 use cases needing IDMS are analyzed and qualitatively ranked according to their sync requirements in [2]. In [4], it is stated that the allowable asynchrony limits for interactive services requiring IDMS may vary between 15ms and 500ms, depending on the type of service. Next, a review and classification of existing IDMS solutions is provided in Table 2 (users can find exhaustive reviews for the existing solutions in [2, 16, 22]).

TABLE II. CLASSIFICATION OF EXISTING IDMS SOLUTIONS (ADAPTED FROM [2])

Reference	Technology	Targeted Use Case
[23, 24, 25]	Proprietary protocol over UDP	CoD / Conferencing / Games / Multi-Sensory 3D Virtual Environments
[26, 27]	Proprietary protocol over UDP	Games in [26], Social TV in [27]
[22, 28, 29]	RTP/RTCP	Social TV, Conferencing, e-learning, tele-surveillance...
[30]	DASH + P2P Overlay over UDP	(Web-based) Social TV
[31]	HTML5 + WebSocket	(Web-based) Social TV

In [23], the Virtual Time Rendering (VTR) algorithm, which provides support for intra- and inter-media sync, is extended to provide support for IDMS in Content on Demand (CoD) services. Since then, the VTR algorithm has been extended and improved to adopt different communication models, and to add support for other use cases in a variety of scenarios, such as live conferencing [24], networked games and multi-sensory 3D virtual environments [25].

In [26], an IDMS solution for networked games is proposed. It is comprised of two algorithms: i) “*local lag*”, used to compensate for short term inconsistencies and ii) “*time warp*”, used to fix inconsistencies that may still occur due to uncontrollable factors. In [27], such algorithms are adapted for their application in Social TV scenarios.

In [28] and [29], the capabilities of RTP/RTCP [32], are extended to provide support for IDMS. The solution in [28] has been implemented and evaluated in [28, 33]. The solution in [29], has been evaluated in [22].

In [30], a client-centric approach to provide IDMS in over-the-top (OTT) HTTP streaming using DASH is presented. This approach adopts a distributed communication model in which the clients within a session build an application-layer peer-to-peer (P2P) overlay over UDP for exchanging timing information and negotiating on a reference timing to synchronize with. The approach in [30] also includes an Adaptive Media Playout (AMP) technique, which considers the temporal distortion of audio and video to determine the most appropriate instants/periods to perform the playout adjustments.

In [31], a web-based framework for distributed media consumption and social interaction between users is presented. It includes a centralized IDMS solution, in which the timing messages are exchanged by using WebSocket technology [34].

C. Social or Interactive TV Studies

Previous studies (e.g., [35, 36]) have investigated and highlighted the benefits that Social TV scenarios can provide to users (e.g., a feeling of being together, improved sense of connection and improved relationships). However, the availability of communication channels between the users and of IDMS functionalities have been also identified in [35, 37] and in [38, 39], respectively, as key requirements to successfully provide these benefits.

Regarding the availability of chat communication tools, text chat can result in distraction for users with low typing ability, who prefer using voice chat tools [40]. However, text chat is preferred by skilled users, as they can chat and watch/hear simultaneously, without the interference of other participants’ voices. The study in [41] also concluded that text chat is the preferred communication modality, as there is no need to respond immediately and users feel it takes less energy. In [39], it was also highlighted that both text and voice chat provide high levels of togetherness.

Regarding the relevance of IDMS, it was found in [35] that the interaction between remote users tend to happen during silence periods and/or scene changes.

In this context, controlled experimental setups have analyzed the effect of the existence of different delay difference levels on the QoE in Social TV scenarios, in which remote users interact via text and/or voice chat [39, 42]. In [39], results indicate that asynchrony levels up to 1s might not be perceptible by users while communicating using audio chat. However, asynchrony levels above 2s generally become annoying, regardless of the chat modality. Similar results were obtained in [42].

A subjective quality assessment to determine the tolerable asynchrony levels between interactive media contents being played out by remote users is conducted in [43]. It is argued that asynchrony levels of 400ms do not have an impact on the QoE, but levels of 750ms are already noticeable and can degrade it.

These previous studies consider the use of chat tools running on the same device on which the TV-related contents are presented. However, the use of companion devices is becoming commonplace. In such scenarios, IDMS becomes essential to time align the presentation of chat messages with the contents played out on the involved devices. Works in [44] and [45] present prototypes implementing the DVB Companion Screens & Streams (DVB-CSS) specification [46], which has been adopted by HbbTV [11] to enable a synchronized playout between a main TV and companion devices. In particular, the work in [45] determines that asynchrony levels between -500ms and 1000ms are unlikely to be noticed. Moreover, it is shown that users get more distracted by the companion device’s contents when the asynchrony becomes higher.

Finally, the study in [6] provides many interesting statistics and insights about the Social TV use case. From more than 1000 participants that took part in the study, around 80% showed interest in Social TV scenarios and more than 20% declared having participated in these kinds of networked shared

experiences before. More than 90% of participants declared having perceived delay differences in these scenarios, and considered this issue as a barrier for satisfactorily watching TV with other remote users. Most of the participants stated that the availability of IDMS solutions is very relevant and felt that better solutions are still necessary. Regarding the chat modality, 54% declared that text chat is their preferred option. Finally, many benefits of Social TV were identified, such as the feeling of togetherness, socialization and increased engagement.

D. Discussion

From the previous reviews, several issues must be remarked. First, delay differences are much larger than the tolerable asynchrony thresholds in the currently identified IDMS use cases [2], including Social TV. This motivates the need for IDMS solutions to compensate for these differences.

Second, many relevant IDMS solutions are uniquely focused on broadband technologies. Moreover, they have only considered situations in which all the involved users are playing out the same contents. Therefore, IDMS solutions for broadcast and hybrid cross-technology environments are needed.

Third, most of the existing Social TV studies have been conducted in controlled environments. These studies have also provided very relevant results regarding the tolerable asynchrony limits, preference regarding chat modalities and benefits. However, users still believe that better technological solutions and services are needed for Social TV [6]. Therefore, an extension to these studies in realistic scenarios would be very beneficial. The subjective evaluations in the referenced studies have been conducted for only two participants at a time, which is a realistic scenario but might be not enough to prove and validate a Social TV scenario. In this paper, we conduct subjective evaluations with groups of four remote participants.

This work focuses on overcoming the above limitations within the context of IDMS and on providing a realistic testbed for conducting further studies in Social TV scenarios.

III. IDMS SOLUTION FOR HYBRID DELIVERY

This section describes the newly designed IDMS solution for hybrid media delivery, including all its components, the interactions between them, as well as the required processes to achieve the sync goal for independent groups of users in shared media sessions.

The key aspects that have been taken into account for the design of the solution, are: i) mechanisms for time-aligning the presentation of media contents; ii) control schemes and involved entities; iii) protocols and communication models to exchange the necessary information to achieve IDMS; iv) strategies for choosing the reference timing; and v) strategies for adjusting the media playout processes according to that reference in order to adapt to any asynchrony variability in an autonomous way.

The proposed solution can be applied to any delivery technology using MPEG2-TS as the encapsulation format, such

as Digital Video Broadcasting (DVB), Dynamic Adaptive Streaming over HTTP (DASH), HTTP Live Streaming (HLS) or Real-Time Transport Protocol/ Real Time Control Protocol (RTP/RTCP). In this work, it has been applied to DVB delivery, which is the broadcast technology adopted by the HbbTV standard [11]. Apart from multi- and cross-technology support, the presented IDMS solution is able to accomplish the synchronization goal when the involved devices are playing out different formats/versions of the same contents and/or even related contents. Both features are provided by relying on the adoption of standard mechanisms to provide absolute and traceable timelines within the delivered media streams.

A. Common Timeline Insertion

In this work, the aforementioned TEMI solution [21] has been adopted for IDMS purposes. On the one hand, the *TEMI location* descriptor allows adding URIs with key metadata for IDMS purposes (explained later). On the other hand, the *TEMI timeline* descriptor allows inserting absolute timelines into the media streams, regardless of the delivery technology in use, and of the media modality and format of the delivered contents.

B. Control Scheme & Involved entities

1) IDMS Control Schemes

According to [22], three main control schemes have been used for IDMS purposes: two centralized (Master/Slave, or M/S Scheme and Synchronization Maestro Scheme or SMS) and a distributed one (Distributed Control Scheme or DCS). In the proposed IDMS solution, SMS has been adopted as it provides many advantages compared to the other schemes (as discussed in [2] and [22]). However, broadcast environments are unidirectional, and do not provide session management functionalities. Therefore, as bidirectional communications and session management functionalities are requirements for group-based IDMS, the availability of a session manager becomes necessary. However, a distributed communication model could have also been adopted, as e.g. in [22, 30], without major modifications. The interactions between the involved entities in the proposed SMS-based IDMS solution are shown in Figure 3.

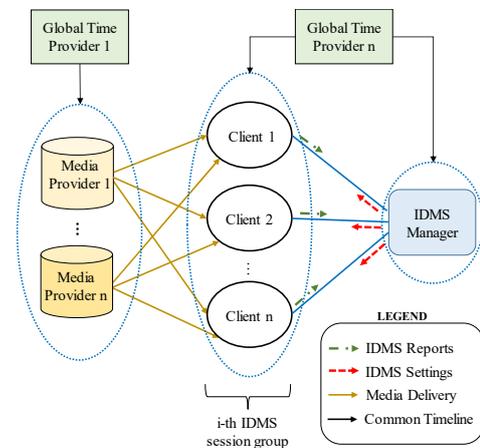


Figure 3: Involved Entities and Communication Model in the proposed IDMS solution

2) *Involved entities*

The entities involved in the sync processes and their roles are briefly described in this sub-section.

a) *Global Time Server*

There must be at least one wall-clock server entity, which provides a traceable reference clock² for all the involved entities. The involved entities can make use of a common or traceable wall-clock server to minimize the impact of clock imperfections [48], allowing for clock alignment and accurate delay measurements. As it can be seen in Figure 3, more than one wall-clock server reference can co-exist. On the one hand, the same global clock or traceable clock references must be used for the insertion of common or traceable timelines in all the streams to be played out in a synchronized manner. On the other hand, the same or another global (or traceable) clock references can be used to provide a common time reference for all the receivers in each group of users sharing a media session.

b) *Media Server(s) or Provider(s)*

There can be one or multiple Media Servers or Providers delivering the same or different media contents via the same or different technology, but inserting the common timelines.

c) *Media Clients*

They are the entities that request and receive the media contents and present them in a synchronized manner. They include a module, called *IDMS Client*, which sends informative reports, called *IDMS reports*, to the *IDMS Manager* entity (explained below). This module receives sync control messages, called *IDMS Settings*, from the *IDMS Manager*, with timing information to be processed. Then, it makes the corresponding adjustments, if needed, in the Media Client's playout processes to be accurately synchronized with the other Media Clients' playout processes.

d) *IDMS Manager*

This entity receives the *IDMS Reports* from *IDMS Clients* and computes the *session asynchrony*. This is the asynchrony between the playout processes of the most advanced and the most lagged *IDMS Clients* in the session. As a result, the *IDMS Manager* can send an *IDMS Settings* message, as explained later, to the *IDMS Clients*. The *IDMS Manager* must be capable of controlling different media sessions (i.e. different groups of users sharing different media sessions). Therefore, it must be aware of every active group of *Clients*. It is in charge of computing the *session asynchrony* and taking the decision to send a playout adjustment message if needed.

C. *IDMS Control Protocol*

In this sub-section, the key aspects of the *IDMS control protocol* are presented.

² According to RFC 7273 [47], a timestamp reference clock source may be labelled "traceable" if it is known to be delivering traceable time, provided adjustments are made for differing epochs, time zones, and leap seconds.

1) *IDMS Control Messages*

Two new control messages for *IDMS* purposes have been defined. On the one hand, the *IDMS Clients* regularly send *IDMS reports* to the *IDMS Manager* (Figure 4). These *IDMS reports* include: i) the *client id*; ii) the *group id* the *Client* sending the report belongs to; iii) a *Round sequence number* (R_{seq}) containing the number of round in order to correlate the *IDMS Reports* with the *IDMS Settings* messages sent by the *IDMS Manager*; iv) the NTP-based *TEMI timestamp* of the video frame being currently played out; v) the NTP-based *presentation timestamp* for that video frame; and vi) the NTP-based timestamp of the transmission instant.

<i>Client id</i>	<i>Group id</i>	R_{seq} <i>n</i> ^o	<i>NTP-based TEMI timestamp</i>	<i>NTP-based Presentation timestamp</i>	<i>NTP-based transmission timestamp</i>
------------------	-----------------	------------------------------------	---------------------------------	---	---

Figure 4: Format of the *IDMS Report*

These reports can be sent in a unicast or multicast manner (depending on the followed *IDMS control scheme* [22]), with a configurable and adaptive transmission rate. The *IDMS Manager* registers the information contained in those reports from all the active *IDMS Clients* in all the active shared sessions. Once all the *IDMS reports* from all the active *IDMS Clients* belonging to a specific group have been gathered, the *IDMS Manager* can compute the session asynchrony. If it exceeds a so-called session asynchrony threshold, the *IDMS Manager* will send a control message, called *IDMS Settings*, to the *IDMS Clients*. This message includes the necessary timing information to make them enforcing the necessary adjustments to the playout processes to achieve *IDMS*.

These *IDMS Settings* can also be sent in a unicast or multicast manner to all the *Clients* of a group. They include the following fields (Figure 5): i) the *group id* the message is sent to; ii) the R_{seq} ; iii) the NTP-based *TEMI timestamp* of a reference video frame; iv) the target NTP-based *presentation timestamp* for that video frame. The calculation of the values of these fields will be explained later.

<i>Group id</i>	R_{seq} <i>n</i> ^o	<i>NTP-based TEMI timestamp</i>	<i>NTP-based Presentation timestamp</i>
-----------------	------------------------------------	---------------------------------	---

Figure 5: Format of the *IDMS Settings*

The R_{seq} number is used to correlate the *IDMS Settings* with new *IDMS Report*, and is increased by the *IDMS Manager* when sending a new *IDMS Settings*. This is a protection mechanism so the *IDMS Manager* can be aware that *Clients* have received and processed previous *IDMS Settings*. This number is also used to prevent the *IDMS Manager* from accepting late/outdated *IDMS Reports* from previous sync rounds. Note that both types of messages have been designed to follow a consistent syntax, in order to promote their adoption for any other *IDMS solution*.

Timestamps taken using clocks synchronised to a traceable time source can be directly compared even if the clocks are synchronised to different sources or via different mechanisms.

2) IDMS Entities Workflow and Exchange of Timing Information

a) IDMS Manager

As mentioned above, this entity is used to collect the *IDMS Reports* and compute the session asynchrony. As a result, it can send an *IDMS Settings* to the IDMS Clients, which, if needed, will perform playout adjustments to acquire IDMS. A high-level block diagram with all the modules implemented within the IDMS Manager is shown in Figure 6.

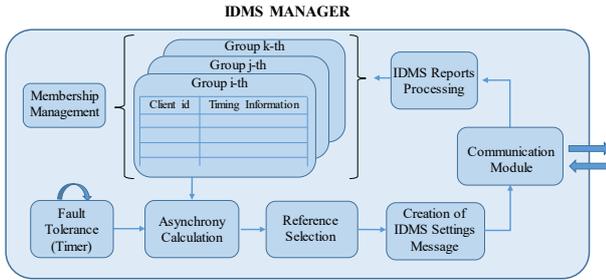


Figure 6: Modules within the IDMS Manager

When the IDMS Manager is active, it waits for *IDMS Reports*. As soon as it receives the first *IDMS Report*, a *timer* is set in order to receive the rest of *IDMS Reports* from all the other involved *IDMS Clients* within a time period called *Synchronization Round*. Every time an *IDMS Report* from an IDMS Client is received, the timing data statistics for that IDMS Client are registered.

Three situations can happen: i) all the involved Clients have sent their *IDMS Reports* before the timer's timeout; ii) the timer's timeout is triggered, but enough *IDMS Reports* from at least two different IDMS Clients have been received; and iii) the timer's timeout is triggered, but there is no *IDMS Report* information from enough Clients. In this last (and undesired) situation, the IDMS Manager deletes the received timing information and starts again, waiting for new *IDMS Reports*. In the first two situations, the IDMS Manager is able to compute a value for the session asynchrony. On the one hand, if it does not exceed a configured session asynchrony threshold, the overall timing information from that shared session is deleted, the timer is stopped and the IDMS Manager starts again, waiting for new *IDMS reports*. On the other hand, if the session asynchrony threshold is exceeded, the IDMS Manager calculates the timing of a Synchronization Reference (hereafter, Sync Reference, explained later), prepares and sends the *IDMS Settings* message to all the IDMS Clients. Then, the overall collected timing information from the *IDMS reports* is also deleted and the number of round (R_{seq}) is incremented. Once the *IDMS Settings* message is sent, the IDMS Manager will not accept any IDMS

Report from any IDMS Client during a *time guard period*. After this interval, the IDMS Manager starts again, waiting for new IDMS reports in a new round. Figures 7 and 8 show the workflow of the IDMS Manager and an example of messages exchange for normal and undesired situations, respectively.

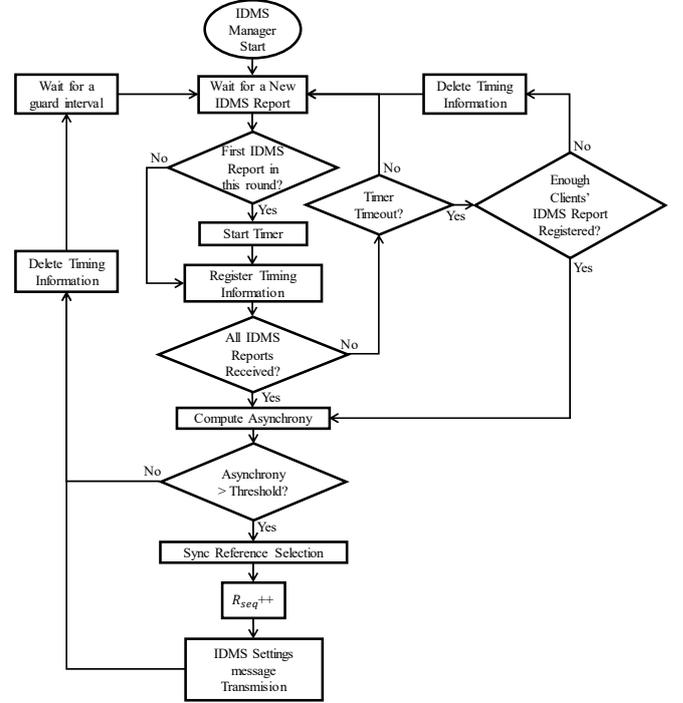


Figure 7: IDMS Manager Workflow

The concept of Synchronization Round allows the IDMS Manager to distinguish time periods between consecutive *IDMS Settings* transmission to the IDMS Clients. As they periodically send their timing information, there is a need to gather this information separately, and thus, the use of rounds for the asynchrony per session value calculation. So, a Synchronization round must last, at most, the time given by Equation 1:

$$T_{SyncRound} = 2 * P_{IDMS} + MAX(e2e_{delay_i}) \quad (1)$$

, where P_{IDMS} is the transmission period between two consecutive *IDMS Reports* sent by an IDMS Client, and $e2e_{delay_i}$ is the end to end delay from i -th IDMS Client to the IDMS Manager (both in seconds). This report interval period could also be dynamically adapted in order to overcome scalability issues and randomized for avoiding feedback implosion. By using this configuration value for the timer, we can make sure that, without losses and errors, the IDMS Manager will receive all the *IDMS Reports* from all the IDMS Clients for each round.

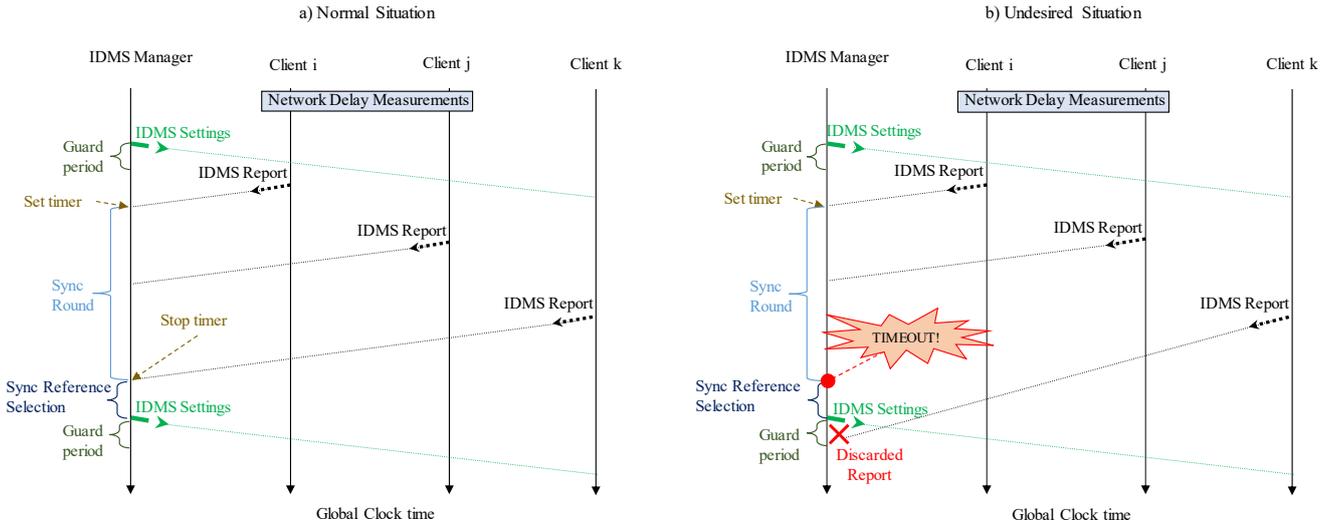


Figure 8: IDMS messages exchange for a normal situation (a) and an undesired situation (b)

b) IDMS Client

Every time an IDMS Client receives an *IDMS Settings* from the IDMS Manager, it will be able to adjust (if needed) the timing of its playout processes, to get in sync with the timing reference selected by the IDMS Manager (explained in Section III.D), and thus with the other Clients in the session. Figure 9 illustrates the workflow of the IDMS Clients when participating in an IDMS session.

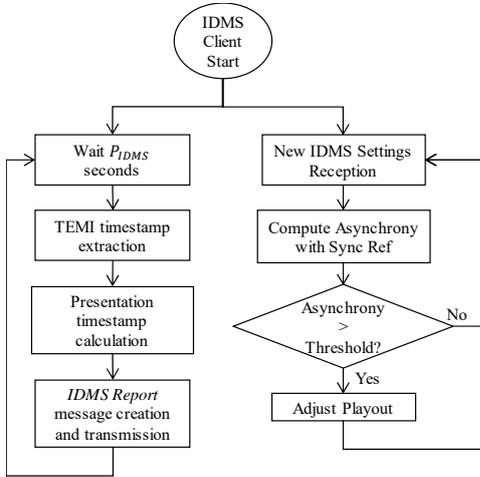


Figure 9: IDMS Client's workflow

A threshold for the asynchrony of their playout timing regarding the reference timing (hereafter, *internal IDMS threshold*) has been defined to avoid too many frequent adjustments in the playout processes of each Media Client, as thus might result in a poor QoE. This way, Clients only adapt their playout timing if that threshold is exceeded.

D. Reference Selection Strategies

Once the overall playout timing information of all the IDMS Clients of a group has been gathered, and the IDMS Manager

has checked that the session asynchrony exceeds a configured session asynchrony threshold, a new *IDMS Settings* message must be generated and sent. The first decision of the IDMS Manager consists of choosing or calculating the timing reference to synchronize with (*Sync Reference*, hereafter). Different strategies for that purpose have already been proposed and evaluated in [22], such as selecting as the reference: i) the most lagged IDMS Client; ii) the most advanced IDMS Client; or iii) the mean playout point. The pros and cons of each strategy, and their implications, are discussed in [2, 22]. In particular, the proposed IDMS solution is able to support any of them, as buffering mechanisms can deal with any of the selected reference and these tasks correspond to the implementation of the IDMS solution.

E. Media Playout Adjustments

When an IDMS Client receives an *IDMS Settings*, it must calculate the asynchrony between its playout timing and the received timing of the Sync Reference. Then, if the internal IDMS threshold is exceeded, it will adjust its playout process.

Regarding playout adjustment techniques, two options can be adopted [49, 50]. On the one hand, aggressive adjustment techniques, such as skips and pauses, are employed when the asynchrony is too high (i.e., it exceeds another configured threshold) and every time the Media Client joins the session, or a new version of the media content is selected. This is done to achieve a nearly immediate sync. On the other hand, AMP techniques can be employed when the asynchrony is within (configurable) specific limits. AMP consists of smoothly adjusting (i.e., fasting up / slowing down) the playout rate to minimize asynchrony situations. It allows achieving higher sync accuracy and avoiding long-term playout disruptions, which can be annoying to the user's perception thus degrading the QoE, as proved in [49, 50].

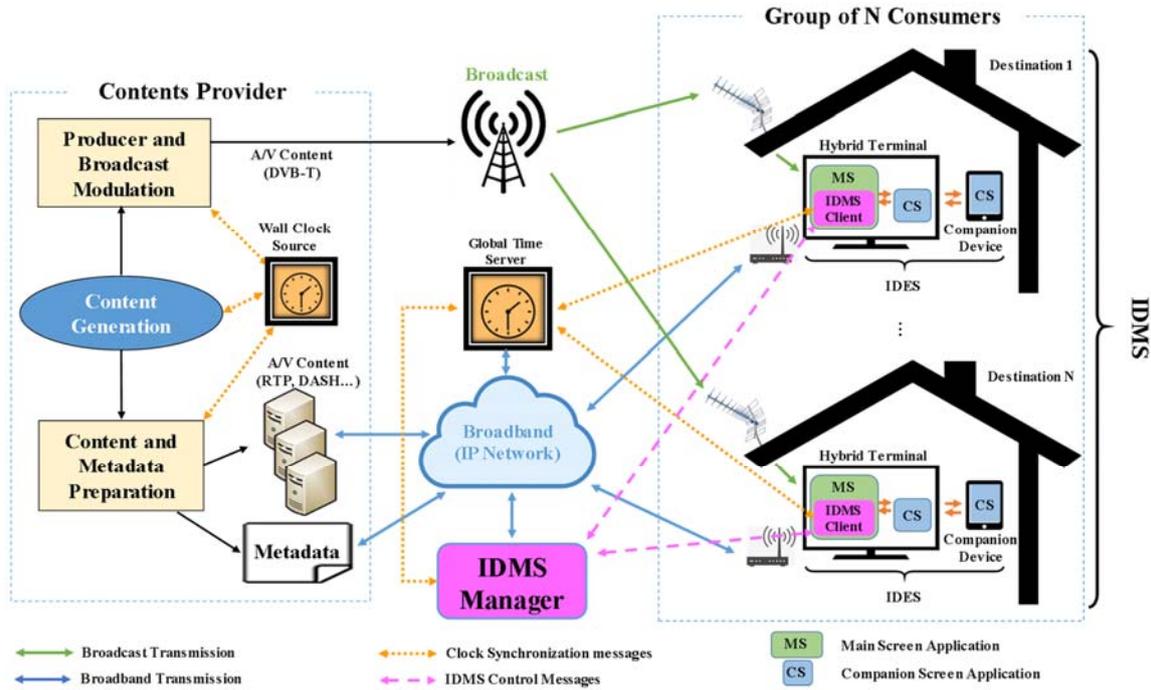


Figure 10: Overview of the platform including IDMS-related modules

IV. INTEGRATION WITHIN THE END-TO-END HbbTV-COMPLIANT PLATFORM

The designed IDMS solution has been integrated within an end-to-end HbbTV-compliant platform for the preparation, delivery and synchronized consumption of related hybrid media contents on single devices and/or on multiple close-by devices, implemented by the authors and presented in [12]. The platform has been developed by mainly using the GStreamer framework [51], in both Linux- and Android-based devices.

In this work, the platform presented in [12] is used but it has been extended to provide support for IDMS (i.e. inter-destination or group sync), in addition to cross-content and cross-technology support and the design of a synchronized chat. The main contributions of the work in [12] were the platform, and their features focused in achieving inter-stream media sync (between hybrid media streams played in a single terminal) or IDES (between hybrid media streams played in different -local-close-by devices). In [12] synchronization or communication mechanisms between geographically dispersed devices were not considered.

This platform includes the necessary requirements to be compatible with HbbTV standard and, besides, adds an extra functionality as IDMS can also be achieved under these HbbTV-based requirements (i.e., the use of WebSockets and XML parsers are already in-use tools in SmartTV/HbbTV devices.)

That platform is mainly comprised of two parts: the contents provider side and the consumer side. Figure 10 provides an overview of these two parts, including the integration of the newly defined IDMS entities in dashed line boxes.

The IDMS Manager has been located independently to the Content Provider side, as third parties can also be involved for this purpose. The rest of functionalities of both sides (Contents Provider and Consumer) are extensively described in [12].

A. Contents Provider Side: IDMS Involved Modules

This part of the platform involves all the needed modules with the appropriate functionalities in order to create, encode and transmit media contents (shown in Figure 11).

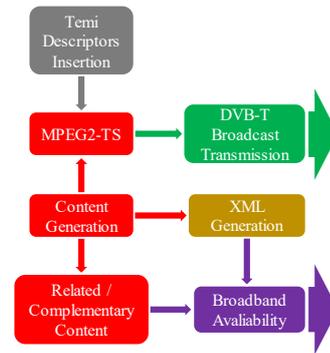


Figure 11: Modules involved in the Content Provider Side

In this part, the following steps are performed: i) firstly, the content is recorded/obtained in raw format, and then converted and encoded (e.g., to an mp4 file by using h264+aac encoding); ii) using GPAC framework [52], an MPEG2-TS is generated with the embedded TEMI descriptors; iii) an XML file is created (described in next sub-section), and its URL is included in the TEMI location descriptor; and, finally, iv) the media content is transmitted to the consumers via broadcast and/or broadband technologies. More detail can be found in [12].

1) XML-formatted metadata for hybrid media delivery and IDMS

In order to discover additional associated services, to the use of a text file is proposed. It contains relevant metadata in order to signal and locate all the heterogeneous available related content, which can be played out by consumers within the same hybrid terminal or on different companion devices. This metadata has been formatted as an XML file (specified in [12]). This file also describes and includes all the locations of the needed resources in order to successfully achieve IDMS sync (e.g., the IP address of the external clock reference server). Many different types of related content, with different formats and encoding, can be signalled in this file [12].

New tags and attributes for IDMS purposes have been defined (Table 3). They are consistent with the already existing tags and attributes in that XML file [12], in order to provide an homogeneous organization, thus facilitating the adoption of this resource for any hybrid solution. With the purpose of identifying the IDMS Manager, an *IDMS* tag is used. It contains many different properties, among which the URI is the most relevant. This is used to point to the IDMS Manager’s URL. Other properties are a unique identification value (i.e., the id property), the required protocol to interact with the specific module defined in the tag (i.e., the protocol property) and a brief description (i.e., the metadata property).

TABLE III. SUMMARY OF NEW DEFINED XML TAGS FOR IDMS PURPOSES

TAG	Property	Description
IDMS	specifies the necessary metadata for any available IDMS Manager	
	id	unique identification value for the element
	protocol	used protocol
	metadata	brief description
	uri	uniform resource identifier
CLOCK	Specifies the metadata for the global clock technology	
	id	unique identification value for the element
	protocol	used protocol
	media_type	(media) content type
	media_format	format of the time value
	uri	uniform resource identifier

Moreover, the *CLOCK* tag (defined in [12]) points to a wall-clock source (e.g. an NTP server) that could also be used for IDMS purposes. An alternative solution could also be employed. It consists of the selection of a common (or traceable) wall-clock source by the IDMS Clients when joining a session. The info about the wall-clock source in use will be received together with the confirmation of the session membership. This contributes to a higher flexibility, scalability and can also contribute to enhance the performance.

B. Consumer Side: IDMS involved Entities and Modules

All the related entities and modules which allow hybrid media consumption are located in this part of the platform (right side in Figure 10). This part has been developed by using the GStreamer framework [51], and it is in charge of the reception,

processing, decoding and presentation of the hybrid media contents.

It includes two types of entities: *Hybrid Terminals*, with broadcast and broadband interfaces, and *Companion Devices*, only with broadband interfaces. Examples of hybrid terminals are main Smart TVs or Set-Top Boxes (STB) while examples of companion devices are smartphones or tablets.

This part also includes two software modules: Main Screen and Companion Screen applications (hereafter, MS and CS, respectively). Both can be run on the hybrid terminals, and CS can also be run on companion devices. Regarding the IDMS solution, the IDMS Client’s functionalities have been integrated within the MS. Thus, the media content the MS receives and presents is the one which will be considered for IDMS purposes and is referred to as the *main content*. The remaining available (secondary) media contents, such as complementary text or additional audiovisual content, will be synchronized with the MS main content by using inter-media or IDES mechanisms, either within the same hybrid terminal or on any associated companion device (running CSs). These IDES mechanisms have been specified and evaluated with satisfactory results in [12]. The involved steps to achieve IDMS are:

- 1) The MS receives the main media content and analyses the streams in order to detect and extract the IDMS information (included in the TEMI descriptors).
- 2) From the TEMI location descriptor, the MS obtains the URL addressing the XML file with the associated metadata.
- 3) The IDMS Client included in the MSs connects to the IDMS Manager via WebSocket [34].
- 4) The IDMS Client starts sending *IDMS Reports* and receiving the corresponding *IDMS Settings* messages from the IDMS Manager.

When an IDMS Settings message is received, a new playout asynchrony value is calculated. If the absolute value of the asynchrony exceeds the internal asynchrony threshold, playout adjustments must be performed. Consequently, the fullness level of the playout buffer of the connected TVs also needs to be adjusted. The involved modules and workflow in this part of the platform can be observed in Figure 12.

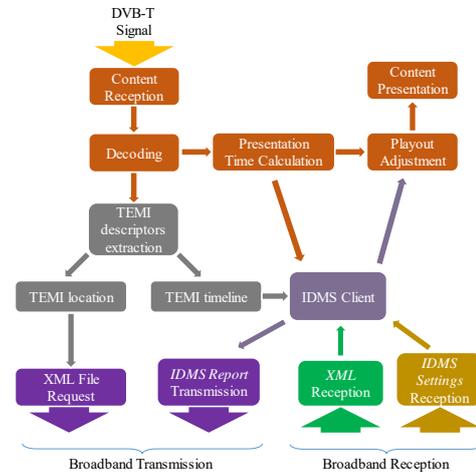


Figure 12: Modules within the Consumers’ Side

By following recommendations described in Section 13.5.3 of HbbTV spec [11], the buffer size has been configured to be capable of storing approximately 12s of content at a 20Mbps rate, which is enough to compensate for the delay differences in most actual delivery scenarios, as analyzed in [4, 5] and reviewed in Section II.

C. IDMS Manager

In the platform, the IDMS Manager has been implemented as a WebSocket-based [34] Server (WSS). It is in charge of both session management and IDMS management.

Wherever the IDMS Manager is located (inside the Content Provider side or outside), an agreement on how to provide its location to the IDMS Client in the MS app should be reached. In this case, as explained in previous sections, the location is stored in the explained XML metadata file, and the URL of this file is included in the TEMI location descriptor.

V. DEVELOPMENT OF THE SOCIAL TV USE CASE IN THE PLATFORM

In this Section, the implementation of a group-based interactive Social TV use case in the platform is described.

This Social TV functionality includes two main services: the Synchronized TV Service, providing a DVB-T stream to all the involved MS running on a hybrid terminal; and an IP-based Private and Synchronized Chat Service, allowing the interaction between all the involved users in a shared session (via CS). This new chat functionality is WebSocket-based and, although being independent, both services can share specific information in order to identify connected IDMS Clients, such as the session id or the user id (see Figure 13).

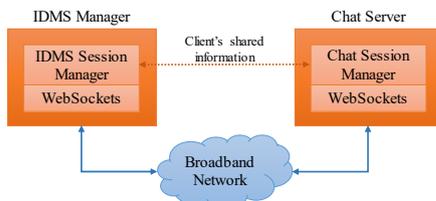


Figure 13: Chat related functionalities and IDMS Manager

An additional lightweight and accurate inter-media sync solution has been developed for time aligning the presentation of text messages with the associated TV-related content. The chat can be launched either on the main device or on companion devices, depending on the users' preferences. In the latter case, HbbTV [11] mechanisms to provide IDMS are also necessary, and have been adopted from [12].

By default, when making use of the developed platform, users watch TV content in the traditional passive manner. If they wish to interact with other remote users, they can create a new shared session, which enables the designed IDMS and chat functionalities to enjoy a shared watching experience. The description of the specific mechanisms to notify about the existence of active sessions, informing about the contents being

watched and the active users in each one of them, are out of the scope of this paper.

Next, the employed delivery technologies, the developed signalization mechanisms and ad-hoc text chat tool are described.

A. Synchronized Digital TV Service

For this particular Social TV use case, DVB-T media content has been selected as the main content to which IDMS is applied. It is due to the fact that it is the traditional delivery technology in TV environments. It will contain the aforementioned TEMI descriptors. The DVB-T contents will be received and processed by the MS on each involved hybrid terminal. By using the proposed IDMS solution all the distributed Media Clients will playout the DVB-T content in a synchronous way.

B. Interactive, private synchronized chat service

The ad-hoc text chat tool has been implemented within the CS module, and can be presented on hybrid terminals, on secondary devices or on both simultaneously. The discovery and association between the involved devices and services is accomplished by using Discovery And Launch (DIAL) protocol [53], which is the adopted protocol by HbbTV [11] for such purposes (already integrated into the platform [12]).

The chat messages are assigned an NTP-based timestamp, which refers to the current NTP-based TEMI timestamp of the video frame being watched at that time by that user, in order to enable synchronization, and thus avoiding spoilers.

Exchanged chat messages include enough information in order to identify the source and associate the related main (TV) content frame being presented when the message is sent (Figure 14).

Group ID	User ID	Main Content NTP-based TEMI timestamp	Chat Message
----------	---------	---	--------------

Figure 14: Chat message structure

Note that this spoiler-protection mechanism will not work for users whose playout process is ahead. Advanced users will be affected if any other users present lagged playout processes, as the latter will not be able to write anything related to content that has not been watched (or even received!) yet. The only way to avoid this situation is with the usage of IDMS adjustment mechanisms, as the ones proposed in this work.

1) Contextualized and Event-Driven messages interface

With the aim of providing faster and interactive communications via text chat, an interface of predefined contextualized messages has been designed and implemented. This allows users to experience more fluid conversations, as this type of predefined messages refer to actions or events that are likely to occur in the show/event being watched. The list of event-driven messages has been implemented as a new XML file (Table 4 presents the defined tags for such purpose). Chat clients can dynamically set up and embed different buttons in the chat layout with predefined messages.

TABLE IV. SUMMARY OF NEW DEFINED XML TAGS FOR THE CONTEXTUALIZED MESSAGES CHAT APP FUNCTIONALITY

TAG	Property	Description
BUTTON	id	unique identification value for the element
	btn_name	the name displayed in the button
	msg2send	the sent message once the button is pressed

2) Server side

The Chat Server forwards the incoming messages from each chat client to all the other chat clients in the same shared session. As soon as a chat message is received by the server, it checks which IDMS shared session the message comes from and forwards it to all the other members of that session.

3) Client side

Users have the option to choose to run the CS chat app either in the hybrid terminal together with the MS displaying audiovisual content (forcing to resize the video region); on any companion device or simultaneously on both devices. Figure 15 shows the appearance of the hybrid terminal with and without the CS chat app. The CS chat app running in an Android-based companion device is shown in Figure 16.

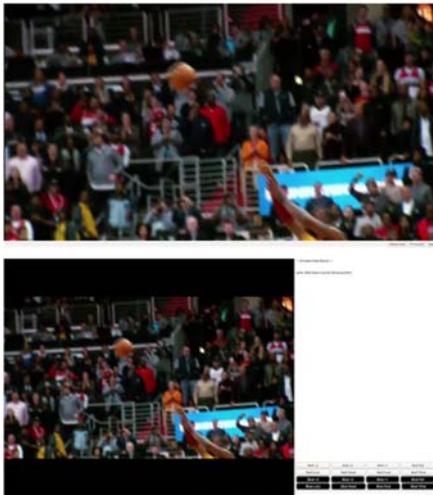


Figure 15: Hybrid terminal only with MS (top) or with both MS and CS chat apps (bottom)

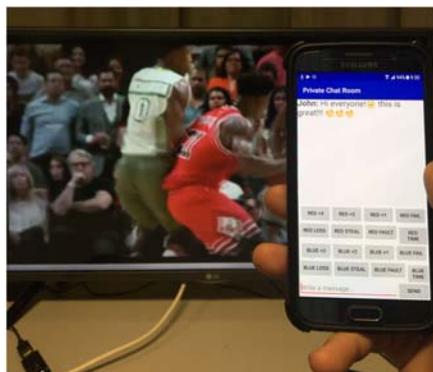


Figure 16: Chat app on a companion device and the main hybrid device for a basketball use case

C. Extensions to the XML Metadata File for Social TV

In addition to the new IDMS-related tags already described in Table 3 for the metadata XML file needed for IDMS, an additional tag has also been defined to signal the availability of the chat server for the Social TV use case (Table 5). It contains all the necessary information to create/join a certain content-related (and private) chat room.

TABLE V. NEW DEFINED XML TAG FOR THE SOCIAL TV USE CASE

TAG	Property	Description
CHAT	specifies the necessary metadata for any available Chat Server	
	id	unique identification value for the element
	protocol	used protocol
	metadata	brief description
	uri	uniform resource identifier

VI. EVALUATION

This section describes, the configured conditions and scenarios, and the conducted objective and subjective evaluations. Finally the obtained results and a brief discussion is also presented.

A. Evaluation Scenario

The evaluation scenario that has been implemented involves the devices and contents shown in Tables 6 and 7, respectively. The involved devices (Table 6) consist of four different hybrid terminals (running both the MS and CS chat apps) with their own router and USB DVB-T receiver located at different labs within the University Campus. It allows to simulate four independent homes with their own LAN environments. Moreover, and mainly focused on the subjective evaluation, different devices have been available to be used as companion devices (running the CS chat app), such as smartphones and/or tablets. Figure 17 shows the implemented scenario for the evaluations.

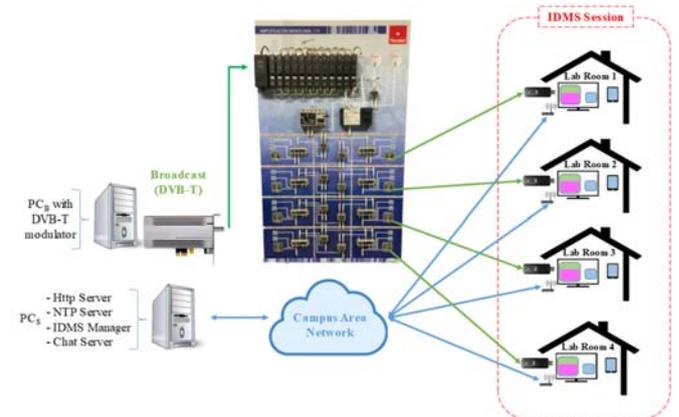


Figure 17: Implemented scenario for the evaluations

Regarding the TV content (Table 7), a basketball match, delivered only via DVB-T broadcast technology, has been used for the evaluations. From the mp4 file with h264+aac encoding, GPAC framework has been used to generate an MPEG2-TS with the embedded TEMI descriptors, as explained in Section IV (more details can be found in [12]). The selected type of

content (i.e., sports events) can be classified as high motion, in which users can perceive lower values of asynchrony. This type of content stimulates participants' interaction and communication due to its fast and engaging nature. Therefore, it has been chosen specifically for this Social TV use case in order to trigger the participants will to communicate via the chat tool.

TABLE VI. INVOLVED DEVICES IN THE EVALUATION PROCESS

Device	Features
PC _B (broadcaster)	Intel Core i7-6700 @ 3.40GHz, 8GB RAM, SSD 240GB, Windows 10
DVB-T Transmitter	DVB-T PCI Card: DekTec DTA-2111
PCs (server)	Intel Xeon E5420 @ 2.50GHz x8, 8GB RAM, Ubuntu 14.04, Apache Server v2.4.7, Fast Ethernet embedded card.
Routers	4xR: TP-Link AC1900 Wireless Dual Band Gigabit. IEEE 802.11ac.
Hybrid Terminal (Set Top Box, STB)	4xPC _H : Intel Celeron 1037U @ 1.80GHz x2, 4GB RAM, Ubuntu 14.04; Fast Ethernet embedded card; WiFi card IEEE 802.11 b/g/n
DVB Receiver	DVB-T USB card. with RTL2832u chipset
TVs x4 (monitors)	TV1: LG 32LF592U (32"); TV2: Panasonic TH42PX80E (42"); TV3 & TV4: Samsung Smart TV UE48H6200AWXXC (48").
Companion Devices	2x Samsung Galaxy Tab S (IEEE 802.11ac 10" tablet) 2x Samsung S5 (IEEE 802.11ac 5" smartphone)

TABLE VII. MEDIA CONTENTS

Technology	Stream Type	Encoding
DVB-T (MPEG2-TS)	Video	H.264, 1920x1080, 25fps (30Mbps)
	Audio	MPEG-4 AAC, 140Kbps (2 channels, 48KHz)
IP	Text	Ascii

Due to the lack of an end-to-end TV infrastructure in our University Campus signals have been modulated and transmitted via coaxial cables to the four hybrid terminals. DVB-T transmission has been set with common parameters (i.e., UHF 770MHz transmission channel, 64-QAM mapping, 8MHz bandwidth, 1/32 guard interval, 8K mode in OFDM modulation, 7/8 coding rate).

Taking into account that MPEG2-TS transport packets are usually extended by error protection mechanisms such as Reed-Solomon, convolution coding and different modulation schemes, a quasi-error free (QEF) transport of DVB services can be guaranteed, meaning that, approximately, one non-correctable error occurs within one hour of program presentation (i.e., a Bit Error Rate –BER– of 10^{-11})³. As the videos used in the evaluation are 10 minutes long, we have evaluated the scenario with a symbolic packet loss probability of 0,001% (much larger than the QEF average) in DVB reception applied for 3 out of the 4 involved MS (to emulate different broadcast conditions for each MS and configured through GStreamer plugins [51]). Figure 18 shows the playout elements structure for every MS, and where specific functionalities are implemented.

³ As stated in ETSI EN 300 744 V.1.6.2 (2015-10) Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television: "Quasi Error Free (QEF) means less than one uncorrected error event per hour, corresponding to BER = 10^{-11} at the input of the MPEG-2 demultiplexer."

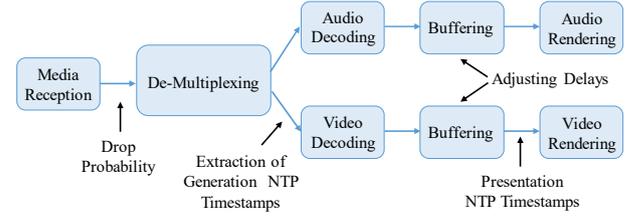


Figure 18: Main blocks of the developed player included in the MS

Moreover, in broadcast delivery, in order to simulate delay variability between receivers, the following constant delay values were artificially introduced in each client: DelayClient₁ = 0s; DelayClient₂ = 1.5s; DelayClient₃ = 3s; and DelayClient₄ = 6s. These values have been considered by following the results obtained in the study in [5], where, as indicated in Section II, the maximum observed broadcast delay accumulated up to 6s.

Regarding broadband connections, with the usage of the Netem⁴ tool, the broadband network delay behaviour was forced to $60\text{ms} \pm 20\text{ms}$ (following a normal distribution), which represents reasonable mobile networks or long-distance fixed line connections [12]. This is representative for a broad range of application scenarios. In terms of network load, *IDMS Report* messages are 133 Bytes long and *IDMS Settings* messages are 109 Bytes long (both sent via WebSocket channels [34]). These packet sizes are small enough to avoid network congestion if sent with a medium/high frequency (i.e., every 2s).

Also, internal clocks typically present skews (the clock signal arrives at different components at different times) and drifts (the clock does not run at the exactly same rate as the reference clock) along its execution [5, 48]. In order to evaluate worst case conditions in real environments, these clock imperfections have also been considered for the playout processes in the platform. The skew value used for this evaluation has been $100\mu\text{s/s}$, based on observed values in some devices [5].

Taking into account the IDMS-related packets size, it has been considered the viability of using small periods of transmission, in order to achieve accurate IDMS as soon as possible and still avoiding network congestion. With this premise in mind, 2s of *IDMS Report* interval was selected. Then, the slowest permitted threshold for the session asynchrony (without negatively affecting the platform's performance) was selected for 160ms, in order to not to exceed the 400ms recommended asynchrony threshold for Social TV use cases, as stated in [43]. Many preliminary performance test were carried out with different values and 160 ms was the smallest one that did not affect the platform's correct behavior. As results will show, the undesired asynchrony values stay lower than the aforementioned 400ms.

Two sync reference strategies have been used in the evaluations: the Most Lagged Client and the Mean Playout Point. Taking the most advanced client as the reference has

⁴ Netem provides Network Emulation functionality for testing protocols by emulating the properties of wide area networks. <https://wiki.linuxfoundation.org/networking/netem>

been discarded, as it implies that lagged clients will skip a certain number of frames in order to achieve IDMS and, consequently, users could miss significant content.

B. Objective Evaluation

The performance of the modules for presentation of TV contents and specially of the IDMS proposed solution, have been objectively evaluated in the described scenario for the Social TV use case described in Section V.

1) Methodology

The performance of the IDMS solution has been evaluated by registering the asynchrony between the TV contents at the stream level, by comparing the timelines of the video frames being played out (just before rendering, as can be seen in Figure 19) by the MS running in each hybrid terminal.

Besides, the frame numbers have been overlaid during the media encoding process via the ffmpeg framework⁵ in order to visually compare the asynchrony level of the involved Clients.



Figure 19: Visual assessment of the IDMS accuracy

The performance of the IDMS solution has been evaluated in four specific and well-differentiated cases. They are distinguished by the selected reference (to the Slowest Client or to the Mean Playout Point) and by the existence (or not) of network degradation (broadcast delivery packet loss and broadband network delay). Table 8 summarizes the four evaluated cases.

TABLE VIII. CONDITIONS FOR EACH EVALUATED CASE (1)

Case 1	Case 3
<ul style="list-style-type: none"> IDMS reference: Most Lagged Client No Broadcast Packet Loss No Broadband Network Delay 	<ul style="list-style-type: none"> IDMS reference: Mean playout point No Broadcast Packet Loss No Broadband Network Delay
Case 2	Case 4
<ul style="list-style-type: none"> IDMS reference: Most Lagged Client Broadcast Packet Loss (0.001%) Broadband Network Delay (60 ±20 ms) 	<ul style="list-style-type: none"> IDMS reference: Mean playout point Broadcast Packet Loss (0.001%) Broadband Network Delay (60 ±20 ms)

(1) Constant broadcast delay values introduced in each client: DelayClient_1 = 0s; DelayClient_2 = 1.5s; DelayClient_3 = 3s; and DelayClient_4 = 6s

2) Results

Figure 20 shows the registered asynchrony values during a media session, when no IDMS solution is applied (solid black line) and when it is applied (case 1 and 3, in red dashed and blue dotted lines, respectively) and only clock imperfections affect

the playout rate (without losses or network delay). It can be clearly observed the rising of the asynchrony values when no IDMS solution is applied (solid black line), reaching values higher than 2s after 60 minutes of playout. However, when IDMS is enabled, the asynchrony remains below the session asynchrony threshold (red and blue lines). This is why authors assume that without any IDMS mechanisms, even initially synchronized devices might quickly end up with annoying asynchrony values.

Tables 9-12 show the results for each evaluated scenario, for a configured session asynchrony threshold of 160ms. For each case, 10 sessions of 10 minutes have been run and the mean value and 95% confidence interval of their results are reported. These obtained values have been registered after each MS has overcome the initial asynchrony state, once the IDMS solution has started.

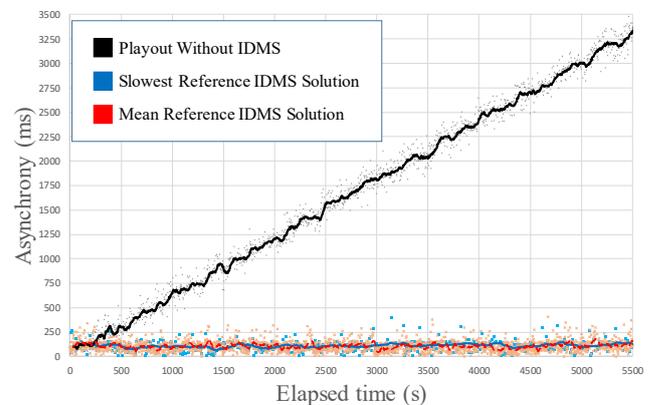


Figure 20: Session asynchrony evolution with and without applying the proposed IDMS solution⁶

Note that the number of adjustments request sent by the IDMS Manager are higher than the number of the adjustments made by the IDMS Clients. This is because after sending the *IDMS Settings* messages, the IDMS Clients decide if they have to adjust or not (as seen in Figure 9).

In cases 1 and 3, when the most lagged client is taken as the sync reference strategy, it was expected to find one of the clients as the most lagged, to which all the other clients would synchronize to. That client should not have done any playout adjustments (or much less than the other clients). Nevertheless, results show that there is not a clear most lagged Client during the session and this role is exchanged during the 10-minute session, since all the clients make some adjustments during it. It can be explained because playout adjustments are challenging in live push-based content. To achieve IDMS, the extraction of as much accurate timing data as possible through the elements in the developed GStreamer pipeline is essential. Unfortunately, it would have been too complex to probe the required metrics with higher accuracy by employing the baseline GStreamer configuration; and evolving GStreamer is out of the scope of this work. Despite these issues, the obtained accuracy is enough for our presented solution, as corroborated in this Section.

⁵ Ffmpeg is a powerful framework able to decode, encode, transcode, mux, demux, stream, filter and play pretty much anything that humans and machines have created <https://www.ffmpeg.org/>

⁶ To smooth out short-term fluctuations, lines represent a moving average, taking groups of 15 samples.

TABLE IX. CASE 1: MOST LAGGED CLIENT AS REFERENCE, NO DELAY AND NO LOSS

IDMS Session Stats				
Average mean (ms)	109.53			
Confidence Interval 95% (ms)	9.02			
Max. session asynchrony (ms)	338.52			
Min. session asynchrony (ms)	3.31			
Number of adjustment requests	33			
Involved Devices Stats				
Client 1	Client 2			
Client 3	Client 4			
Mean Square Value of Corrected Asynchronies (ms ²)	18983 (138ms)	21116 (145ms)	14101 (119ms)	19737 (140ms)
Number of performed adjustments	19	21	14	19

TABLE X. CASE 2: MEAN PLYOUT POINT AS REFERENCE, NO DELAY AND NO LOSS

IDMS Session Stats				
Average mean (ms)	112.845			
Confidence Interval 95% (ms)	4.932			
Max. session asynchrony (ms)	397.850			
Min. session asynchrony (ms)	2.97			
Mean no. of adjustment requests	30			
Involved Devices Stats				
Client 1	Client 2			
Client 3	Client 4			
Mean Square Value of Corrected Asynchronies (ms ²)	14361 (120ms)	11649 (108ms)	11507 (107ms)	10817 (104ms)
Number of made adjustments	16	15	15	14

TABLE XI. CASE 3: MOST LAGGED CLIENT AS REFERENCE, BROADCAST PACKET LOSS 0.001%, NETWORK DELAY 60 ± 20MS

IDMS Session Stats				
Average mean (ms)	152.787			
Confidence Interval 95% (ms)	4.876			
Max. session asynchrony (ms)	398.44			
Min. session asynchrony (ms)	2.68			
Mean no. of adjustment requests	30			
Involved Devices Stats				
Client 1	Client 2			
Client 3	Client 4			
Mean Square Value of Corrected Asynchronies (ms ²)	24227 (156ms)	23286 (153ms)	21216 (146ms)	22889 (151ms)
Number of made adjustments	18	18	18	19

TABLE XII. CASE 4: MEAN PLYOUT POINT AS REFERENCE, BROADCAST PACKET LOSS 0.001%, NETWORK DELAY 60 ± 20MS

IDMS Session Stats	
Average mean (ms)	153.865
Confidence Interval 95% (ms)	9.892
Max. session asynchrony (ms)	439.71
Min. session asynchrony (ms)	2.3
Mean no. of adjustment requests	29
Involved Devices Stats	
Client 1	Client 2
Client 3	Client 4

Mean Square Value of Corrected Asynchronies (ms ²)	22124 (149ms)	21630 (147ms)	23188 (152ms)	19450 (139ms)
Number of made adjustments	14	14	14	11

The achieved mean session asynchrony of both sync reference selection strategies are very similar in scenarios with or without network degradation (cases 3-4, and cases 1-2, respectively). Nevertheless, note that although the number of received adjustments requests from the IDMS Manager are the same for all the Clients (around 30), they do not make adjustments a similar number of times when having the most lagged Client as the reference than when having the mean playout point as the reference. As it can be expected, when the selected reference is the most lagged Client, more adjustments are done.

During the evaluation of cases 2 and 4, with packet loss probability, many situations in which the session asynchrony exceeded the session asynchrony threshold were detected but rapidly corrected by the IDMS solution. This was due to the randomness of the broadcast packet loss parameter. If any MPEG2-TS packet containing an I-frame is lost, up to 1s of media content could be discarded as the Group of Pictures⁷ (GoP) size was set to 25 during the encoding of the content. Even the considered probability for broadcast packet loss is considerably higher than QEF, the proposed IDMS solution is still capable of recovering from loss situation, while keeping the mean asynchrony lower than the configured threshold.

C. Subjective Evaluation

The success of multimedia platforms and applications is mainly determined by their acceptability by consumers. With this premise in mind, a subjective evaluation study for the platform, in a Social TV scenario, has been planned. For the sync reference strategy most lagged Client has been selected, since the results of the objective evaluation for it are better than for the mean playout point strategy. Moreover, in this assessment, no broadcast or broadband network degradation has been forced, as the goal of this subjective evaluation is to assess the IDMS functionalities rather than its fault-tolerance (left for future work).

A total of 59 subjects participated in the study. After analyzing the validity of all the evaluations, 6 subjects' responses have been discarded due to contradictory and inconsistent answers. The results from the remaining 53 subjects have been taken into account. Table 10 indicates the subjects' gender and age, among other useful statistics. None of the participants presented any audiovisual impairment that could affect the evaluation process.

⁷ A GoP is an encoding of a sequence of frames that contain all the information that can be completely decoded within that group.

TABLE XIII. PARTICIPANT'S PROFILE

Female (%)	Male (%)	Age	%	Educational Background	%
		51	49		
		18-24	33.96	Bachelor Student	45.28
		25-29	30.19	Bachelor Degree/Msc	30.19
		30-39	9.43	PhD	18.87
		40-49	18.87		
		50-59	1.89		

For the subjective evaluation, 4 consumer sites (each with one hybrid terminal running the MS with the IDMS Client) were located in distributed labs within our Campus. Therefore, groups of up to 4 users have been formed in each test (i.e., each user in a different site in front of the hybrid terminal). As companion devices running the CS chat app, a tablet was employed in two sites, while a smartphone was employed in the other two. Table 6 summarizes their features.

The subjective tests have been targeted at determining: i) the users' satisfaction regarding the performance and usability of the platform; ii) the users' opinions regarding the applicability of the platform and the usefulness of its functionalities; iii) the users' interest in these kinds of platforms and Social TV scenarios; iv) if the IDMS capabilities and Social TV scenarios provide a higher feeling of togetherness, engagement and immersion.

1) Methodology

The subjective evaluation process has followed the recommendations from [54] in order to adequately prepare the home environment, and to not last more than 30 minutes, including a previous training stage.

After that initial training stage, participants evaluated the Social TV in two parts. The first one has consisted of evaluating (by pairs) five cases with different forced asynchrony conditions in order to analyze their impact on the perceived QoE. Users have rated different forced asynchrony scenarios by using the MOS (Mean Opinion Score) likert-type scale [55], which implies rating from 1 to 5 the perceived quality (or impairment) of the content. The values of the forced asynchrony in each evaluated case were the same as the ones considered in [39], which are -4, -2, 0, 2, and 4s. In order to increase the chances of asynchrony levels to be noticed, the chat's sync functionality has been disabled.

The second part of the subjective evaluation consisted on allowing users to freely use the platform in private group sessions of 4 participants each. It was sought that users in every single group knew each other previously, so the communication could be fluent and natural. After this, a questionnaire was given to each participant. The main goals for this second part were to determine: i) if they already had any similar experience with this type of platforms; ii) the usability and applicability of the platform; and iii) if the suitability for the validated maximum session asynchrony value of the objective evaluation does not (negatively) affect the user experience.

2) Conditions

As for the objective evaluation, the same broadcast extra delays to each of the receivers has been emulated (0s, 1.5s; 3s, and 6s for Clients 1 to 4, respectively). Nevertheless, for the subjective evaluation, no broadcast packet loss or broadband network delay has been forced, as the goal of this evaluation is to assess the platform and its functionalities, without the existence of other conditions which might affect the perceived QoE.

3) Results

First part of the study: Participant's feeling of synchronization

Five cases with different fixed asynchrony conditions were presented to the users (but they were not informed about it). The purpose was to force fixed asynchrony within groups of 2 in order to evaluate the impact on the perceived QoE by the users, as seen in [39], by adopting the observed values in that work. When the participants were confident about the score to rate each condition, they notified that to a test supervisor, and then a screen overlay with a questionnaire for that condition was shown to them. This procedure was repeated for the 5 cases. The results in Figure 21 show the mean values for the MOS scale, plus the 95% confidence interval for each case. Negative/positive values in the X-axis mean that the playout process of the user's Media Client was advanced/lagged regarding the one of the sync reference.

Asynchrony started to be noticeable when reaching 2s and started to become annoying when reaching 4s. A rating score close to 5 was obtained when there was no forced asynchrony, reflecting the high users' satisfaction in such a case.

More specifically, when the content was being consumed in a delayed manner regarding the playout reference, participants scores were lower (worse) than when consuming the same content ahead of the playout reference. This may be due to the received spoilers via the chat tool, as this scenario represents the worst case situation, where a participant receives a message related to a content which has not viewed yet.

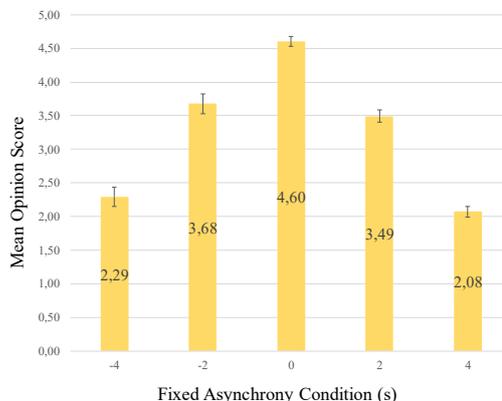


Figure 21: MOS ratings for fixed asynchrony conditions

In conclusion, this first part of the subjective evaluation confirms that the higher the asynchrony value, the more bothering the experience results. Moreover, situations where

users are behind the reference playout, imply a worse rating than when users are forward this playout reference.

Second part of the study: Participant's preferences, habits and opinion about the presented platform

Many interesting insights have been obtained from the answers obtained in the second part of the subjective evaluation. Its purpose was to collect data about any previous experience with similar platforms and its issues. Besides, it confirms the awakened interest in this platform, plus the acceptance that this platform could have in the market. All the collected statistics have been segmented by gender, in order to analyze its potential impact and show more detailed information. The collected statistics have been divided into three different parts: i) previous experience and habits; ii) usability and performance of the platform; and iii) awakened interest and applicability of the platform.

a) *Previous experience and habits*

The participants were asked about their own habits and previous experience (if any) regarding this type of platforms. Responses are summarized in Figures 22 and 23. More than 80% of participants often meet in the same location (e.g. common house) when they want to watch specific events on TV. Besides, only 13% of participants stated they never interacted with other remote people (i.e., not physically present in the same location) while watching TV about the content being watched (Figure 23). 77% of the participants state that geographical distance is a barrier to overcome for this kind of shared experiences, which still have problems or issues that prevent users from enjoying them. However, 88% of the participants consider that this type of platforms enhance the QoE. Notice that only 25% of participants have already used a similar platform to the one presented in this work. 62% of them declared to have experienced problems or issues with that similar platforms, such as, mainly, spoilers through the communication channel or too many chat participants because of having non-private chat rooms.

Regarding gender segmentation, answers from male or female participants are quite similar without significant differences. Independently from the gender, participants have similar habits when watching TV and have experienced the same issues when willing to watch the same TV content with non-close-by friends or relatives. Some answers with 100% values of only one gender have not been considered significant, as this correspond to a small percentage of the overall participants (a 2% value is equivalent to 1 participant).

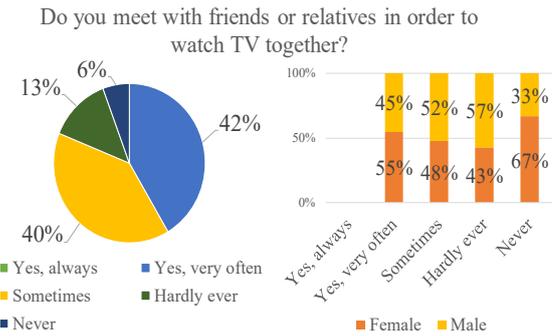


Figure 22: Participants' social habits while watching TV

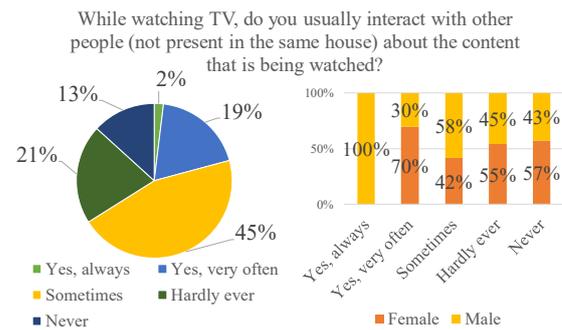


Figure 23: Participants' social habits with non-close-by people while watching TV

b) *Usability and performance of the platform*

Participants were also asked about other relevant features of the presented platform (performance, accuracy, smoothness, user interface...). 89% stated that the platform increases the feeling of togetherness, 95% declared that it is easy to use, and 94% thought it is a useful and interesting platform. More than 2/3 of the participants preferred using a private, interactive and synchronized chat tool, like the one developed in this work, against the use of popular and non-integrated social media based chats (Figure 24). This majority opinion may be mainly due to many factors, such as: i) the spoiler avoidance, thanks to the synchronization with the main content; ii) the privacy of the sessions, against when using social networks; and iii) the inclusion of the contextualized and event-driven messages, as they enable more interactive and faster communications, and thus more engaging and personalized experiences.

Indeed, 74% of participants stated that the contextualized and event-driven messages (buttons) allow having more fluid conversations (Figure 25).

Regarding synchronization, IDMS has been successfully achieved, as participants considered not to have received out-of-sync chat messages or having the feeling of going ahead or behind the rest of the participants's content (76% and 82%, respectively). It should be noted that when participants were asked for the reason of receiving significantly delayed chat messages, 64.2% thought this delay was human caused (i.e., due to the delays in typing the message). Moreover, 84.9% declared that chat delays were acceptable and not annoying.

No significant differences have been devised from male and female answers in this case either. Note that the observed good

performance of the platform in the objective evaluation can be corroborated and complemented with the results obtained in this specific subsection.

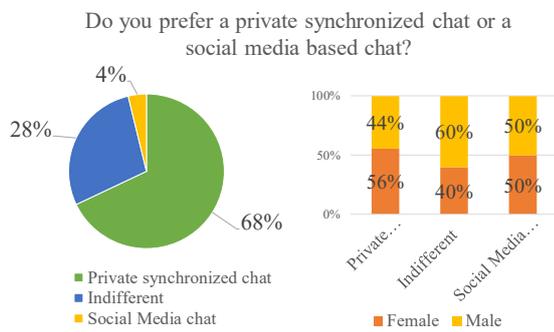


Figure 24: Chat preferences

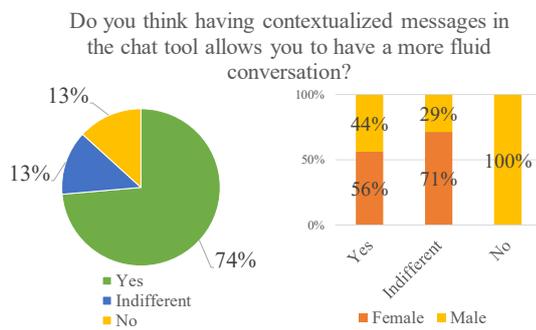


Figure 25: Contextualized messages suitability

c) *Awaken interest and applicability of the platform*

Participants also provided their opinion about the usefulness and their impression regarding the potential of the platform. 74% considered this platform solves a real problem. Most of the participants (approximately 78%) thought the developed platform and Social TV service could have an impact in the current TV content consumption market (Figure 26). 83% would use it, if available (Figure 27). Only 4% of participants disagreed with the idea that this platform provides more personalized, immersive and enriched consuming experiences. Finally, 87% of the participants stated that this type of platforms could also be used for other applications beyond Social TV, such as e-learning, reflecting its wide applicability.

Independently from the gender, participants have similar expectations to experience this way of Social TV consumption.

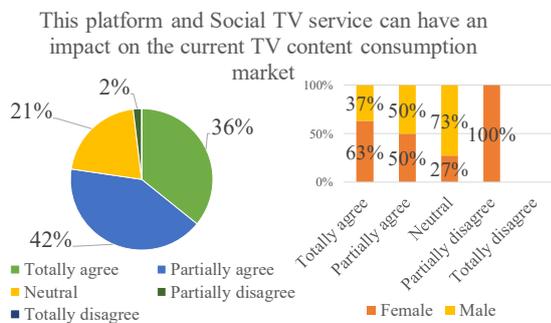


Figure 26: Impact of the platform on the current market

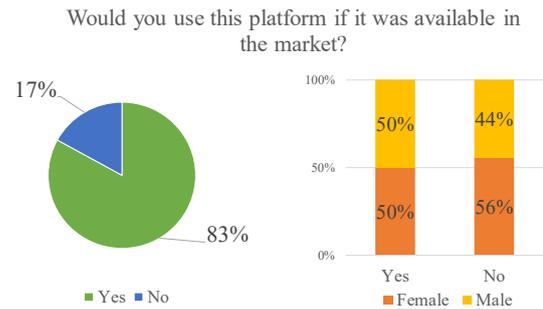


Figure 27: Participants' will of using this platform if available

VII. CONCLUSIONS AND FUTURE WORK

This paper has presented an adaptive IDMS solution for enabling shared media experiences between remote users, when hybrid media delivery is employed. Apart from providing cross-technology support, it is able to accomplish synchronization when different formats/versions of the same (or related) contents are being played out in each shared session. The solution has been integrated within an end-to-end HbbTV-compliant platform, which enables the delivery, reception and synchronized consumption of hybrid media contents, either on single devices or on multiple close-by devices.

To evaluate the proposed solution, the representative interactive Social TV use case has been selected and implemented in the platform. To achieve this, an ad-hoc, interactive, private and synchronized text chat tool has also been developed and integrated.

The results of both objective and subjective assessments prove the consistent behaviour and satisfactory performance of the overall platform, paying special attention to the designed IDMS solution and to the experienced delays. The objective evaluation confirms the achievement of accurate IDMS, while the participants in the subjective evaluation rates this platform with satisfactory QoE levels. Some interesting conclusions are summarized in the next points:

- The proposed solution solves a real problem, as asynchrony between remote users might exist (and even increase) if no action is taken.
- The platform's performance is satisfactory, as the asynchrony value is maintained below the configured threshold during the IDMS session.
- Users' perceived QoE improves as the asynchrony decreases. This justifies the usefulness of IDMS mechanisms.
- Participants showed an awakened interest in the presented platform, as they stated that this proposed solution provides personalized, immersive and enriched consuming experiences.

The contributions of this paper are timely and relevant, as they allow extending the capabilities of worldwide deployed standard technologies to successfully enable networked shared media experiences. Within the context of Social TV, they will contribute to providing more interactive, social and immersive TV watching experiences.

As future work, authors expect to define, implement and evaluate a distributed IDMS control scheme (DCS) also using the described platform, and including related broadband content consumption (requiring both IDMS+IDES combined mechanisms).

REFERENCES

- [1] Miyahista, Y., Ishibashi, Y., Fukushima, N., Sugarawa, S., Psannis, K.E.: QoE assessment of group synchronization in networked chorus with voice and video. In Proceedings of IEEE TENCON'11, pp. 393-397, 2011.
- [2] M. Montagud, F. Boronat, H. Stokking, R. van Brandenburg, "Inter-Destination Multimedia Synchronization; Schemes, Use Cases and Standardization", *Multimedia Systems*, 18(6), 459-482, November 2012.
- [3] F. Boronat, M. Montagud, H. M. Stokking and O. Niamut, "The need for inter-destination synchronization for emerging social interactive multimedia applications," in *IEEE Communications Magazine*, vol. 50, no. 11, pp. 150-158, November 2012. doi: 10.1109/MCOM.2012.6353695
- [4] M. O. Van Deventer, H. Stokking, O. A. Niamut, F. A. Walraven, and V. B. Klos., "Advanced Interactive Television Service Require Synchronization", 15th International Conference on Systems, Signals and Image Processing, IWSSIP'08, Bratislava (Slovak Rep.), June 2008.
- [5] W. Kooij, Stokking, R. van Brandenburg, P-T de Boer, "Playout delay of TV signals: measurement system design, validation and results", *ACM TVX 2014*, Newcastle (UK), June 2014.
- [6] F. Boronat, M. Montagud, D. Marfil, C. Luzón, "Hybrid Broadcast/Broadband TV Services and Media Synchronization: Demands, Preferences and Expectations of Spanish Consumers", *IEEE Transactions on Broadcasting*, vol. PP, no. 99, pp. 1-18, August 2017.
- [7] M.O. van Deventer, H. Stokking, M. Hammond, J. Le Feuvre, P. Cesar, "Standards for Multi-stream and Multi-device Media Synchronization", *IEEE Communications Magazine*, 54(3), pp.16-21, 2016.
- [8] NIELSEN, "The Nielsen Comparable Metrics Report", Q2-2016, Available at: <http://www.nielsen.com/>, Accessed in November 2017 us/en/insights/reports/2016/the-comparable-metrics-report-q2-2016.html
- [9] A. Domínguez, M. Agirre, J. Flórez, A. Lafuente, I. Tamayo and M. Zorrilla, "Deployment of a Hybrid Broadcast-Internet Multi-Device Service for a Live TV Programme", *IEEE Transactions on Broadcasting*, vol. PP, no. 99, pp. 1-11, 2017
- [10] R. Mekuria, "Inter-destination media synchronization for TV broadcasts", Master Thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, Dept. of Network architecture and Services, Delft University of Technology, April 2011.
- [11] Hybrid Broadcast Broadband TV (HbbTV) 2.0.1 Specification, HbbTV Association Resource Library, <https://www.hbbtv.org/resource-library>, July 2016.
- [12] F. Boronat, D. Marfil, M. Montagud, J. Pastor, "HbbTV-Compliant Platform for Hybrid Media Delivery and Synchronization on Single- and Multi-Device Scenarios", *IEEE Transactions on Broadcasting* Volume PP, Issue 99, January 2018.
- [13] ITU-T Rec. G.1050: "Network model for evaluating multimedia transmission performance over Internet Protocol", 2007.
- [14] J. Jansen, P. Cesar, D.C.A. Bulterman, T. Stevens, I. Kegel, J. Is-sing, "Enabling Composition-Based Video-Conferencing for the Home", *IEEE Transactions on Multimedia (TMM)*, 13(5), pp. 869-881, October 2011.
- [15] M. Montagud, P. Cesar, F. Boronat, J. Jansen (editors), "MediaSync. Handbook on Multimedia Synchronization", Springer International Publishing, 2018. DOI 10.1007/978-3-319-65840-7
- [16] F. Boronat, J. Lloret, M. García, "Multimedia group and inter-stream synchronization techniques: A comparative study", *Information Systems*, 34(1), pp. 108-131, March 2009.
- [17] Z. Huang, K. Nahrstedt, R. Steinmetz, "Evolution of temporal multimedia synchronization principles: A historical viewpoint", *ACM TOMCAP*, 9, 1s, Article 34, October 2013.
- [18] L. Beloqui, F. Boronat, M. Montagud, H. Melvin, "Understanding Timelines within MPEG Standards", *IEEE Comm. Surveys & Tutorials*, 18(1), pp. 368-400, 2016.
- [19] Network Time Protocol Version 4: Protocol and Algorithms Specification, IETF Internet Standard RFC 5905, 2010. [Online]. Available: <https://tools.ietf.org/html/rfc5905>
- [20] ETSI 102 823, "Digital Video Broadcasting (DVB);Specification for the carriage of synchronized auxiliary data in DVB transport streams", 2005.
- [21] ISO/IEC 13818-1:2013/PDAM 6 Delivery of Timeline for External Data, 2013, <http://mpeg.chiariglione.org/standards/mpeg-2/systems/text-isoiec-13818-12013pdam-6-delivery-timeline-external-data>
- [22] M. Montagud, F. Boronat, H. Stokking, P. Cesar, "Design, Development and Assessment of Control Schemes for IDMS in a Standardized RTPC-based Solution", *Computer Networks*, Volume 70, 9, September 2014.
- [23] Y. Ishibashi, A. Tsuji, S. Tasaka, "A Group Synchronization Mechanism for Stored Media in Multicast Communications", *INFOCOM '97*, Washington, April 1997.
- [24] Y. Ishibashi, S. Tasaka, "A group synchronization mechanism for live media in multicast communications", *IEEE GLOBECOM'97*, pp. 746-752, November 1997.
- [25] R. Arima, M. Sithu, Y. Ishibashi, "QoE assessment of fairness between players in networked virtual 3D object identification game using haptic, olfactory, and auditory senses", *International Journal of Communications, Network and System Sciences (IJCNSS)*, 10(7), pp. 129-141, July 2017.
- [26] M. Mauve, J. Vogel, V. Hilt, W. Effelsberg, "Local-Lag and Timewarp: Providing Consistency for Replicated Continuous Applications", *IEEE Transactions on Multimedia*, Vol.6, No.1, February 2004.
- [27] I. Vaishnavi, P. Cesar, D. Bulterman, O. Friedrich, S. Gunkel, D. Geerts, "From IPTV to synchronous shared experiences challenges in design: Distributed media synchronization", *Signal Processing: Image Communication*, Vol. 26, Issue 7, pp. 370-377, August 2011.
- [28] F. Boronat, J.C. Guerri, J. Lloret. 2008. An RTP/RTCP based approach for multimedia group and inter-stream synchronization. *Multimedia Tools Appl.* 40, 2 (November 2008), 285-319. DOI=<http://dx.doi.org/10.1007/s11042-008-0208-1>
- [29] R. van Brandenburg, H. Stokking, O. Van Deventer, F. Boronat, M. Montagud, K. Gross, "Inter-destination Media Synchronization RTPion using the RTP Control Protocol (RTCP)", IETF Standard, RFC 7272, June 2014.
- [30] B. Rainer, C. Timmerer, "Self-Organized Inter-Destination Multimedia Synchronization For Adaptive Media Streaming", *ACM MM 2014*, Florida (USA), November 2014.
- [31] J. Belda, M. Montagud, F. Boronat, M. Martínez, J. Pastor, "Wersync: A Web-based Platform for Distributed Media Synchronization and Social Interaction", *ACM TVX 2015*, Brussels (Belgium), June 2015.
- [32] RTP: A Transport Protocol for Real-Time Applications, IETF Internet Standard RFC 3550, 2003. [Online]. Available: <https://tools.ietf.org/html/rfc3550>.
- [33] M. Montagud and F. Boronat. 2012. Enhanced adaptive RTCP-based Inter-Destination Multimedia Synchronization approach for distributed applications. *Comput. Netw.* 56, 12 (August 2012), 2912-2933. DOI=<http://dx.doi.org/10.1016/j.comnet.2012.05.003>
- [34] I. Fette, A. Melnikov, "The WebSocket Protocol", IETF Standard, RFC 6455, December 2011.
- [35] L. Oehlberg, N. Duchenaud, D. Thornton, "Social TV: Designing for distributed, sociable television viewing", *Euro ITV 2006*, Athens, 2006.
- [36] M. Montagud, P. Cesar, F. Boronat, D. Marfil, "Social Media Usage Combined with TV/Video Watching: Opportunities and Associated Challenges", *IEEE STCSN E-Letter* Vol.3 No.2, 2015.
- [37] J. D. Weisz, Sara Kiesler, Hui Zhang, Yuqin Ren, Robert E. Kraut, Joseph A. Konstan. "Watching Together: Integrating Text Chat with Video", *ACM CHI '07*, San Jose, 2007.
- [38] D.A. Shamma, M. Bastea-Forte, N. Joubert, Y. Liu, "Enhancing online personal connections through synchronized sharing of online video", *ACM CHI'08*, Florence (Italy), April 2008.
- [39] D. Geerts, I. Vaishnavi, R. Mekuria, O. van Deventer, P. Cesar, "Are we in sync?: synchronization requirements for watching online video together", *ACM CHI'11*, Vancouver (Canada), May 2011.
- [40] D. Geerts, "Comparing voice chat and text chat in a communication tool for interactive television", *NordiCHI '06*, Oslo, 2006.
- [41] E. Huang, Gunnar Harboe, Joe Tullio, Ashley Novak, Noel Massey, Crysta Metcalf, Guy Romano. "Social Television comes home:" a field study of communication choices and practices in TV-based text and voice chat", *ACM CHI '09*, Boston (USA), 2009.

- [42] R. Mekuria, P. Cesar, D. Bulterman, "Digital TV: the effect of delay when watching football", EuroITV '12, Berlin (Germany), July 2012.
- [43] B. Rainer, S. Petschamig, C. Timmerer, H. Hellwagner, "Is one second enough? evaluating QoE for inter-destination multimedia synchronization using human computation and crowdsourcing", IEEE QoMEX 2015, Greece, 2015.
- [44] V. Vinayagamoorthy, R. Ramdhany, M. Hammond, "Enabling Frame-Accurate Synchronized Companion Screen Experiences", ACM TVX'16, Chicago (USA), June 2016.
- [45] C. Ziegler, C. Keimel, R. Ramdhany, V. Vinayagamoorthy, "On Time or Not on Time: A User Study on Delays in a Synchronized Companion-Screen Experience", ACM TVX '17, Hilversum, Netherlands, June 2017.
- [46] ETSI TS 106 286-1, "Companion Screens and Streams; Part 2: Content Identification and Media Synchronization", 2015.
- [47] RTP Clock Source Signaling. IETF Internet Standard RFC 7273, June 2014. [Online]. Available <https://tools.ietf.org/html/rfc7273> and D. Veitch, "Principles of robust timing over the Internet," Commun. ACM, vol. 53, no. 5, pp. 54–61, May 2010.
- [49] M. Montagud and F. Boronat, "On the Use of Adaptive Media Playback for Inter-Destination Synchronization," in IEEE Communications Letters, vol. 15, no. 8, pp. 863-865, August 2011. doi: 10.1109/LCOMM.2011.061611.110072
- [50] M. Montagud, F. Boronat, B. Roig, A. Sapena, "How to perform AMP? Cubic adjustments for improving the QoE", Computer Communications, Volume 103, 1, Pages 61-73, May 2017.
- [51] GStreamer framework, <https://GStreamer.freedesktop.org/>
- [52] GPAC Framework, <https://gpac.wp.mines-telecom.fr/>
- [53] DIAL (Discovery And Launch) protocol specification, Version 1.7.2, 2015, <http://www.dial-multiscreen.org/dial-protocol-specification/DIAL-2ndScreenProtocol-1.7.2.pdf>
- [54] ITU-R Rec. BT.500-13: "Methodology for the subjective assessment of the quality of television pictures", 2012.
- [55] ITU-T Rec. P.800: "Methods for objective and subjective assessment of quality", 1998.



Dani Marfil was born in Gandia (Spain) and studied Informatics Technical Engineering (2011), Telecommunications Degree (2015) and Telecommunication Technologies, Systems and Networks MsC (2016) in Universitat Politècnica de València (UPV, Spain). He is a PhD student and an assistant researcher and

developer in the Immersive Interactive Media R&D Group. His main topics of interest are communication networks, code developing and media synchronization. He is the author of one book chapter and nine research and conference papers.



Fernando Boronat (M'93–SM'11), was born in Gandia (Spain), and went to the Universitat Politècnica de València (UPV, Spain) in Spain, where he studied Telecommunications Engineering. He received the M.E. and Ph.D. degrees in telecommunication engineering from the UPV in 1994 and 2014, respectively. After

working for several Spanish telecommunication companies, he moved back to the UPV in 1996. Currently he is an Assistant Professor in the Communications Department. Dr. Boronat is the Head of the Immersive Interactive Media R&D Group at the Gandia Campus of UPV. His main topics of interest are communication networks, multimedia systems, multimedia protocols, and media synchronization. He is the author of two books, several book chapters, an IETF RFC and more than 100 research papers. He is Editor of "MediaSync: Handbook on Multimedia Synchronization" (Springer, 2018). He is involved in several IPCs of national and international journals and conferences. He is member of IEEE and ACM.



Mario Montagud was born in Montitxelvo (Spain). He received a BsC in Telecommunications Engineering in 2011, an MsC degree in "Telecommunication Technologies, Systems and Networks" in 2012 and a PhD degree in Telecommunications (Cum Laude Distinction) in 2015, all of them at the Universitat Politècnica de València (UPV,

Spain). During his PhD degree and after completing it, he did 3 research stays (18 months) at CWI (The National Research Institute for Mathematics and Computer Science in the Netherlands) in Amsterdam. His topics of interest include Computer Networks, Interactive and Immersive Media, Synchronization, Human Computer Interaction (HCI) and QoE. Mario is (co-) author of over 50 scientific and teaching publications, and has contributed to standardization within the IETF. He is TPC member of several international conferences, co-organizer of the international MediaSync Workshop series, member of the Editorial Board of international journals, and Communication Ambassador of ACM SIGCHI. He is also Editor of "MediaSync: Handbook on Multimedia Synchronization" (Springer, 2018).



Almanzor Sapena received his BS degree in mathematics from the University of Valencia. In April 2002, he received his PhD degree from the Universitat Politècnica de València (UPV, Spain), with a focus of topological properties in fuzzy metric spaces. He is now an associate professor in the Department of Applied Mathematics at UPV and is doing research

on fuzzy topology and noise reduction in digital images and on adaptive media playout techniques. He has published some papers on these topics in international journals and conferences. He is a member of the Immersive Interactive Media R&D Group at Gandia Campus of the UPV.