IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 5, 2018, accepted December 4, 2018. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2018.2885580

Synchronization Mechanisms for Multi-User and Multi-Device Hybrid Broadcast and Broadband Distributed Scenarios

DANI MARFIL [1], FERNANDO BORONAT [1], (Senior Member, IEEE), ALMANZOR SAPENA [2], AND ANNA VIDAL [2]

[1]Department of Communications, Universitat Politècnica de València, Campus de Gandia, 46730 Gandia, Spain
[2]Department of Mathematics, Universitat Politècnica de València, Campus de Gandia, 46730 Gandia, Spain

Corresponding author: Dani Marfil (damarre@dcom.upv.es)

This work supported by the Generalitat Valenciana, Investigación Competitiva Proyectos, through the R&D Program "Grants for research groups to be consolidated, AICO/2017," under Project AICO/2017/059.

ABSTRACT Traditionally, co-located TV content consumption has been a social habit (e.g., family members in the living room watching TV). It cannot be always possible to enjoy these shared experiences in the current global society, in which relatives or friends might live apart. In this context, the emergence of new technologies together with the advent of social media and conferencing services brings the possibility of recreating them when users are geographically distributed. When remote users are involved in a networked shared TV consumption experience (Social TV), inter-destination media synchronization (IDMS) is a key requirement to provide them with a satisfactory experience. Moreover, enriching the broadcasted TV content with alternative (or complementary) broadband content is gaining momentum. When several co-located consumption devices are involved in the TV viewing experience, inter-device synchronization (IDES) mechanisms are also needed to allow successful multi-screen experiences. This paper presents a combined synchronization mechanism (including both IDMS and IDES solutions) for distributed scenarios involving remote users and hybrid (broadcast and broadband) contents, to be consumed in one or several devices. The mechanism provides multi- and cross-technology support and achieves accurate synchronization levels even when different versions or formats of the media contents are being played out. It has been integrated into a hybrid broadcast broadband TV standard-based platform, and the achieved synchronization accuracy and the performance of the solution for the Social TV use case have been evaluated, obtaining satisfactory and promising results.

INDEX TERMS Broadband, broadcast, DASH, DVB, hybrid TV, IDES, IDMS, media synchronization, shared media experiences, Social TV.

I. INTRODUCTION

Traditionally, media contents have been conceived to be delivered through specific technologies and consumed through particular devices. For instance, traditional TV content has been received via aerials (i.e., broadcast content) and played out on simple TV sets. Nonetheless, nowadays media delivery and consumption have become heterogeneous and globally accessible. Media contents can be encoded in many formats or versions and can be delivered through different technologies or networks. This evolution allows the increase in availability of media contents for the users, providing more adaptive and ubiquitous media services.

On the one hand, the availability of connected TVs (i.e., Main Screen -MS-), along with a large range of different consumption devices (e.g., smartphones, tablets...) acting as secondary or Companion Screens (CS) in multi-screen applications is a reality. Nevertheless, there is an extra challenge associated to the complementary use of these types of devices, as they can receive media contents via different delivery technologies and might have heterogeneous performance, capabilities or limited resources. This heterogeneous scenario may lead to incompatibility issues and conflicts between providers or manufacturers, but it also opens new research and development opportunities. Recent HbbTV (Hybrid Broadcast Broadband TV) standard [1] is a proof of the importance of the use of hybrid technologies to enrich TV services. It is currently being deployed and provides functionalities and mechanisms to synchronously consume hybrid content in the MS and, optionally, in one (or more) CSs. Evidence of the importance of this standard and of the new hybrid TV consumption scenario is the existence of the many European research projects which have been focused on the use of hybrid technologies to enrich TV services, such as HBB-NEXT,[1] HBB4ALL,[2] TV-RING,[3] MEDIASCAPE,[4] MPAT,[5] ImAc[6] or ImmersiaTV.[7]

On the other hand, the social consumption of media content around a single device (e.g., a group of friends watching a football match together at home) has been a traditional habit. However, many issues might prevent users from meeting in a common place and share a local media consumption experience. As the world has become globalised, people might have moved to different geographical areas for work, study or for any other reason. Despite this physical distance between groups of users, shared media consumption experiences are possible (but not easy to accomplish).

The latest advances in delivery technologies, the global adoption of social networking and the huge connected media environment make the desired "watching apart together" experiences (i.e., network togetherness) possible. The evolution of these networked shared media experiences will provide the connection and interaction between remote users who share common interests, reducing distance gaps, saving time and minimizing costs. Figure 1 shows the networked shared TV viewing experience, which is known as Social TV.

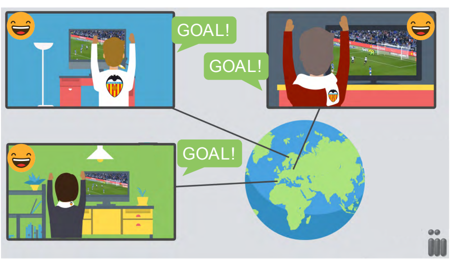

FIGURE 1. Successful network shared media experiences.

Nevertheless, there are still many challenges to be tackled in order to provide those shared experiences, such as dynamic community building, media synchronization (sync hereafter), Quality of Service/Experience (QoS/E), scalability, presence awareness, privacy concerns, and social networking integration [2]. In particular, this paper focuses on one of these challenges, media sync, which should be guaranteed between the playout processes of multiple remote devices (a.k.a. Inter Destination Media Sync or IDMS) and/or close-by devices (a.k.a. Inter Device Sync or IDES) [3]). As the main contribution, this paper presents a combined sync mechanism including IDMS and IDES solutions for networked shared media consumption scenarios, involving hybrid (broadcast and broadband) media delivery and multiple devices. It works for hybrid scenarios with any content encapsulation format which involves MPEG2 Transport Streams (MPEG2-TS). In [4], Marfil *et al.* presented a preliminary IDMS solution following only a centralized control scheme (the existing types of control schemes are briefly described in section II) without considering IDES. In this paper, on the one hand, the IDMS part of the proposed sync mechanism can follow both centralized and distributed control schemes. On the other hand, it is combined with an IDES mechanism, previously proposed by Boronat *et al.* [5]. Moreover, as another novelty, a set of messages are also proposed in this paper to provide session management capabilities and allow the sync of the involved users in a shared session.

In order to assess the accuracy obtained by the proposed mechanism, it has been integrated into an end-to-end hybrid delivery platform, described in [4] and [5]. It has been implemented and evaluated for the hybrid Social TV use case with multi-screen and multi-view functionalities, with distributed users and involving several consumption devices in each destination. This use case has recently been considered by Spanish users as relevant and with a high commercial potential [6]. The proposed sync mechanism allows to manage the media contents' playout processes of distributed online users with two or more consumption devices (screens) involved (at least one connected TV as an MS and one secondary device as a CS).

The presented mechanism supports multi- and cross-technology and the obtained results show that it provides high (IDMS and IDES) sync accuracy even when the involved devices are playing out diverse formats or versions of the same (or related) hybrid content. These features have been included by adopting standard mechanisms, which supply absolute and traceable timelines within the delivered media streams. Besides, the proposed sync mechanism is adaptive (it can correct asynchrony situations in an adaptive and timely manner as soon as they are detected), accurate (it maintains the asynchrony below very restrictive thresholds defined for most of the current Social TV scenarios) and standard-compliant (it follows HbbTV standard [1]).

The remainder of this paper is structured as follows. In Section II, a review of previous related works is presented. In Section III, the proposed sync mechanism including combined (IDMS + IDES) solutions is described. In Section IV, the integration of both mechanisms in the Social TV use case implemented into an end-to-end hybrid delivery platform is explained. Next, in Section V, the validation results of the proposed mechanisms are shown. Finally, in Section VI, some conclusions and future work are summarized.

[1]https://cordis.europa.eu/project/rcn/100252_es.html (last access, October 2018)
[2]http://www.hbb4all.eu/ (last access, October 2018)
[3]http://www.tvring.eu/ (last access, October 2018)
[4]http://www.mediascapeproject.eu/ (last access, October 2018)
[5]http://mpat.eu/ (last access, October 2018)
[6]http://www.imac-project.eu/ (last access, October 2018)
[7]http://www.immersiatv.eu/ (last access, October 2018)

II. RELATED WORKS

In this section, previous works regarding networks' delay variability, timelines accessibility, timestamping techniques and existing IDES/IDMS sync solutions are overviewed.

A. DELAY VARIABILITY CAUSED BY DELIVERY NETWORKS

Along the end-to-end delivery chain, different causes and system elements can add additional delays (besides its variability). In [7] and [8] the sources of these delays and their effect for broadband and broadcast technologies, respectively, are described. Undesired situations can cause that media contents which have been generated simultaneously can not necessarily be received simultaneously by the involved consumption devices. Consequently, these contents will not be played out at the same time, reaffirming the need for inter device/destination media sync solutions [6]. Existing delay differences between destinations in current delivery networks have already been identified in previous works (e.g., [8]–[11]).

Broadcast technologies use different techniques and distribution channels and, consequently, this fact can affect the final end-to-end delays. The work in [8] presents measurements of delay differences for different TV scenarios. In national scenarios (The Netherlands) delay differences can be up to 5s. However, in international scenarios (between The Netherlands and the UK), delays can grow up to 6s. Significant delay differences are detected even when different receivers use exactly the same TV setup (i.e., the same delivery technology, subscription type and equipment).

Regarding broadband technologies, typical values of delays (from 20 to 500ms) and jitter (from 0 to 500ms) in Internet are reported in the International Telecommunications Union (ITU-T) G.1050 standard [12]. According to [9], for video conferencing systems in heterogeneous scenarios, end-to-end delays can range between tens of ms up to 300ms. In Internet Protocol Television (IPTV) scenarios, some studies show that end-to-end delay differences can be higher than 6s [10]. In [11] it is stated that depending on the resolution of the video content (i.e., Standard or High Definition -SD or HD-), delay differences can reach 8s.

B. TIMELINES AND TIMESTAMPING METHODOLOGIES

Typically, media delivery technologies rely on the insertion of *intrinsic* and relative timelines into the media streams, which are commonly obtained from local clocks. 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. Thus, these types of timelines are not enough for IDES/IDMS, as media content can be generated by different sources, or even networking equipment can override *intrinsic* timelines throughout the delivery chain. Unlike intrinsic timelines, absolute and *extrinsic* timelines can be obtained from the same or traceable clock sources (e.g., Network Time Protocol -NTP- [13]) and its reference keeps unmodified throughout all the delivery chain (see Figure 2). This is the approach that follows RTP/RTCP streaming, as well as other mechanism proposed by the European Telecommunications Standards Institute (ETSI) in [14]. The latter is based on inserting an external and absolute timeline into MPEG2-TS streams.

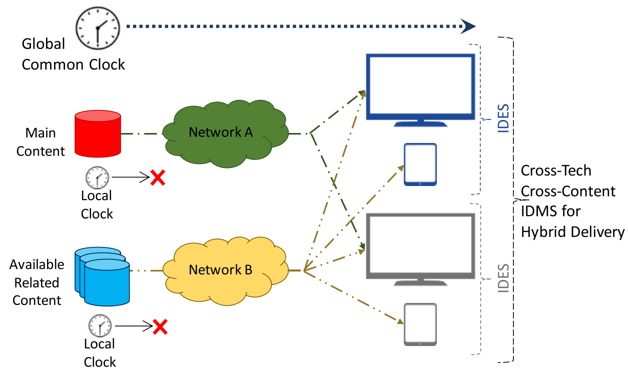

FIGURE 2. Global and Common Timelines in Media Streaming.

Additionally, another mechanism called Timeline and External Media Information (a.k.a. TEMI) has been proposed by MPEG and DVB as an amendment to ISO/IEC 13818-1 [15]. It has been adopted for the solution proposed in this work, as it is contemplated in the HbbTV specification as a mechanism to synchronize media contents.

C. INTER-DEVICE SYNC (IDES)

The need for IDES has become more popular due to the spread of heterogeneous consumption devices, as addressed in [5]. This fact enables new ways of content consumption, for example, as stated in [6], nowadays multi-view scenarios are growing their demand and popularity among users.

Specifically, regarding IDES in hybrid TV consumption scenarios, the new HbbTV standard [1], currently under development, provides functionalities and mechanisms to synchronously consume hybrid content in the main TV device (MS) and, optionally, in one (or more) secondary devices (CS), such as smartphones or tablets.

In specific multi-device scenarios, in order to achieve IDES, the involved devices (i.e., a MS and one or more CS) must first discover each other and establish a communication channel in order to allow (bi-directional) communication. With reference to discovery mechanisms, in [16] different approaches are reviewed and compared from different points of view: a) market solutions; b) discovery and association challenges; and c) the translation of general scientific challenges to a TV home environment.

Regarding hybrid media sync, the ETSI Timeline Mechanism [14] is used in [17] to achieve IDES in second screen TV experiences. In a similar way, by also using [14], [18] combines a tiled streaming solution via HTTP Live Streaming (HLS) for ultra-high resolution media content while receiving a DVB stream through the main TV.

In [19] and [20], implementations of the DVB-Companion Screens and Streams (DVB-CSS) specification [21] (also adopted by HbbTV standard [1]) are presented. Although no conclusive results are stated, in [20] asynchrony delays from -500ms to 1000ms are shown to be unnoticeable for the participants in the subjective test.

The work in [22] provides results on a large pilot for multi-device hybrid service for a live TV program. This use case included multi-view streams (delivered via HLS), social media, election results and statistics (charts and tables) data. In that work, accurate sync was not critical as the broadcaster preferred a best-effort service, so as not to add extra delay to the delivered main signal.

D. INTER DESTINATION MEDIA SYNC (IDMS)

There is a clear need to synchronize playout processes by using an IDMS solution in shared media experiences. Temporal unalignment between participants in this kind of experience can cause incoherent interactions (e.g., being aware of a goal via the interaction channel before watching it on the local TV), and, consequently, can worsen the perceived Quality of Experience (QoE). A detailed description of different use cases which use diverse IDMS solutions can be found in [7], [23], and [24]. In [7], up to 20 use cases where IDMS is needed (or becomes beneficial) are described.

Regarding synchronization control schemes used in IDMS, two different approaches exist: centralized and distributed. When having a centralized control, two possibilities can be found: the Master/Slave (M/S) receiver scheme and the Synchronization Maestro Scheme (SMS). The other approach is a Distributed Control Scheme (DCS). Figure 3 presents those three control schemes. A more detailed description of them can be found in [7]. In the sync mechanism proposed in this paper, SMS and DCS control schemes are adopted.

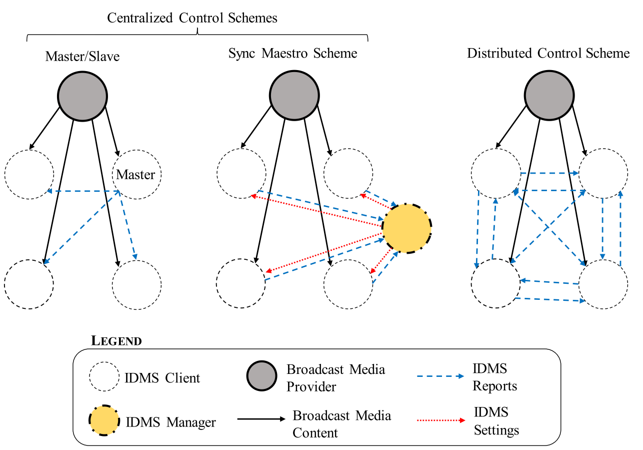

FIGURE 3. Sync Control Schemes for IDMS solutions.

Existing IDMS solutions (most of them compiled in [7]) still present important limitations:

i) Most of them are proprietary protocols (e.g., [25]–[29]) and techniques. However, as discussed in [30] and [31], the availability of standard or even standard-compliant solutions will provide many benefits.

ii) They are exclusively focused on broadband environments [24]–[29], [32]–[35]. Nevertheless, broadcast technologies can still be considered the most appropriate to reach a huge audience [36]. Previous studies have shown significant delay differences between destinations in broadcast scenarios [8], [11], and, therefore, availability of IDMS solutions for broadcast environments is also necessary.

iii) Every proposed solution only works for a single delivery technology at a time. It is not possible to combine more than one delivery technology for a single shared media session [24]–[29], [32]–[35]. Nevertheless, nowadays it is realistic to expect a scenario where different participants in a shared session receive media contents through different delivery technologies.

iv) Current solutions are only intended to provide shared media experiences where the content being played out is the same for all the involved users in the session [24]–[29], [32]–[35], regardless of the device which is playing out this content (i.e., without considering the devices' performance or resources).

v) As far as authors know, no IDMS solutions have been tested in realistic, nearly to commercial environments but our previous work in [4] and the presented work in this paper. This is a gap, as IDMS solutions must be integrated with real setups in order to be validated and to, finally, enhance the conventional media services.

The sync mechanism proposed in this paper overcomes these limitations. First, it is based on open protocols. Second, the proposed sync mechanisms can handle both broadband and broadcast delivered content. They can be used when more than one delivery technology in one single session is used, so one user can consume the content via broadcast and another user in the same session can receive the same content via broadband. Third, more than one content can be consumed in the same shared session. For instance, a user in a shared session can watch the content via broadcast, and another user in the same shared session can watch a related content (e.g., a different point of view of the same event). Finally, the proposed sync mechanism has been tested in close to commercial scenarios, as one of the goals is to enhance real hybrid services.

III. SYNCHRONISATION MECHANISM

In this section, the proposed sync mechanism for multi-device and multi-user hybrid scenarios is described. It is based on two types of media sync solutions: IDMS and IDES. First, to achieve sync between destinations (IDMS), only the sync of their playout processes of the broadcasted (main) content is considered. Then, in each location, an additional IDES solution is adopted to achieve the sync of the playout of all the other related (secondary) broadband contents with the playout of the main content. Content signaling, media timelines inclusion and the combination of both IDES and IDMS solutions are explained in the following subsections.

A. RELATED CONTENT SIGNALING & MEDIA TIMELINES

In the proposed sync mechanism, it is assumed that the main content can be received via broadcast or broadband technologies, while related contents are received via broadband technologies. Therefore, there is a need to signal and locate additional related content and embed a reference to this information somehow in the main content. The use of an XML formatted file containing all the extra needed information is proposed. In it, useful information for the sync mechanism is provided, as in [4] and [5]. All the additional data referenced in this file will be available via broadband networks. This data involves a wide range of complementary services to the main content, from related contents to the main content to different control data useful to achieve IDMS or IDES (e.g., the IDMS Manager's location, described in [4], if it exists – in SMS control). Besides, in order to achieve accurate IDES and/or IDMS, a common global time reference is needed.

As in our previous works in [4] and [5], in this paper the aforementioned TEMI solution [15] has also been adopted to insert content signaling and an external and absolute timeline into MPEG2-TS streams. In particular, the *location descriptor* is used to signal the aforementioned XML file which contains all the extra related information to the main content. Besides, the *timeline descriptor* is used to embed absolute timelines in all the multimedia related content.

B. IDES FEATURES

Regarding IDES, in the proposed sync mechanism, the solution defined in [5] has been adopted, which follows the IDES requirements of the HbbTV standard [1].

1) SERVICE DISCOVERY & COMMUNICATION CHANNEL

In hybrid scenarios, every involved user should have the option to choose to consume related media contents in one or several CSs while simultaneously the main content is being played out in the MS. Therefore, both types of devices (MS and CSs) should be able to discover each other, and then a bi-directional communication channel between them should be established to exchange useful information to achieve accurate IDES. For this purpose, as specified in [1], the Discovery and Launch (DIAL) service [37] has been adopted and implemented into the MS. With this mechanism, CSs can detect available DIAL services in a LAN. Specifically, for this work, the required service is the HbbTV DIAL Application Name, as registered in the official website.[8] Once the CSs discover the MS, a bi-directional communication channel between each CS and the MS is stablished by using WebSocket technology [38]. Then, MS and CSs can exchange information, regarding broadband available content and playout timing information (for sync purposes).

[8]http://www.dial-multiscreen.org/dial-registry/namespace-database#TOC-Registered-Names

2) CONTENT ACQUISITION

Regarding content acquisition, MS receives the main content (MPEG2-TS) with the embedded TEMI descriptors. As indicated above, the *location descriptor* contains the URL of the aforementioned XML file. Hence, the MS is able to provide the information about the related available contents to the connected CSs via the established bi-directional communication channel. Then, the CSs will be able to receive them via broadband networks and present them to the user.

3) IDES SYNC SOLUTION

In the adopted IDES solution, an M/S scheme is followed to get inter-stream synchronization (the broadcasted content played by the MS is the main content and is considered as the master stream; whereas the broadband complementary contents are considered as slave streams). The MS periodically shares its current playing content's (NTP-formatted) timestamp by extracting it from the timeline descriptor (together with the current NTP-formatted global clock's time) with the connected CSs via the communication channel [5]. As WebSocket [38] is the used technology for communications, messages are transmitted via TCP and unicast, although more efficient ways, such as UDP and multicast, could be used in specific situations, as when sharing timestamping information to several connected CSs. Either way, when any CS receives playout timing information from the MS, it can compute the value of the asynchrony between its playout (or presentation) process and the MS's one, and then, it will decide whether to adjust its playout process or not (depending on whether that value exceeds a configured threshold or not, respectively). CSs have been configured with a specific asynchrony threshold (*IDES threshold*). When a CS computes the value of the asynchrony between the MS's content playout process and its own process (*IDES async*), the CS compares it to the *IDES threshold*. If the value of the *IDES async* exceeds the *IDES threshold*, the CS makes a playout adjustment (explained in the next subsection). More detailed information about this IDES solution can be found in [5].

4) MEDIA PLAYOUT ADJUSTMENTS

Different techniques can be used to make playout adjustments. On the one hand, simple Skips and Pauses (S&P) in the playout can be made (aggressive adjustments). Even though they are noticeable for the users, they are extensively used. On the other hand, smooth playout adjustments can be made by using Adaptive Media Playout (AMP) techniques ([39], [40]). This type of adjustment consists in modifying the playout rate (i.e., by smoothly increasing or decreasing it) in order to maintain asynchrony values under the *IDES threshold*.

C. IDMS FEATURES

Broadcast delivery technologies are unidirectional. So, in multi-user hybrid scenarios requiring IDMS functionalities broadband delivery technologies will be used for bi-directional communications.

In this work, two different approaches have been designed in order to achieve IDMS, by following two different sync control schemes: an SMS centralized scheme, as in [4]; and a distributed scheme, DCS. The M/S centralized scheme has been discarded as previous works concluded that the use of this scheme for IDMS scenarios should require additional adaptive techniques to successfully achieve a synchronized state. An example of these techniques consists of buffer fullness monitoring and control in order to avoid buffer's underflow or overflow in long sessions when the master IDMS Client is significantly advanced or lagging regarding the other slave IDMS clients [24].

1) SYNC CONTROL SCHEMES: SMS VS DCS

In this section, both the centralized and distributed control schemes (SMS and DCS, respectively) for the IDMS solution are described. A set of control messages have been designed in order to create, join and leave a session, as well as to achieve accurate IDMS sync.

In an SMS scheme [4], two different entities are involved (Figure 4): one IDMS Manager and several IDMS Clients.

In the DCS scheme, only one type of entity is involved (Figure 5): the IDMS Client.

In the following subsections the defined types of messages, the procedures to create, join or leave a group session by

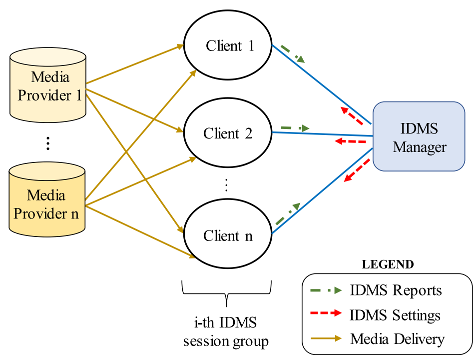

FIGURE 4. Proposed centralized SMS control scheme.

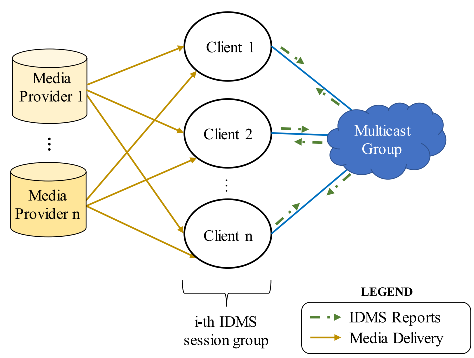

FIGURE 5. Proposed distributed control scheme.

IDMS Clients, as well as the exchange of the involved messages to achieve IDMS are explained.

Next, the proposed control messages for session management and for sync purposes in both schemes are presented. These control messages have been designed to be compatible with any of the proposed control schemes (with small variations in some fields related to specific-scheme requirements) in order to design a generic solution. Table 1 sums up the defined messages, including their names, in which control scheme they can be used, the sending direction and their purpose (SM: session management or S: synchronization).

The syntax of the aforementioned set of messages follow the structure presented in Figure 6. The three first fields are common for all the proposed messages. The first one identifies the type of message. The second one contains the identifier of the sending IDMS Client. The third one stores a checksum value, calculated with the data of the first two fields, following RFC 1071 [41].

IDMS Msg Type	*IDMS Client Id*	*Checksum*	*Other fields (if required)*

FIGURE 6. Generic format of the proposed set of IDMS messages.

Next, each proposed message is explained.

a: SESSION MANAGEMENT MESSAGES

i) IDMS HELLO REQUEST

This message (Figure 7), only contains the above common fields: including the *Hello Request* message type, the identification of the sending IDMS Client and the checksum value.

IDMS Hello Request	*IDMS Client Id*	*Checksum*

FIGURE 7. Format of the IDMS Hello Request message.

ii) IDMS HELLO RESPONSE

When an IDMS Client receives an IDMS Hello Request message, it must immediately answer it with an IDMS Hello Response message (in DCS, only if a response from another IDMS Client has not been received yet). This message (Figure 8) also includes the common fields: including the type of the message as a *Hello Response*; the identifier of the sending IDMS Client; and the checksum value.

IDMS Hello Response	*IDMS Client Id*	*Checksum*

FIGURE 8. Format of the IDMS Hello Response message.

iii) IDMS CREATE SESSION

This message (Figure 9), also contains the common fields: including the *Create Session* message type, the identification of the sending IDMS Client wanting to create a new session, and the checksum value.

TABLE 1. Set of the proposed IDMS control messages.

IDMS Message	Scheme	Direction	Purpose
Hello Request	SMS & DCS	IDMS Manager → IDMS Client (in SMS) IDMS Client → Multicast IP (in DCS)	SM (Checking connectivity)
Hello Response	SMS & DCS	IDMS Client → IDMS Manager (in SMS) IDMS Client → IDMS Client (in DCS)	SM (Response to a Hello Request)
Create Session	SMS	IDMS Client → IDMS Manager	SM (Create a new session)
Create Session ACK	SMS	IDMS Manager → IDMS Client	SM (Acknowledgement of session creation)
Join	SMS & DCS	IDMS Client (joining an IDMS session) → IDMS Manager (in SMS) IDMS Client (joining an IDMS session) → IDMS Clients (in DCS)	SM (Inform the rest of IDMS Clients that the Client IDMS is joining the session)
Leave	SMS & DCS	IDMS Client (leaving an IDMS Session) → IDMS Manager (in SMS) IDMS Client (leaving an IDMS Session) → IDMS Clients (in DCS)	SM (Inform the rest of IDMS Clients that the Client IDMS is leaving the session)
Report	SMS & DCS	IDMS Clients → IDMS Manager (in SMS) IDMS Clients → IDMS Clients (in DCS)	S (Send timing information to compute the session asynchrony)
Settings	SMS	IDMS Manager → IDMS Clients	S (Inform IDMS Clients that a playout adjustment may be required)

IDMS Create Session	*IDMS Client Id*	*Checksum*

FIGURE 9. Format of the IDMS Create Session message.

iv) IDMS CREATE SESSION ACK

This message (Figure 10) contains the three common fields (type of message, IDMS Client identifier and checksum value) and an additional field containing the identifier of the created IDMS Group (i.e., session). In case that the session cannot be successfully created for any reason (e.g., an IDMS Manager error when storing the required data to create the session), this field will contain a null value.

IDMS Create Session ACK	*IDMS Client Id*	*Checksum*	*Group id*

FIGURE 10. Format of the IDMS Create Session ACK message.

v) IDMS JOIN

This message (Figure 11) contains the three common fields (type of message, IDMS Client identifier and checksum value), as well as a field where the identifier of the group the IDMS Client is requesting to join is contained; a field indicating the number of list items included in this message; and a list of items related to extra information about the IDMS Client. Specifically, these adjacent fields are based on the RTCP Source Description (SDES) message, defined in [42]. Table 2 defines the available items. Notice that the only mandatory field to send is the NAME item.

IDMS Join	*IDMS Client id*	*Checksum*	Group id	*Number of items*	*List of items [...]*

FIGURE 11. Format of the IDMS Join message.

TABLE 2. IDMS join items

item	id	Description
NAME	1	[Mandatory] Friendly name of the user
EMAIL	2	[Optional] Stores user's email
PHONE	3	[Optional] Stores user's phone number
LOC	4	[Optional] Geographic user location, degree of detail depending on the app
TOOL	5	[Optional] Stores the name of the app

vi) IDMS LEAVE

In this message (Figure 13), the common fields are contained, indicating the message type, the identification of the IDMS Client and a checksum to validate the first two fields, as well as an additional field to identify the session which the IDMS Client wants to leave.

b: SYNC MESSAGES

i) IDMS REPORT

This message (Figure 14) contains the three common fields, an five additional ones. The fourth field, includes the group

item id	*length*	*payload*

FIGURE 12. **Format of the items inside an IDMS Join message.**

IDMS Leave	*IDMS Client id*	*Checksum*	*Group id*

FIGURE 13. **Format of the IDMS Leave message.**

IDMS Report	*IDMS Client id*	*Checksum*	*Other fields for IDMS Report*

Other fields for IDMS Report:

Group id	R_{seq} *nº*	*Content timestamp*	*Presentation timestamp*	*Transmission timestamp*

FIGURE 14. **Format of the IDMS Report.**

identifier and the fifth field stores the current sync round value (this concept will be explained later). Finally, the three remaining fields contain NTP-formatted timestamps: content embedded timestamp, the time when that content is to be presented and the time when the message has been transmitted.

ii) IDMS SETTINGS

In this message (Figure 15), the fields are similar to the *IDMS Report* message, with two differences: the first field identifies this message as an *IDMS Settings* and there is no field to store the time when the message is transmitted.

IDMS Settings	*IDMS Client id*	*Checksum*	*Other fields for IDMS Settings*

Other fields for IDMS Settings:

Group id	R_{seq} *nº*	*Content timestamp*	*Presentation timestamp*

FIGURE 15. **Format of the IDMS Settings message.**

2) CREATING AN IDMS SESSION

On the one hand, when the SMS scheme is used, the *IDMS Create Session* message is sent by the IDMS Client which wants to create the session to inform the IDMS Manager of this purpose. In particular, the IDMS Client knows the location of the IDMS Manager because it is signaled in the XML file (among other useful information, see Figure 16), whose URL is embedded in the TEMI location descriptor, as described in [4] and [5].

Then, the *IDMS Create Session ACK* is sent by the IDMS Manager and informs if the session has been created successfully or not, by indicating it in the transmitted message in the corresponding field (i.e., a valid or a null value in the *Group id* field, respectively).

On the other hand, when implementing the distributed DCS scheme, multicast communications are used. The IP multicast address is used as identifier of the IDMS group (or session). When an IDMS Client wants to create a new session, it needs to find an available (free) multicast IP address in a specific range (how to manage the multicast IP addresses by the service provider is out of the scope of this paper). It chooses a random multicast address in that range and sends an *IDMS Hello Request*, in order to check if it is free or not. The message is sent inside a datagram with that multicast IP address as the destination address. If the selected address is already being used by another IDMS session, the IDMS Clients in that session will send a response (*IDMS Hello Response* message). This response will indirectly notify to the creator of the new shared session that the selected multicast IP is not available. This type of message is sent three times within a (configurable) short period of time, in order to ensure that no shared session is using the selected multicast IP address. It is sent up to three times to confirm that, if no response is received, this is not due to packet loss issues. The process will be repeated until a free IP multicast address is found. Figure 17 shows the explained process.

Once the IDMS Client has found a free IP multicast address, the IDMS session is considered as established.

```
<Hybrid Media Contents File>
[…]
<MEDIA id = "media1" media_type = "AV" media_format = "h264/aac" metadata =
"alternative_view/english" temi_init = "36992554712919070 22">
<source protocol = "http/dash" uri = "http://
official_website.com/multicam/dash/cam1.mpd"/>
</MEDIA>
[…]
<WEB id = "web1" protocol = “http” media_type = “website” media_format =
“html5” metadata = “url_event/english” uri =
"http://official_website.com/hybrid_IDMS_app"/>
[…]
<IDMS id = "idms-mgr" protocol = “websocket” metadata = “idms manager uri”
uri = "http://official_website.com:port/idms_manager1"/>
[…]
<CLOCK id = "global-clock" protocol = "ntp" media_type = "time" media_format
= "64_bit_ntp_time" metadata = "" uri = "ntp.specific_ntp_server.com"/>
<LASTUPDATE protocol = "http" media_type = "time" metadata = "" format =
"mm/dd/yyyy-hh:mm:ss" value = "09/08/1999-23:59:59"/>
[…]
</ Hybrid Media Contents File >
```

FIGURE 16. **Example of the XML file with metadata for hybrid media delivery and IDMS functionalities.**

3) JOINING AN IDMS SESSION

Once an IDMS session is created, the creator IDMS Client can share the required information to join the session (*Group id* in SMS and multicast IP address in DCS) with other IDMS Clients (i.e., users) in order to invite them to join the session (e.g., through any popular communication app, such as e-mail, WhatsApp, Facebook or Twitter). Once an IDMS Client has joined a session, it can also share that information to invite other IDMS Clients.

Independently from the adopted control scheme, when an IDMS Client wants to join a session, it will send the *IDMS Join* message.

In the SMS scheme, the IDMS Client will send this message to the IDMS Manager. Moreover, it will have to specify the group identification in the corresponding field and provide the information contained in the items list from Table 2. Once the IDMS Manager receives this message, immediately, it will respond the IDMS Client with an *IDMS Settings*

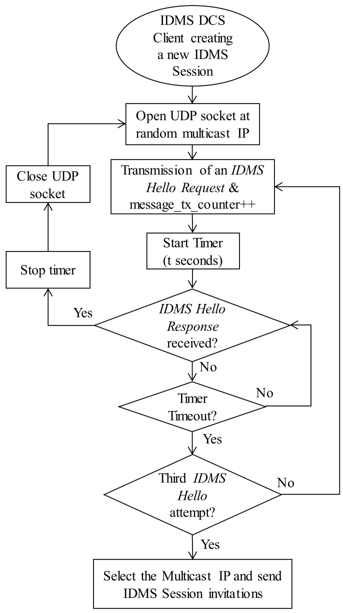

FIGURE 17. **Workflow of an DCS IDMS Client creating a new IDMS session.**

message. This way, the IDMS Client will know that it has successfully joined the session and will be able to make a first playout adjustment to achieve IDMS as soon as possible.

In the DCS scheme, the IDMS Client will send the join message to the multicast IP address used in that session. In this case, the IDMS Client also provides extra information with the item list from Table 2. However, the group id field is not used, as it is not necessary,

4) LEAVING AN IDMS SESSION

This type of message, in charge of informing that the IDMS Client wants to leave an IDMS session, is the same for both control schemes.

Regarding the SMS scheme, the leave message is sent to the IDMS Manager. Then, the IDMS Manager is able to identify the IDMS Client and deletes all its related information from the group session.

Regarding the DCS scheme, the leave message is sent to the multicast IP address used in that session. Then, the active IDMS Clients which receive this message will delete the leaving IDMS Client's related information.

5) IDMS SYNC SOLUTION

In the SMS scheme, the IDMS Manager, computes the asynchrony of the involved IDMS Clients in a session by processing their received *IDMS Report* messages (explained in more detail in [4]). 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 *Sync Round*. The *Sync Round* ends when the timeout triggers or when the IDMS Manager receives the *IDMS Reports*from all the involved IDMS Clients in a session. After that, the current asynchrony of the session (*session async*) is calculated and, if it exceeds the maximum acceptable session asynchrony (*session threshold*), reference playout timing values (*sync reference*) are calculated and sent to the IDMS Clients (Figure 18). After the IDMS Client receives the *IDMS Settings* message, if the difference between its playout timing and the timing of the *sync reference* (i.e., the *IDMS internal async*) exceeds a configurable maximum acceptable asynchrony (*IDMS internal async threshold*), the IDMS Client will have to make and adjustment to its playout process in order to achieve IDMS.

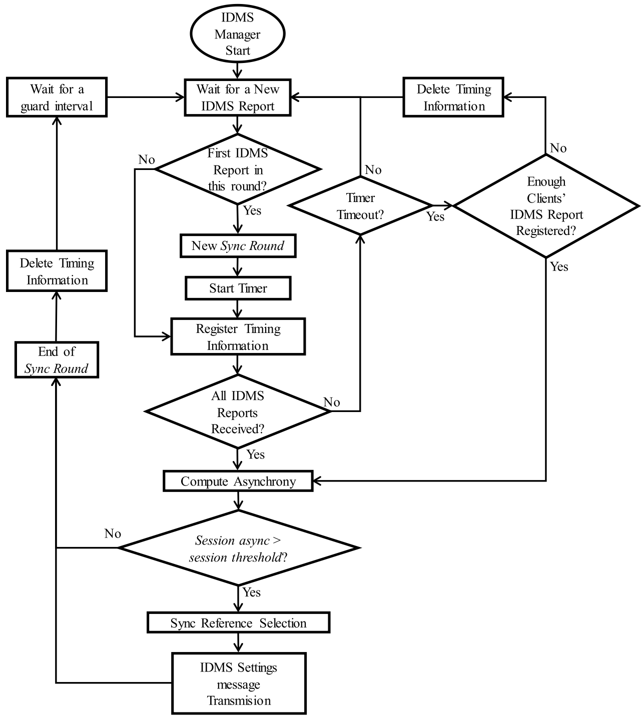

FIGURE 18. **IDMS Manager Workflow in the SMS control scheme.**

In the DCS scheme, unlike in the SMS scheme, there is no IDMS Manager entity. Thus, all the involved logic to compute the *Session async* is contained in each IDMS Client. Each IDMS Client receives the IDMS report messages from the rest of IDMS Clients an then it calculates the asynchrony of the session (*session async*). Like in the SMS scheme, if the *session async* exceeds the maximum acceptable session asynchrony (*session threshold*), reference playout timing values (*sync reference*) are calculated. Then, if the difference between its playout timing and the *sync reference* (i.e., the *IDMS internal async*) exceeds a configurable maximum acceptable asynchrony (*IDMS internal async threshold*), the IDMS Client will have to make an adjustment to its playout process in order to achieve IDMS. Figure 19 describes the behavior of an IDMS Client who joins an already existing IDMS session (the behavior of an IDMS Client creating an IDMS Session has been explained previously, see Figure 17). After joining successfully an IDMS session, the IDMS Client performs two tasks simultaneously. On the one hand, every $P_{IDMS} \pm \Delta$ seconds, an *IDMS Report* is

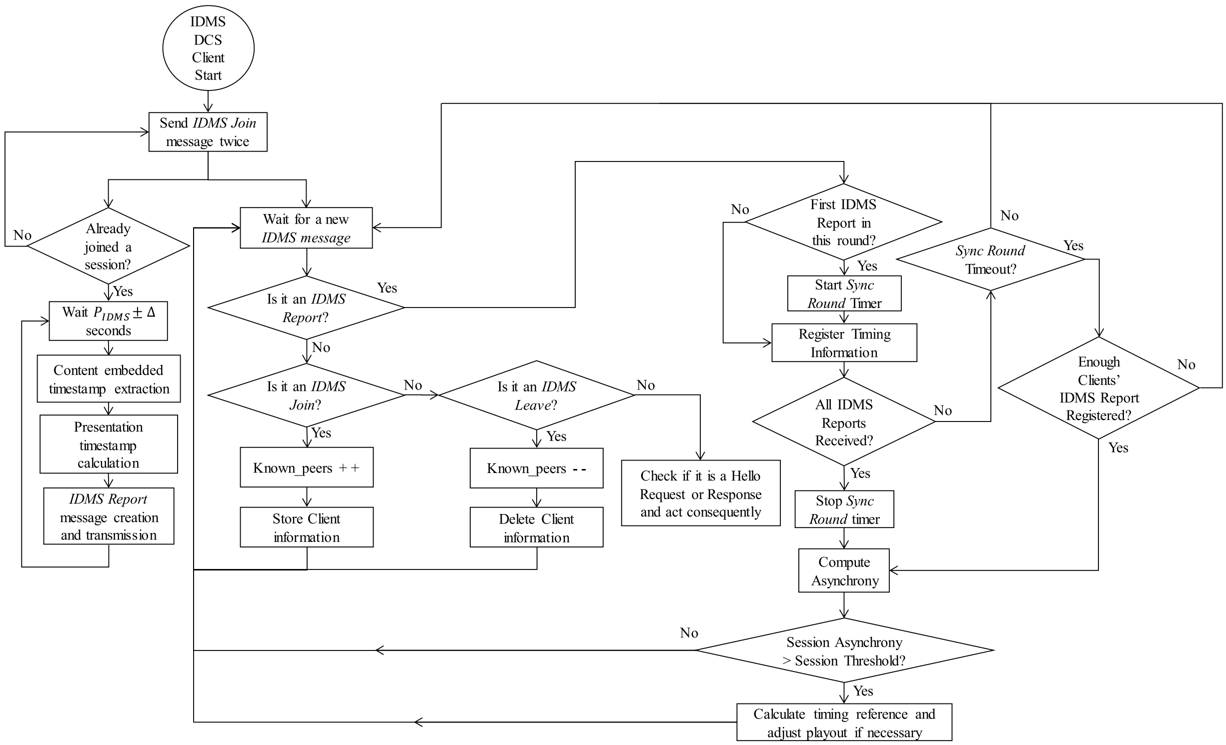

FIGURE 19. Simplified workflow of a DCS IDMS Client (IDMS Hello Response and Request are omitted).

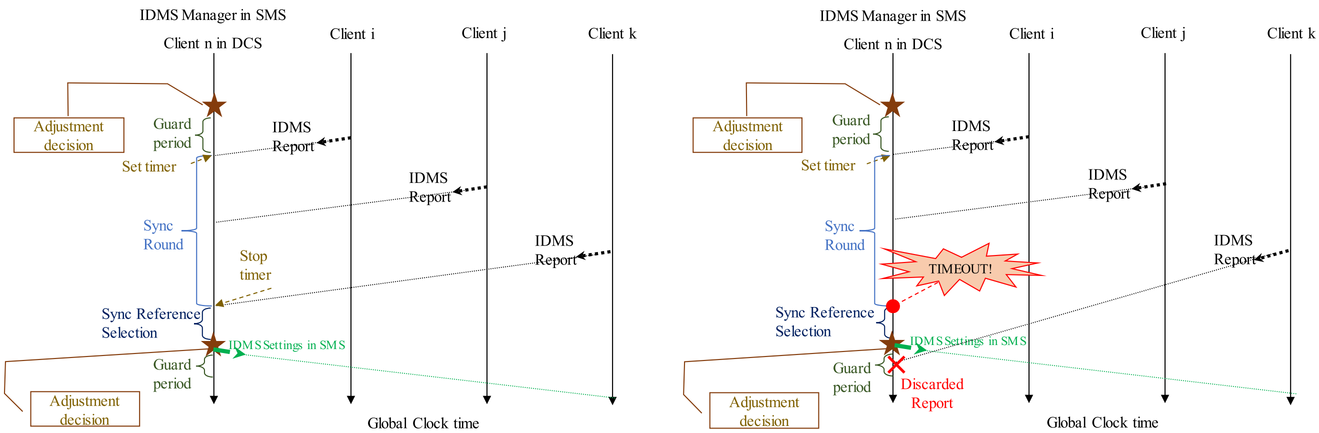

FIGURE 20. Normal (left) and undesired (right) situations.

generated and transmitted. P_{IDMS} is the time period between two consecutive transmissions of an *IDMS Report* and Δ is a randomly generated small value to avoid network congestion. On the other hand, the IDMS Client waits for new IDMS messages. When it receives an IDMS message, it behaves according to the type of that message. Note that in Figure 19, behaviors when *IDMS Hello Request/Response* messages are received have been omitted for simplicity. As explained, when the IDMS Client receives an *IDMS Hello Request*, it will immediately answer it with an *IDMS Hello Response* (in DCS, only if a response from another IDMS Client has not been received yet).

Additionally, Figure 20 describes a normal and an undesired situation for the proposed control schemes. A normal situation is considered when all the involved IDMS Clients send their *IDMS Reports* within the *Sync Round*. When the IDMS Manager or an IDMS Client (in SMS or DCS, respectively) has collected all the reports from the other IDMS Clients, it calculates the value of the *Session async* and starts a new round. On the contrary, an undesired situation implies at least one IDMS Client sending its *IDMS Report* out of time. To avoid this situation, a timer (*Sync Round Timer*) is reset every time a new round is started. If the timeout is triggered and there are still pending IDMS reports, the session async will be calculated taking into account only the received reports of that round (if at least two have been received) and a new round will be started. Late IDMS reports for the previous sync round will be discarded.

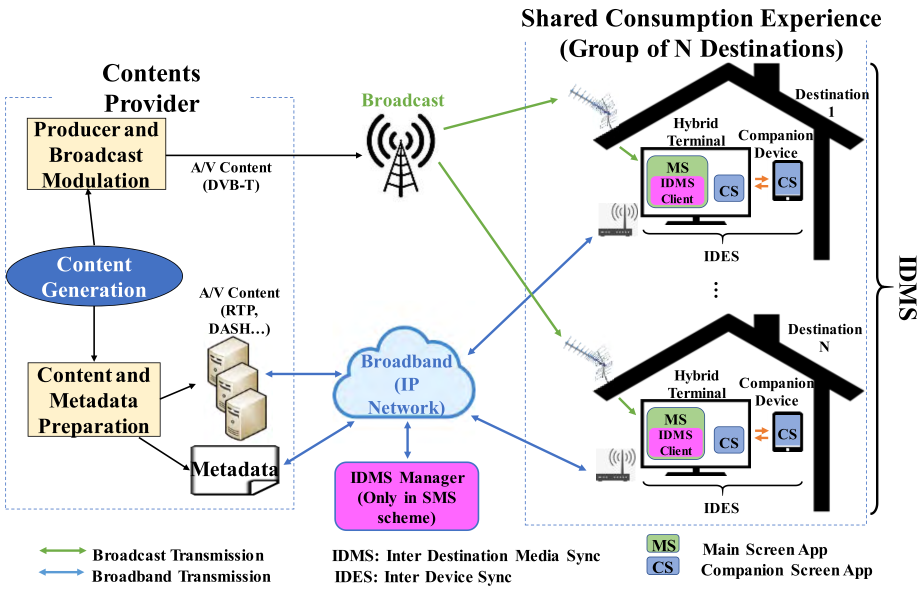

FIGURE 21. Overview of the end-to-end hybrid delivery platform where the IDMS + IDES solutions are integrated [4].

6) MEDIA PLAYOUT ADJUSTMENTS

In this case, similarly to IDES, different playout adjustment techniques can be applied, such as aggressive (S&P) or smooth (AMP) ones.

7) REFERENCE SELECTION STRATEGIES

Different reference selection strategies can be applied in order to adjust the playout processes to achieve accurate sync:

- Most Lagged User: in this strategy, the involved MS adjust their playout to the most lagged MS. This strategy presents few drawbacks in terms of performance if the buffer is not large enough (overflow situations could occur).
- Most Advanced User: in this strategy, the involved MS adjust their playout to the most advanced MS. This strategy might be unrecommended in some cases. Some MS can trigger a buffer underflow if the content associated to the playout point reference has not been received yet. Moreover, when using aggressive adjustments, according to obtained subjective results in [4] and [5], skips in playout are more annoying than pauses. This is one of the reasons why AMP is also proposed in this solution.
- Mean Playout Point: in this strategy, the involved MS converge in a mean playout position, calculated by taking all the received playout timings and computing the mean point. This strategy might involve forward and backward adjustments (depending on MS's playout positions) but a lower number of adjustments (with lower magnitude) are expected, as the target point is more likely to be closer (in general terms) to the playout point of every MS than in the two previous strategies.
- Optimum Playout Point: in this strategy, more factors than timestamped playout positions are considered. Parameters such as buffer occupancy, network delays (for each MS) or particular MS performance information can also be transmitted in order to compute a reference. This strategy might add some extra complexity to the calculations of the playout reference, so it can be easier to adopt in a centralized scheme rather than in a distributed scheme, as the latter is more complex.

IV. PLATFORM IMPLEMENTATION

In order to validate the proposed sync mechanism, which involves the combination of IDMS (SMS or DCS) and IDES solutions, the end-to-end hybrid delivery platform presented in [4] has been extended and adapted. This section describes the process that has been carried out to integrate the proposed sync mechanism in the platform. In particular, this platform supports any technology using the MPEG2-TS encapsulation format, such as HLS, DASH, RTP/RTCP and DVB, and the proposed sync mechanism works with all of them. Besides, by following HbbTV [1] recommendations (particularly Section 13.5.3) the buffer size of the IDMS Client's main content player (i.e., the MS) has been set to store approximately 12s of content at a 20Mbps rate, which is enough for most of the current scenarios regarding broadcast delay [8], [10].

Figure 21 shows an overview of the platform, in which all the involved modules and entities can be observed. As shown, two well differentiated parts have been implemented. On the left, the Contents Provider part is presented. That part is in charge of generating, encoding, encapsulating, signaling, storing and transmitting all the involved media contents.

It can support different media formats and delivery technologies as long as they carry MPEG2-TS streams, where the global timeline is embedded. Note that, in addition to multimedia content, this part also stores the XML file with the signaling and metadata of additional related media content.

On the right side of the platform (Figure 21), there is the Users' part, including the consumption devices. The extended functionalities for IDMS and IDES purposes are mostly performed in this side. However, depending on the adopted IDMS control scheme, some of the logic can be located "outside" of this part. Specifically, when adopting the SMS scheme, the IDMS Manager can be placed in the Contents Provider part or in an independent location [4]. However, as explained in the previous section, when adopting the DCS scheme, all the involved calculations are included in every IDMS Client.

A. SPECIFIC IDMS FEATURES

As explained in previous sections, in order to extend IDMS features to the platform presented in [5], some logical functionalities have to be implemented. *Session async* must be calculated by an independent entity (in SMS) or by the connected IDMS Clients (in DCS). Therefore, in order to compare both control schemes, a different approach has to be developed for each scheme. On the one hand, to achieve IDMS through an SMS scheme, a WebSocket server (i.e., the IDMS Manager) is adopted and implemented with the required IDMS functionalities (i.e., managing different IDMS sessions, identifying users, computing the *session async*, receiving *IDMS Report* messages, sending *IDMS Settings* messages, etc.) [4]. On the other hand, when adopting a DCS scheme, all the IDMS functionalities related to the calculation of the *session async* are included in every IDMS Client.

V. EVALUATION AND RESULTS

The evaluation has been carried out in order to objectively validate the sync mechanism performance and the achieved sync accuracy. In this section, first, the selected use case is described and then, the evaluation and the obtained results are presented and discussed.

A. SOCIAL MULTISCREEN AND MULTIVIEW TV SCENARIO

In particular, the use case which has been selected to carry out the evaluation has been the Social TV with multi-view (or multi-camera) content and a private chat service. This combined scenario has widely been assessed (i.e., Social TV and Multiscreen Multi-view/Multi-screen use cases separately) as interesting and willing to be used by consumers as stated in [4]–[6]. Both Social TV use case and Multi-view/Multiscreen use case were positively valued in those works. In order to visually assess the IDMS and IDES accuracy, the number of each video frame has been overlaid in all the related video contents and they all start at exactly the same instant. Moreover, although it has not been objectively assessed in this work, a synchronized chat tool was also implemented (the one already described, developed and subjectively evaluated in [4]). The features of the media content used for the evaluation purposes is summarized in Table 3. In that Table, Tech. stands for Technology, Enc. for Encoding and F.R for Frame Rate.

TABLE 3. Media content information.

Delivery	Tech.	Views	Enc.	Resolution	F. R.
Broadcast	DVB-T	1	H.264 + AAC	1920x 1080	25 fps
Broadband	DASH	4	H.264 + AAC	1920x 1080, 1280x 720, 854x480, 640x360, 426x240	25 fps

On the one hand, one view has been generated to be transmitted via broadcast, by using DVB-T technology. On the other hand, four different views (including the broadcasted one) have also been generated to be transmitted via broadband by using DASH technology. This allows the user to change to a different view when consuming the broadband content, thus being able to watch the same event through a different point of view in a synchronized way. The used XML file, whose URL has been embedded in the MPEG2-TS broadcasted content, can be seen in Figure 22.


```
<Hybrid Media Contents File>
<MEDIA id = "media1" media_type = "AV" media_format = "h264/aac" metadata =
"main_view/spanish" temi_init = "3699255471291907022">
<source protocol = "http/dash" uri =
"http://158.42.X.X/multicam/dash/cam1.mpd"/>
</MEDIA>
<MEDIA id = "media2" media_type = "AV" media_format = "h264/aac" metadata =
"left_view/spanish" temi_init = "3699255471291907022">
<source protocol = "http/dash" uri =
"http://158.42.X.X/multicam/dash/cam2.mpd"/>
</MEDIA>
<MEDIA id = "media3" media_type = "AV" media_format = "h264/aac" metadata =
"alternative_front_view/spanish" temi_init = "3699255471291907022">
<source protocol = "http/dash" uri =
"http://158.42.X.X/multicam/dash/cam3.mpd"/>
</MEDIA>
<MEDIA id = "media4" media_type = "AV" media_format = "h264/aac" metadata =
"right_view/spanish" temi_init = "3699255471291907022">
<source protocol = "http/dash" uri =
"http://158.42.X.X/multicam/dash/cam4.mpd"/>
</MEDIA>
<IDMS id = "idms-mgr" protocol = “websocket” metadata = “idms manager uri
SMS” uri = "http:// 158.42.X.X:YY/idms_manager1"/>
<CLOCK id = "global-clock" protocol = "ntp" media_type = "time" media_format =
"64_bit_ntp_time" metadata = "" uri = "158.42.X.X "/>
<LASTUPDATE protocol = "http" media_type = "time" metadata = "" format =
"mm/dd/yyyy-hh:mm:ss" value = "31/10/2018-23:59:59"/>
</ Hybrid Media Contents File >
```


FIGURE 22. Used XML file with the required metadata for hybrid media delivery and IDMS functionalities.

B. EVALUATION SETUP

The evaluation has been set up within the Polytechnic University of Valencia's Campus, involving four labs, each with 2 consumption devices (an MS and a CS).

Four different scenarios have been implemented by combining SMS/DCS schemes with Ideal/Loss+Delay situations.

Table 4 summarizes the forced delay and loss configuration in broadcast and broadband networks, which have been selected by following the same criteria as in [4] and [5].

TABLE 4. Forced delay and loss configuration for non-ideal conditions.

Delivery Technology	Forced Parameters
Broadcast (DVB-T)	Random Packet Loss Generation of 0.001% in the modulator software DekTec StreamXpress
Broadband (DASH)	Delay of 60±20 ms, following a normal distribution by using NetEm[9] in the Broadband Media Server

Figure 23 describes the scenario and devices used to perform the evaluation tasks. The role and the features of the devices and equipment are included in Table 5. Regarding the Contents Provider part, two computers have played the role of: 1) Broadcaster (DVB-T) and 2) Broadband Media Server. Specifically, the Broadband Media Server includes:

- An NTP global clock provider.
- A web server with stored DASH complementary content.
- The IDMS Manager, only active when the SMS scheme is used.
- A WebSocket based Chat server, active in the SMS scheme and optional in the DCS scheme (only if the multicast network is not used for this purpose).

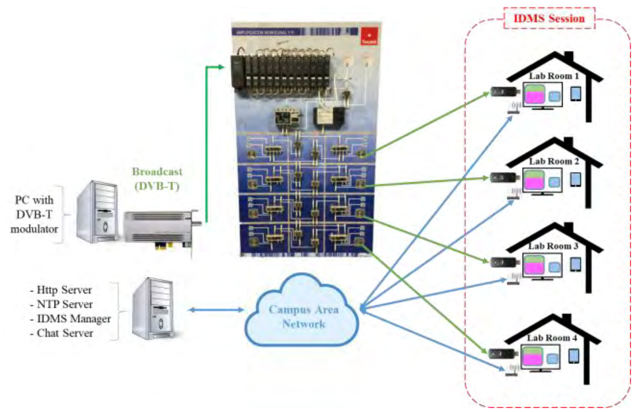

FIGURE 23. Implemented scenario for validating the proposed sync mechanism.

In the User's part, the four MS and CS are grouped in pairs (1 MS + 1 CS in each lab), each connected to the Campus Area Network via an independent Wi-fi Access Point. The Broadband Media Server was connected to the Campus Area Network.

In order to assess the sync accuracy achieved by the proposed mechanism, the values of the *session async*, the *IDMS internal async* and the *IDES async* have been registered. The results presented in this section have been obtained by analyzing these values. Additionally, while debugging the developed platform, a faster (but less accurate) method has also been used. It consisted of visually comparing the overlaid frame numbers (between MSs and between MS and CS, for IDMS and IDES, respectively), in order to ensure that after achieving a stable in-sync state, the configured threshold is not exceeded during the media session (see Figure 24).

[9]http://manpages.ubuntu.com/manpages/trusty/man8/tc-netem.8.html
[10]SMATV: Single Master Antenna Television (or Satellite Master Antenna Television)

TABLE 5. Role and specification of the used devices.

Role	Specification
DVB-T Modulation	*Intel Core i7-6700 8GB RAM W10 + DekTec DTA-2111 + DekTec StreamXpress*
DVB Reception Infrastructure	*Tecatel SMATV[10] Training Panel (with amplifiers, splitters, etc.)*
MS	*4x NUC Intel Celeron N3050 4GB RAM Ubuntu 14.04 + DVB USB rtl 2832 + {TV1: LG 32LF592U (32"); TV2: Panasonic TH42PX80E (42"); TV3 & TV4: Samsung Smart TV UE48H6200AWXXC (48")}*
CSs	*2x tablets Samsung Galaxy Tab S, 1 smartphone Samsung S5 1 smartphone Samsung S6*
Media Server	*Intel Xeon E5420 8GB RAM Ubuntu 14.04*
IDMS Manager (only in SMS)	*Same as Media Server*
WLAN Routers	*TP-Link AC1900 Wireless Dual Band Gigabit*

C. THRESHOLD CONFIGURATION

Regarding IDES, the *IDES threshold* has been configured to 80ms, as this asynchrony value is stated to be tolerable in this type of scenarios [43]. This value implies a maximum asynchrony of 2 frames between MS and CS players for 25fps content. As demonstrated in [5], this *IDES threshold* value provides accurate performance and satisfactory QoE results.

Regarding IDMS, as recommended in [44], the platform has been configured to ensure that the *Session async* is maintained below 400ms in stable conditions. Some pre-tests were done in order to select the appropriate values for thresholds. The *Session threshold*was initially set to 200ms, and then it was progressively decreased until the platform presented a poor performance (i.e., frequent unnecessary adjustments due to playout fluctuations). In the end, the S*ession threshold* has been set to 160ms, as this value provides accurate IDMS sync and does not affect users' QoE perception [4]. In particular, for 25 fps content, every 40ms a new frame is displayed. This means that, in the worst case, while enabling IDMS features, users in a shared session are consuming the media content with a difference of at most 4 frames.

D. EVALUATION RESULTS

In this section the obtained results are presented and discussed. Regarding the performance of the IDMS solution, a comparison between both SMS and DCS control schemes has been done, when used in combination with the IDES solution. The four described scenarios have been implemented

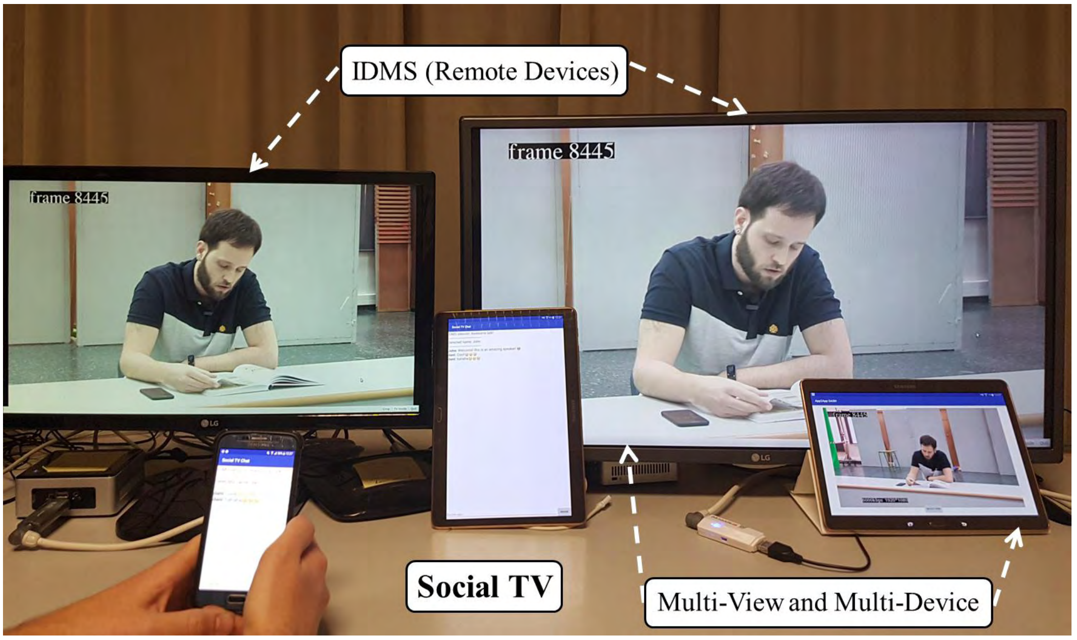

FIGURE 24. Appearance of the developed platform, 2 MS and 3 CS (2 with a chat tool and 1 with Multiview content).

to assess the proposed mechanism. Two reference selection strategies, to the most lagged and the mean playout point, have been applied. Moreover, ideal and realistic (i.e., non-ideal, with loss and delay) scenarios have been implemented. In Table 6, the evaluated cases are described. It includes, for each case, the type of sync (IDMS and/or IDES) the associated asynchronies have been registered for.

TABLE 6. Summary of the Evaluated Cases and Type of Registered Asynchrony.

	SMS		DCS	
	Ideal conditions	Non-ideal conditions	Ideal conditions	Non-ideal conditions
Most Lagged	*IDMS*	*IDMS, IDES*	*IDMS*	*IDMS, IDES*
Mean Playout Point	*IDMS*	*IDMS, IDES*	*IDMS*	*IDMS, IDES*

For each case, at least 10 measurements of approximately 3 minutes have been carried out in each device. This time duration has been enough for the involved devices to achieve sync stable states and to recover when undesired situations were detected (mostly in non-ideal scenarios when packet loss or delay could negatively affect the playout process of the involved devices).

1) IDMS PERFORMANCE

Figure 25 shows the mean values of *Session async*, with the 95% of confidence interval, for the four different cases that have been applied to DCS and SMS schemes. It can be seen that these mean values do not exceed the configured *Session threshold* (160ms), and only in the *DCS Most Lagged + Non-Ideal* case, the 95% confidence interval slightly exceeds it. This situation probably occurs because unexpected random packet loss or network delay negatively affects the playout processes. However, registered values prove the good performance of the solutions. Note that the values for DCS control scheme represented in this figure are the mean values of the registered values on each MS. Figure 26 shows the detailed values which have been registered in each MS, in the 4 evaluated cases when adopting the DCS control scheme. It can be observed that the four MSs have very similar *Session async* mean values for each case. However, as seen in Figure 25 and now observed in Figure 26, for the *most lagged* strategy and *non-ideal* case, the higher values correspond to MS1. As already mentioned, random generated errors may cause this imprecise behavior (although obtained values can be considered accurate enough).

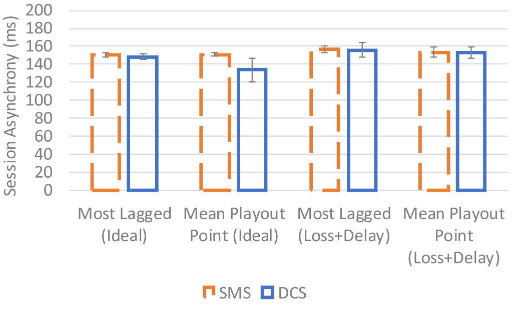

FIGURE 25. DCS and SMS mean session asynchrony registered values with a 95% confidence interval.

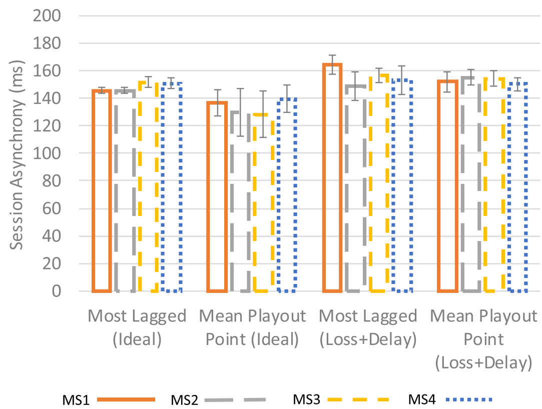

FIGURE 26. **Mean Session Asynchrony registered values for each MS, under the DCS control scheme with a 95% confidence interval.**

Figure 27 shows the mean value of the number of adjustments that have been performed during the registered sessions. This is a relevant parameter to consider, as it can affect users' perception of QoE. It is assumed that, while keeping under the configured *Session threshold*, the lower the number of made adjustments the better the performance of the platform and the better the expected QoE values. Paying attention to this figure, it can be observed that the SMS scheme has a better performance (less adjustments) than the DCS scheme and that the use of the *mean playout point* as reference is better than the use of the *most lagged* one. On the one hand, when using the *most lagged* one the mean number of adjustments in DCS is more than twice the mean number of adjustments made when using SMS. On the other hand, when using the *mean playout point* as reference, the mean number of adjustments is quite similar in both SMS and DCS control schemes (still a bit higher in the latter).

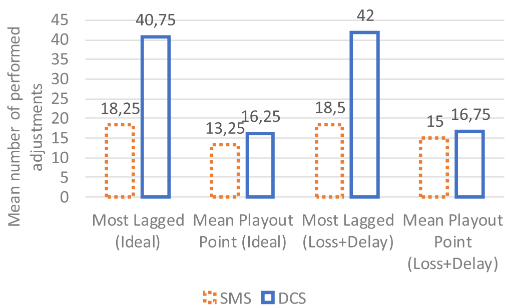

FIGURE 27. **Mean number of required adjustments by MSs depending on the evaluated case.**

2) IDES PERFORMANCE IN IDMS SCENARIOS

In this section, the evaluation of the IDES performance when combined with IDMS is presented. For it, only tests in *non-ideal* scenarios have been carried out. All of them involve broadcast random packet loss and broadband network delays, according to the configuration in Table 4. As described in Table 5, heterogeneous devices have been used, showing different performance in stable IDES states. Figure 28 illustrates the mean inter-device asynchrony (*IDES async*)

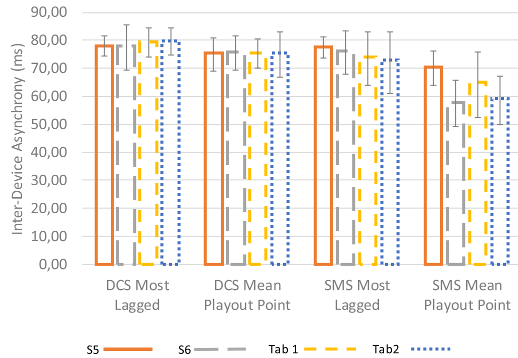

FIGURE 28. **Mean IDES async values registered by the CSs in each test.**

values, with the 95% confidence interval. They do not exceed the configured *IDES threshold*, although some of the values which are part of the confidence interval do exceed it. This can be due to playout variations in the MSs, as those devices are also adjusting their playout in order to achieve IDMS. Nevertheless, obtained mean values prove that the proposed sync mechanism is performing satisfactorily. It can also be observed (in Figure 28 and also in the following ones) that the Samsung S5 smartphone can be considered as the device with poorer performance: in both SMS reference strategies this device is clearly the one which has higher mean async value. Regarding DCS reference strategies, it is not the device with the highest mean async value, however there is not any device which has clearly the best mean asynchrony value. So, in general terms, the Samsung S5 can be considered the less accurate device in terms of achieved asynchrony, as it will be confirmed in the following Figures 30-33.

The Cumulative Frequency Distribution graphs for each case is presented in Figure 29. In this type of charts, it can be observed which one presents a better performance in terms of registered *IDES async* values frequency. The device with the steepest slope means that it has registered more lower async values. The case with the best performance is the SMS scheme with the Mean Playout Point reference strategy.

Finally, four additional measurements (one for each case, including forced loss and delays as specified in Table 4), of approximately 5 minutes, have been performed in order to illustrate the behavior of the CSs when switching the playout to a different view. In each case, one or two different devices perform the view switching. In Figures 30 to 33, it can be observed how the CSs behave differently depending on their resources. In particular, the Samsung S5 smartphone makes a significant number of adjustments (every time the measured asynchrony values exceed the configured threshold). It is the oldest device and thus, the one with worst technical specifications. The Samsung S6 smartphone and the two Galaxy Tab S tablets have better resources and, therefore, better performance. As CSs play DASH media content, once any device changes the view (i.e., the media source), the registered asynchrony value falls down. This can be observed when the *IDES async* is registered with high negative values, as the playout process of the main content in the co-located

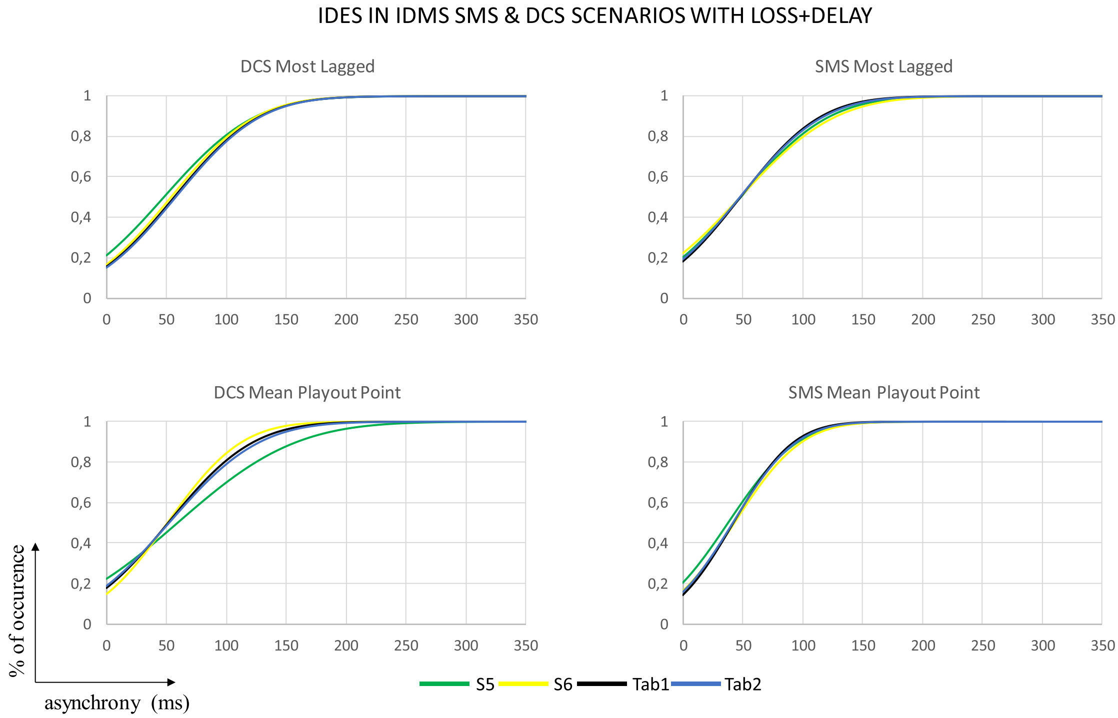

FIGURE 29. IDES CFD charts of each CS, classified by case.

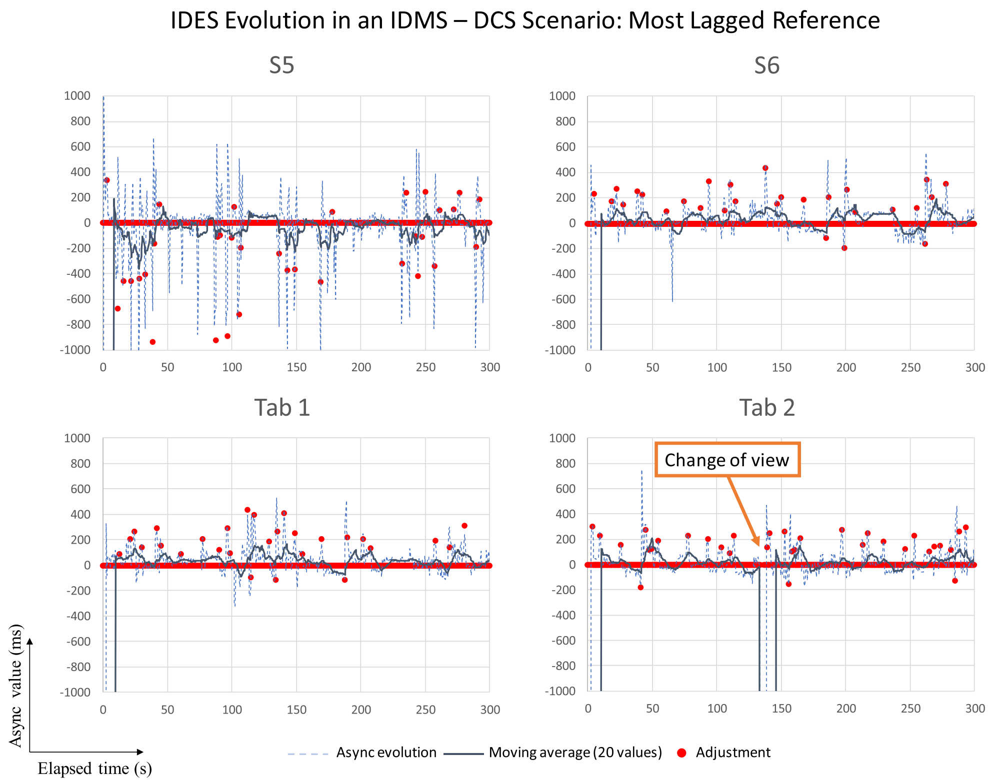

FIGURE 30. CS asynchrony evolution in the IDMS DCS – Most Lagged Reference scenario.

MS device is significantly advanced. Anyway, it is observed that after this first *IDES async* registration, an adjustment is performed and it quickly reaches an IDES stable state.

As it can be observed, in Figure 30, which corresponds to CSs' behavior under the IDMS DCS scheme with the *Most Lagged* reference, the Samsung Galaxy Tab S (Tab 2)

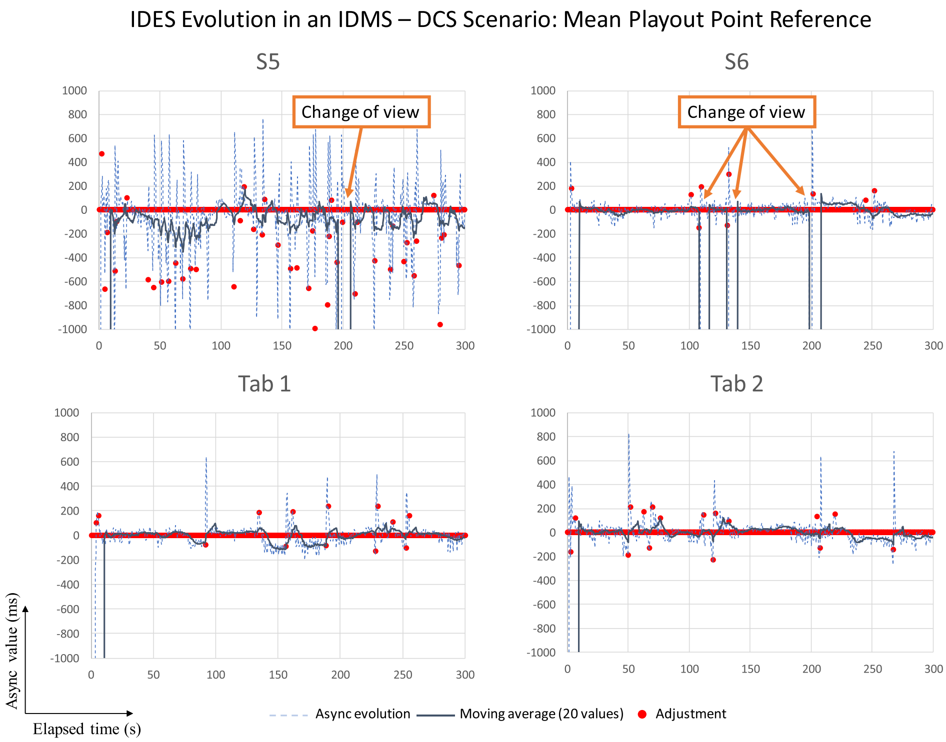

FIGURE 31. CS asynchrony evolution in the IDMS DCS – Mean Playout Point scenario.

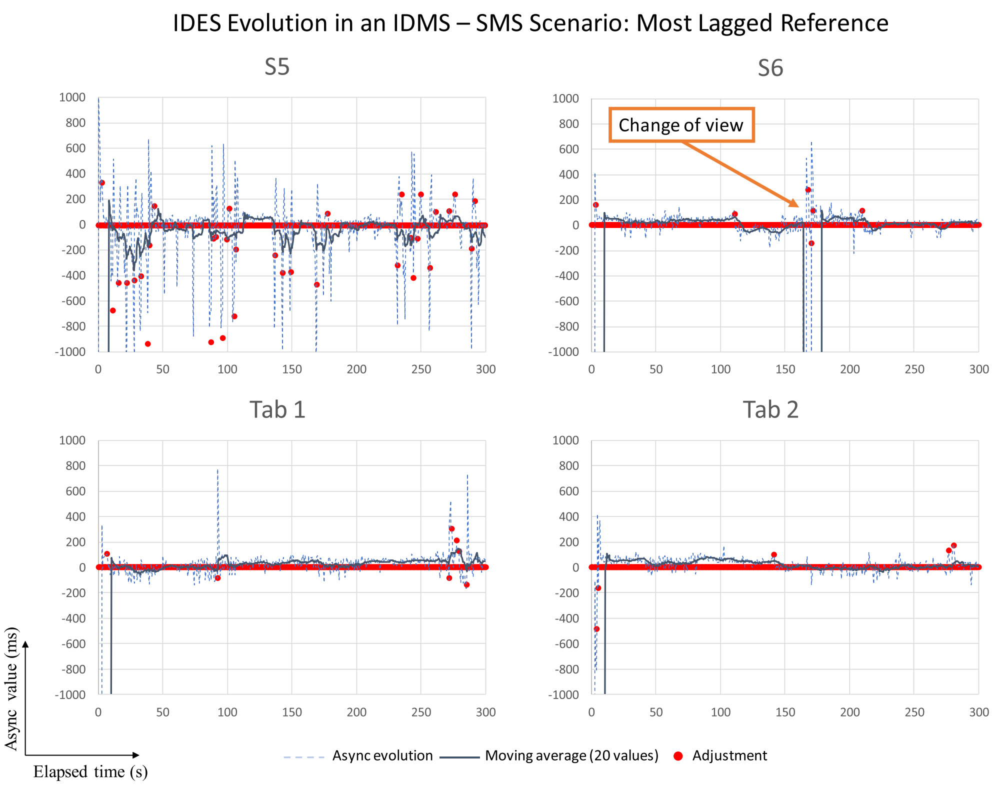

FIGURE 32. CS asynchrony evolution in the IDMS SMS – Most Lagged Reference scenario.

switches to a different channel once, at 140s approximately. In that picture, it can be seen how Samsung's devices S6 and the Galaxy Tabs S (both Tab 1 and Tab 2) have a good performance and, during the session, adjust their playout process with a similar frequency. However, the Samsung S5 has adjusted its playout more times, although running the same

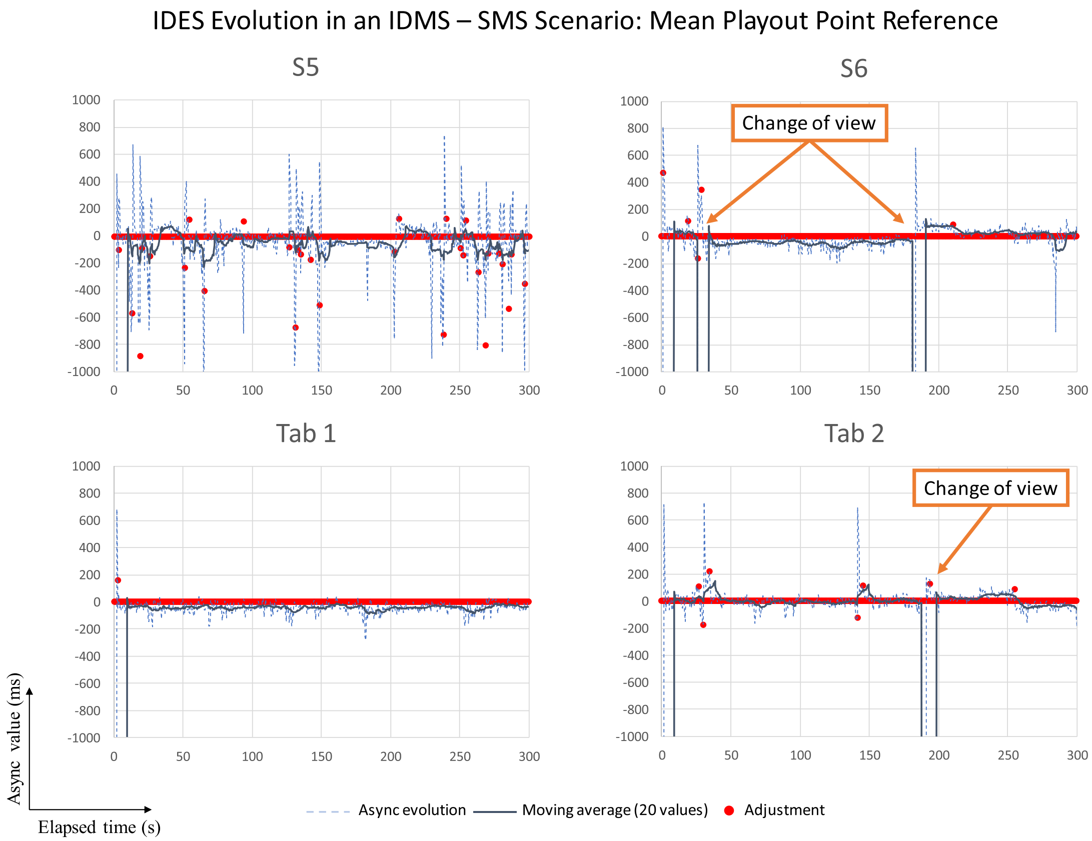

FIGURE 33. CS asynchrony evolution in the IDMS SMS – Mean Playout Point scenario.

application as the other devices. Making more adjustments imply worse performance, as the playout rate is unstable. This might be due to the technical specifications of the device, as mentioned above.

Regarding the IDMS DCS scheme with the *Mean Playout Point* reference, Figure 31 shows the CS' behavior under this scenario. For this case the smartphone Samsung S5 has switched the views at approximately 200s; and the smartphone Samsung S6 at approximately 110s and 130s and 200s. Although the switching in the S6 device could be considered excessive, it evidences the good performance of the solution when running under a device with good or enough resources. Similar to Figure 30, it can be seen how the Samsung S5 has the worst behavior, as the number of performed adjustments are significantly higher than in the other devices. Besides, comparing *Most Lagged* and *Mean Playout Point* references, it can be seen that for the latter, a smaller number of adjustments are required. This fact is supported with Figure 27, where it is stated that IDMS adjustments applied to the MSs are lower when adopting a *Mean Playout Point* strategy. Consequently, CSs need to adjust less times than when following the IDMS DCS *Most Lagged* strategy.

Figure 32 shows the CS' behaviors when the IDMS SMS *Most Lagged* strategy is adopted. For this case, only the Samsung S6 has switched the view once, at 160s approximately. Again, the Samsung S5 presents the worst performance, as the number of playout adjustments are more frequent than in the rest of devices. Moreover, if this centralized case is compared with the distributed one (i.e., IDMS DCS with *Most Lagged* strategy), it can be observed that when the session is under a centralized control scheme, less playout adjustments are made. This can also be noticed in Figure 27, where the number of adjustments when following the *Most Lagged* strategy is significantly lower under a centralized control scheme compared to the distributed control scheme, thus forcing less playout adjustments in the involved CSs.

Finally, Figure 33 shows the behaviors of the CSs under an IDMS SMS session with a *Mean Playout Point* reference. In this case, the Samsung S6 has switched the view twice at approximately 25s and 175s. Also, the Samsung Galaxy Tab S (Tab 2) has switched the view once, at approximately 200s. One more time, the Samsung S5 shows the worst performance, although it has not switched the view in this case. The rest of devices present similar number of adjustments, taking into account that two of them (S6 and Tab 2) have switched their view at some time. When comparing *Most Lagged* and *Mean Playout Point* strategies under the centralized scheme, it can be stated that the *Mean Playout Point*strategy forces less adjustments (see also Figure 31), although the difference is minimum, and it should not be considered as significative. However, when comparing control schemes, differences are significative: the SMS scheme induces a smaller number of adjustments, in the involved MSs and, consequently, in the co-located CSs.

As expected, the centralized SMS scheme provides a better sync accuracy, as only one module (i.e., the IDMS Manager) is in charge of calculating the *session async* and sending (if needed) a playout adjustment request. Unlike in DCS, where each IDMS Client has to calculate the *session async* and small differences in the collected timing data in one IDMS Client can cause more significant sync inaccuracies in the overall session. Moreover, the *Mean Playout Point* reference strategy has been proved to be less aggressive to the playout processes, as the frequency and magnitude of the playout adjustments are lower, providing a more stable playout.

VI. CONCLUSIONS AND FUTURE WORK

In this work, a sync approach, combining IDMS and IDES solutions, for Hybrid Multi-view and Multi-screen scenarios has been presented, implemented and evaluated. Regarding IDMS, the approach can include two different sync control schemes for IDMS (centralized or SMS and distributed or DCS) and different sync reference selection strategies. Apart from the sync approach, a set of messages have been defined regarding both shared session management and synchronization purposes, regardless the sync control scheme involved in the approach.

The approach has been integrated into an end-to-end hybrid delivery platform and has been objectively evaluated to test its performance and the achieved sync accuracy. Results prove that (under several ideal and non-ideal scenarios) IDMS stable states can be achieved with a Session async below 160ms, which is the configured *Session threshold* and is significantly lower than the recommended 400ms [44] in order to ensure that it never exceeds that value in stable realistic conditions. Simultaneously, IDES async values have been maintained below 80ms (*IDES threshold*) in the evaluated scenarios. Results also show evidence that, due to the availability of heterogeneous devices as CSs, not all can support satisfactory sync states although achieving and showing accurate IDES values, as other features, such as the frequency of performed adjustments might annoy final users and these kinds of issues must also be taken into account instead of only considering average sync values.

In conclusion, the approach including the combined sync solutions can be deployed in the market, as they are based on technologies and protocols already in-use. Furthermore, Digital Terrestrial Television is still widely used, as IP-only delivery technologies are still a very long way off. The combined use of hybrid technologies fits in with the demands of the modern consumer and many opportunities and challenges are yet to come.

Subjective evaluation of the approach is left for future work. A previous IDMS solution based on centralized SMS scheme was already evaluated with promising results in [4], and similar results are expected. As other future work, applications included in the platform will be optimized to perform better in many different devices. As an example, quality selection policy in the DASH player should take into account the performance or resources of the device (CS).

REFERENCES

[1] (2018). *Hybrid Broadcast Broadband TV (HbbTV) 2.0.1 Specification, HbbTV Association Resource Library*. [Online]. Available: https://www.hbbtv.org/resource-library

[2] Y. Miyashita, Y. Ishibashi, N. Fukushima, S. Sugawara, and K. E. Psannis, "QoE assessment of group synchronization in networked chorus with voice and video," in *Proc. IEEE TENCON*, Nov. 2011, pp. 192–196.

[3] M. Montagud, P. Cesar, F. Boronat, and J. Jansen, "Introduction to media synchronization (MediaSync)," in *MediaSync*, M. Montagud, P. Cesar, F. Boronat, and J. Jansen, Eds. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-65840-7_1.

[4] D. Marfil, F. Boronat, M. Montagud, and A. Sapena, "IDMS solution for hybrid broadcast broadband delivery within the context of HbbTV standard," *IEEE Trans. Broadcast.*, to be published, doi: 10.1109/TBC.2018.2878285.

[5] F. Boronat, D. Marfil, M. Montagud, and J. Pastor, "HbbTV-compliant platform for hybrid media delivery and synchronization on single- and multi-device scenarios," *IEEE Trans. Broadcast.*, vol. 64, no. 3, pp. 721–746, Sep. 2018.

[6] F. Boronat, M. Montagud, D. Marfil, and C. Luzón, "Hybrid broadcast/broadband TV services and media synchronization: Demands, preferences and expectations of spanish consumers," *IEEE Trans. Broadcast.*, vol. 64, no. 1, pp. 52–69, Mar. 2018.

[7] M. Montagud, F. Boronat, H. Stokking, and R. van Brandenburg, "Inter-destination multimedia synchronization: Schemes, use cases and standardization," *Multimedia Syst.*, vol. 18, no. 6, pp. 459–482, Nov. 2012.

[8] W. Kooij, J. Stokking, R. van Brandenburg, and P.-T. de Boer, "Playout delay of TV signals: Measurement system design, validation and results," in *Proc. ACM TVX*, Newcastle, U.K., Jun. 2014, pp. 23–30.

[9] J. Jansen, P. Cesar, D. C. A. Bulterman, T. Stevens, I. Kegel, and J. Issing, "Enabling composition-based video-conferencing for the home," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 869–881, Oct. 2011.

[10] M. O. Van Deventer, H. M. Stokking, O. A. Niamut, F. A. Walraven, and V. B. Klos, "Advanced interactive television services require content synchronization," in *Proc. 15th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Bratislava, Slovakia, Jun. 2008, pp. 109–112.

[11] R. N. Mekuria, "Inter-destination media synchronization for TV broadcasts," M.S. thesis, Dept. Netw. Archit. Services, Fac. Elect. Eng., Math. Comput. Sci., Delft Univ. Technol., Delft, The Netherlands, Apr. 2011.

[12] *Network Model for Evaluating Multimedia Transmission Performance Over Internet Protocol*, document ITU-T Rec. G.1050, 2007.

[13] *Network Time Protocol Version 4: Protocol and Algorithms Specification*, IETF Internet Standard, document RFC 5905, 2010. [Online]. Available: https://tools.ietf.org/html/rfc5905

[14] *Specification for the Carriage of Synchronised Auxiliary Data in DVB Transport Streams*, document ETSI TS 102 823 V1.2.1, 2014.

[15] (2013). *Text of ISO/IEC 13818-1:2013/PDAM 6 Delivery of Timeline for External Data*. [Online]. Available: http://mpeg.chiariglione.org/standards/mpeg-2/systems/text-isoiec-13818-12013pdam-6-delivery-timeline-external-data

[16] M. Zorrilla, A. Martin, I. Tamayo, S. O'Halpin, and D. Hazael-Massieux, "Reaching devices around an HbbTV television," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (SBMSB)*, Beijing, China, Jun. 2014, pp. 1–7, doi: 10.1109/BMSB.2014.6873499.

[17] C. Howson, E. Gautier, P. Gilberton, A. Laurent, and Y. Legallais, "Second screen TV synchronization," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Berlin, Germany, Sep. 2011, pp. 361–365.

[18] R. van Brandenburg, and A. T. Veenhuizen, "Immersive second-screen experiences using hybrid media synchronization," Presented at the MediaSync Workshop, Nantes, France, Oct. 2013.

[19] V. Vinayagamoorthy, R. Ramdhany, and M. Hammond, "Enabling frame-accurate synchronised companion screen experiences," in *Proc. ACM TVX*, New York, NY, USA, 2016, pp. 83–92, doi: 10.1145/2932206.2932214.

[20] C. Ziegler, C. Keimel, R. Ramdhany, and V. Vinayagamoorthy, "On time or not on time: A user study on delays in a synchronised companion-screen experience," in *Proc. ACM TVX*, 2017, pp. 105–114.

[21] *Companion Screens and Streams; Part 2: Content Identification and Media Synchronization*, document ETSI TS 106 286-1, 2015.

[22] A. Domínguez, M. Agirre, J. Flörez, A. Lafuente, I. Tamayo, and M. Zorrilla, "Deployment of a hybrid broadcast-Internet multi-device service for a live TV programme," *IEEE Trans. Broadcast.*, vol. 64, no. 1, pp. 153–163, Mar. 2018, doi: 10.1109/TBC.2017.2755403.
[23] F. Boronat, J. Lloret, and M. García, "Multimedia group and inter-stream synchronization techniques: A comparative study," *Inf. Syst.*, vol. 34, no. 1, pp. 108–131, Mar. 2009.
[24] M. Montagud, F. Boronat, H. Stokking, and P. Cesar, "Design, development and assessment of control schemes for IDMS in a standardized RTCP-based solution," *Comput. Netw.*, vol. 70, pp. 240–259, Sep. 2014.
[25] Y. Ishibashi, A. Tsuji, and S. Tasaka, "A group synchronization mechanism for stored media in multicast communications," in *Proc. INFOCOM*, Washington, DC, USA, Apr. 1997, pp. 692–700.
[26] Y. Ishibashi and S. Tasaka, "A group synchronization mechanism for live media in multicast communications," in *Proc. IEEE GLOBECOM*, vol. 97, Nov. 1997, pp. 746–752.
[27] R. Arima, M. Sithu, and Y. Ishibashi, "QoE assessment of fairness between players in networked virtual 3D objects identification game using haptic, olfactory, and auditory senses," *Int. J. Commun., Netw. Syst. Sci.*, vol. 10, no. 7, pp. 129–141, Jul. 2017.
[28] M. Mauve, J. Vogel, V. Hilt, and W. Effelsberg, "Local-lag and time-warp: Providing consistency for replicated continuous applications," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 47–57, Feb. 2004.
[29] I. Vaishnavi, P. Cesar, D. Bulterman, O. Friedrich, S. Gunkel, and D. Geerts, "From IPTV to synchronous shared experiences challenges in design: Distributed media synchronization," *Signal Process., Image Commun.*, vol. 26, no. 7, pp. 370–377, Aug. 2011.
[30] F. Boronat, M. Montagud, H. M. Stokking, and O. Niamut, "The need for inter-destination synchronization for emerging social interactive multimedia applications," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 150–158, Nov. 2012, doi: 10.1109/MCOM.2012.6353695.
[31] M. O. van Deventer, H. Stokking, M. Hammond, J. Le Feuvre, and P. Cesar, "Standards for multi-stream and multi-device media synchronization," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 16–21, Mar. 2016.
[32] F. B. Seguí, J. C. G. Cebollada, and J. L. Mauri, "An RTP/RTCP based approach for multimedia group and inter-stream synchronization," *Multimedia Tools Appl.*, vol. 40, no. 2, pp. 285–319, Nov. 2008, doi: 10.1007/s11042-008-0208-1.
[33] R. van Brandenburg, H. Stokking, M. O. Van Deventer, F. Boronat, M. Montagud, and K. Gross, *Inter-Destination Media Synchronization (IDMS) Using the RTP Control Protocol (RTCP)*, IETF Standard, document RFC 7272, Jun. 2014.
[34] B. Rainer and C. Timmerer, "Self-organized inter-destination multimedia synchronization for adaptive media streaming," in *Proc. ACM MM*, Orlando, FL, USA, Nov. 2014, pp. 327–336.
[35] J. Belda, M. Montagud, F. Boronat, M. Martínez, and J. Pastor, "Wersync: A Web-based platform for distributed media synchronization and social interaction," in *Proc. ACM TVX*, Brussels, Belgium, Jun. 2015, pp. 9–10.
[36] NIELSEN. *The Nielsen Comparable Metrics Report: Q4 2016*. Accessed: Oct. 2018. [Online]. Available: http://www.nielsen.com/us/en/insights/reports/2016/the-comparable-metrics-report-q2-2016.html
[37] (2015). *DIAL (DIscovery And Launch) Protocol Specification, Version 1.7.2*. [Online]. Available: http://www.dial-multiscreen.org/dial-protocol-specification/DIAL-2ndScreenProtocol-1.7.2.pdf
[38] I. Fette and A. Melnikov, *The WebSocket Protocol*, IETF Standard, document RFC 6455, Dec. 2011.
[39] M. Montagud, F. Boronat, B. Roig, and A. Sapena, "How to perform AMP? Cubic adjustments for improving the QoE," *Comput. Commun.*, vol. 103, pp. 61–73, May 2017.
[40] B. Rainer and C. Timmerer, "Adaptive media playout for inter-destination media synchronization," in *Proc. 5th Int. Workshop Qual. Multimedia Exper. (QoMEX)*, Klagenfurt am Wörthersee, Jul. 2013, p. 44–45, doi: 10.1109/QoMEX.2013.6603206.
[41] R. Braden, D. Borman, and C. Partridge, *Computing the Internet Checksum*, document RFC 1071, Sep. 1988.
[42] *RTP: A Transport Protocol for Real-Time Applications*, IETF Internet Standard, document RFC 3550, 2003. [Online]. Available: https://tools.ietf.org/html/rfc3550
[43] R. Steinmetz, "Human perception of jitter and media synchronization," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 1, pp. 61–72, Jan. 1996.
[44] B. Rainer, S. Petscharnig, C. Timmerer, and H. Hellwagner, "Is one second enough? Evaluating QoE for inter-destination multimedia synchronization using human computation and crowdsourcing," in *Proc. 7th Int. Workshop Qual. Multimedia Exper. (QoMEX)*, Pylos-Nestoras, Greece, May 2015, pp. 1–6, doi: 10.1109/QoMEX.2015.7148107.

DANI MARFIL was born in Gandia, Spain. He received the B.Sc. degree in informatics technical engineering, the B.Sc. degree in telecommunications, and the M.Sc. degree in telecommunication technologies, systems and networks from the Universitat Politècnica de València, Spain, in 2011, 2015, and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Immersive Interactive Media R&D Group. He is also an Assistant Researcher and a Developer with the Immersive Interactive Media R&D Group. He has authored one book chapter and several research and conference papers. His main topics of interest are communication networks, code developing, and media synchronization.

FERNANDO BORONAT (M'93–SM'11) was born in Gandia, Spain. He received the M.E. and Ph.D. degrees in telecommunication engineering from the Universitat Politècnica de València (UPV), Spain, in 1994 and 2004, respectively. After working for several Spanish telecommunication companies, he moved back to UPV in 1996, where he is currently an Associate Professor with the Communications Department. He is also the Head of the Immersive Interactive Media R&D Group, UPV, Gandia Campus. He has authored two books, several book chapters, an IETF RFC, and more than 100 research papers. He is an Editor of *MediaSync: Handbook on Multimedia Synchronization* (Springer, 2018). He is involved in several IPCs of national and international journals and conferences. His main topics of interest are communication networks, multimedia systems, multimedia protocols, and media synchronization. He is a member of the ACM.

ALMANZOR SAPENA received the B.S. degree in mathematics from the University of Valencia and the Ph.D. degree from the Universitat Politècnica de València (UPV), Spain, in 2002, with a focus of topological properties in fuzzy metric spaces. He is currently an Associate Professor with the Department of Applied Mathematics, 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 also a member of the Immersive Interactive Media R&D Group, UPV, Gandia Campus.

ANNA VIDAL received the B.Sc. degree in mathematics and the Ph.D. degree from the Universitat Politècnica de València (UPV), in 1986 and 1999, respectively, with a focus on fuzzy topology research and its applications. She started to work at UPV, in 1986, where she became an Assistant Professor, in 1990. In the past, she has participated in several research projects, mainly focused on teaching and educational innovation projects, leading some of them and managing the Educational Innovation and Research Group in Scientific Matters, UPV, Gandia Campus. She is currently researching in the areas of design, modeling, and analysis of media synchronization techniques. She is a member of the Immersive Interactive Media R&D Group, UPV, Gandia Campus.