Inter-Destination Multimedia Synchronization; Schemes, Use Cases and Standardization

Mario Montagud[1], Fernando Boronat[1], Hans Stokking[2], Ray van Brandenburg[2]

[1] *Universitat Politécnica de Valencia*

Calle Paranimf, 1, 46730, Grao de Gandia

Corresponding autor: Fernando Boronat

Phone. +34962849341

Fax. +34962849309

mamontor@posgrado.upv.es, fboronat@dcom.upv.es

[2] *TNO: Netherlands Organisation for Applied Scientific Research TNO*

Brassersplein 2, Delft, the Netherlands

Phone. 31 88 86 67278

{hans.stokking;ray.vanbrandenburg}@tno.nl

Abstract. Traditionally, the media consumption model has been a passive and isolated activity. However, the advent of media streaming technologies, interactive social applications, and synchronous communications, as well as the convergence between these three developments, point to an evolution towards dynamic shared media experiences. In this new model, geographically distributed groups of consumers, independently of their location and the nature of their end-devices, can be immersed in a common virtual networked environment in which they can share multimedia services, interact and collaborate in real-time within the context of simultaneous media content consumption. In most of these multimedia services and applications, apart from the well-known intra and inter-stream synchronization techniques that are important inside the consumers' playout devices, also the synchronization of the playout processes between several distributed receivers, known as multipoint, group or Inter-Destination Multimedia Synchronization (IDMS), becomes essential. Due to the increasing popularity of social networking, this type of multimedia synchronization has gained in popularity in recent years. Although Social TV is perhaps the most prominent use case in which IDMS is useful, in this paper we present up to 19 use cases for IDMS, each one having its own synchronization requirements. Different approaches used in the (recent) past by researchers to achieve IDMS are described and compared. As further proof of the significance of IDMS nowadays, relevant organizations' (such as ETSI TISPAN and IETF AVTCORE Group) efforts on IDMS standardization (in which authors have been and are participating actively), defining architectures and protocols, are summarized.

Keywords. Multimedia Synchronization; IDMS; Multipoint Synchronization; RTP/RTCP; Standardization.

Abbreviations:

3DTI. 3D Tele-Immersion.

AVT. Audio Video Transport.

AVTCORE. Audio/Video Transport Core Maintenance.

C-to-C. Cluster-to-Cluster.

CMTS - Cable Modem Termination System.

CSCW. Computer-Supported Collaborative Workspaces.

DCS. Distributed Control Scheme.

DMP. Distributed Multimedia Presentations (DMP).

DSLAM - Digital Subscriber Line Access Multiplexer.

ETSI. European Telecommunications Standards Institute

HD. High Definition.

HTTP. Hyper-Text Transfer Protocol.

ID. Internet Draft

IDMS. Inter-Destination Multimedia Synchronization.

IETF. Internet Engineering Task Force

IMS. IP Multimedia Subsystem.

IPTV. Internet Protocol Television.

M/S. Master/Slave.

MSAS. Media Synchronization Application Server.

MU. Media Unit.

NGN. Next Generation Networks.

NTP. Network Time Protocol.

QoE. Quality of Experience.

QoS. Quality of Service.

RFC. Request For Comments.

RR. (RTCP) Receiver Report

RTP. Real-time Transport Protocol

RTCP. RTP Control Protocol.

RTSP. Real Time Streaming Protocol.

SC. Synchronization Client.

SCF. Service Control Function.

SD. Standard Definition.

SDP. Session Description Protocol.

SIP. Session Initiation Protocol.

SR. (RTCP) Sender Report.

SMS. Synchronization Maestro Scheme.

SSRC. Synchronization Source.

TAI. International Atomic Time.

TISPAN. Telecoms & Internet Converged Services & Protocols for Advanced Networking.

UE. User Equipment.

UTC. Coordinated Universal Time.

VTR. Virtual-Time Rendering.

WG. Working Group.

XR. (RTCP) Extended Report.

Introduction

Traditionally, the media consumption model has been a passive and isolated activity. However, the advent of media streaming technologies, interactive social applications, and synchronous communications, as well as the convergence between these three developments, point to an evolution towards dynamic shared media experiences. In this new paradigm, geographically distributed groups of consumers, independently of their location and the nature (fixed, nomadic or mobile) of the end-device they are using, can be immersed in a common virtual networked environment in which they can share services, interact and collaborate in real-time within the context of simultaneous media content consumption.

Nowadays, communicating (e.g. by using text, audio or video chat) while watching TV is already quite common. However, in the current situation it is mainly a parallel activity, not integrated with the primary function of watching TV. In order to integrate them further, and provide an enjoyable dynamic shared media experience, various technical challenges must be faced. Examples are universal session handling, user mobility, social interaction modeling, user preferences management, automatic media resource discovery, contextual personalization, synchronization, intelligent (device-tailored) media adaptation and delivery, QoS, QoE, scalability, coverage-based solutions, noise reduction, presence awareness, design guidelines, privacy concerns, and social networking integration ([1], [2]).

This paper mainly focuses on one of these challenges, which is the synchronization of media streams across multiple separated locations, also known as multipoint, group or Inter-Destination Multimedia Synchronization (IDMS). It is one of the major challenges ahead to enable a satisfying feeling of togetherness (defined in [3] and closely related to QoS or QoE) in some emerging synchronous media sharing applications. Several use cases in which IDMS is essential are compiled in this work, and they are qualitatively categorized according to their temporal synchronization requirements.

The structure of the paper is as follows: first the definition and various types of multimedia synchronization are given, several use cases in which IDMS is useful are introduced and new challenges to tackle the IDMS problem in current content delivery networks are presented; in Section 2 some related works are summarized; then, an exhaustive qualitative comparison among different IDMS schemes proposed by researchers up to date is presented in Section 3; Section 4 briefly outlines the current standardization efforts on IDMS; and finally, Section 5 presents some conclusions and future work.

Multimedia Synchronization

Multimedia applications usually involve the integration of various independent media streams, including both continuous (audio or video) and discrete streams (text, data, static images, …), sent (unicast or multicast) by one or more sources to one or several receivers, which can be playing one or several of those streams simultaneously. Due to the temporal, spatial or semantic relationships between the Media Units (MUs[1]), such as video frames or voice samples, within or among the involved media streams, a precise mechanism of coordination and organization in time is needed in order to ensure a time-ordered presentation of the received MUs, in the same way as the MUs were captured or generated. Such a process of maintenance and integration, in the presentation instant (or playout point), of the temporal (or spatial) relationships of the different types of media streams is referred to as multimedia synchronization [4].

Three kinds of multimedia synchronization techniques can be distinguished: intra-stream, inter-stream and inter-destination synchronization (IDMS). Fig. 1 shows an example of each of them. In it we can see a group of distributed receivers on an IP network, which are playing video and audio streams corresponding to a football penalty shot sequence. First, intra-stream synchronization deals with the maintenance, during the playout, of the temporal relationships among subsequent MUs within each media stream. In Fig. 1, we can observe a proper and continuous playout process of each media stream in all the receivers, such as the evolution of the video stream, with the associated audio stream of the sportscaster

[1] Multimedia information can be modeled as streams that are made up of a time sequence of finite MUs (also called in other works Media Data Units or MDU, Information Units or IU, and Logical Data Units, or LDU).

relating the sequence. As an example, if the multimedia source captures the video sequence at 25 MUs (video frames) per second, they must be played out (displayed) during 40 ms (each frame) at the receiver side. Inter-stream synchronization refers to the preservation of the temporal dependencies between playout processes of different, but correlated, media streams (time dependent or not, e.g. a still image with capture text) involved in the application (audio and video sequences of the penalty shot in the figure). One example of inter-stream synchronization is the synchronization between the sportscaster's audible words and the associated movement of the lips, which is referred to as lip synchronization (lip-sync [5], [6]). Another innovative example is scented audiovisual synchronization which is referred to as the maintenance of the temporal inter-media relationships between computer generated streams of smell (olfactory data) and associated audiovisual content, so as to produce an olfaction-enhanced multimedia presentation [7].

These first two kinds of synchronization techniques are usually considered and implemented in typical multimedia applications. Nevertheless, a new third type of synchronization, IDMS, is also essential in many emerging distributed multimedia applications (see IDMS use cases in Section I.2). IDMS involves the simultaneous synchronization of one or more playout processes of one or several media streams at geographically distributed receivers, to achieve fairness among them. In the IDMS context, fairness is concerned with the problem of ensuring that the playback timing of MUs at all the distributed receivers should be (almost) the same; otherwise, the earlier a receiver gets MUs, the earlier it can react to specific events. In some IDMS use cases, lagged clients may feel unfairness because some other advanced clients will have an advantage over them. It can be noticed in Fig. 1 that, at any moment during the multimedia session, all the receivers are playing the same MU of each media stream (IDMS). In the distributed video watching scenario in Fig.1, users should experience the goal event almost simultaneously to have a fair shared experience. As an example of a lack of IDMS, it will be very frustrating for one user, who is watching an on-line football match together with some remote friends, also chatting and commenting the match events, to know about a goal from a friend's chat message, before seeing it on his or her screen. In this paper, we mainly focus on this multimedia synchronization type.

Two models of media sharing can be distinguished: asynchronous content sharing among members of a social group distributed in time and space, and synchronous sharing of content among members of a social group temporally collocated, either being spatially distributed or even sharing a physical space [8]. An example of the former is a community channel application which forms an aggregation point for related content from different technology domains (e.g., in IPTV, when sharing a documentary about some topic) combined with content from a user's storage devices (e.g., photos, brochures, news…). In these kinds of applications the achievement of strict synchronization is not needed. So, in this work, we are primarily concerned in synchronous sharing media among disjoint groups of users, in which more strict synchronization is required. Next, up to 19 synchronous media sharing use cases are presented.

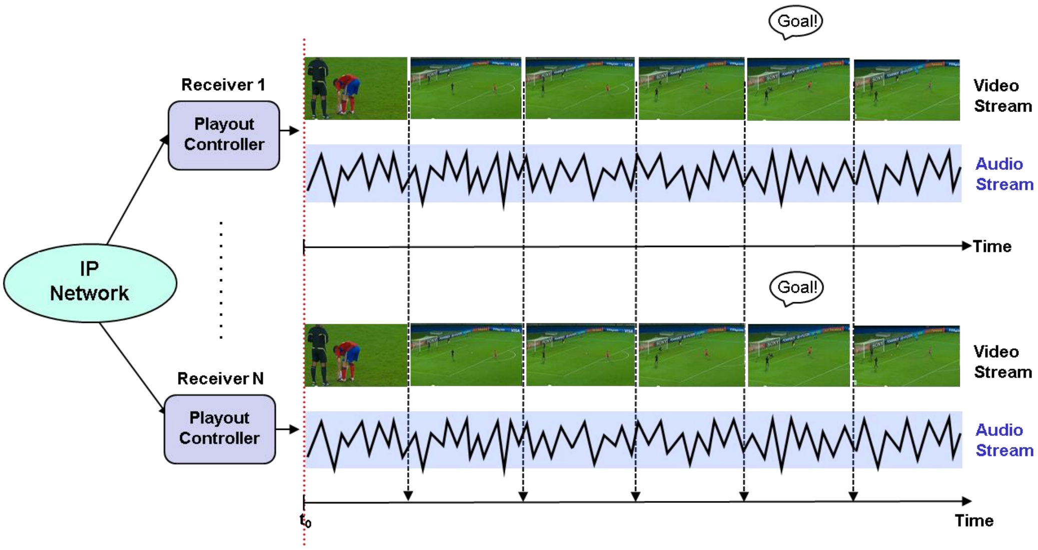


Fig.1 Multimedia Synchronization.

Examples of Applications in which IDMS is needed

IDMS can be applied to any type and/or combination of streaming media, including both live and stored content streams, such as audio, video and scene information (e.g. chat, subtitles, images, etc.). Nowadays, we can find many distributed social multimedia applications in which the lack of IDMS may affect the user experience (QoE) in many different ways [9]. Here we present a large compilation of typical use cases in which IDMS is needed to show its wide applicability:

1. *Synchronous e-learning*. In real-time synchronous distance learning applications (e.g. the one in [10]), an instructor can distribute a multimedia lesson to a group of students (that could be attending it from different locations), and he or she can occasionally make some comments or questions about its content. Hence, it is crucial that each one of the students receives the multimedia question (possibly transmitted in several streams) at the same time, and, as a result, has a fair chance of answering.
2. *Networked quiz shows* with distributed on-line participants in which the winner is the first one to answer a multimedia question correctly. In such a case, in absence of IDMS, participants may feel unfairness because the contestant at the shortest delay destination will have an advantage over the others. National laws may even prohibit this, as broadcasters are not allowed to offer games of chance without a specific license for this, and without IDMS such quiz shows may become a game of chance.
3. *Networked real-time multiplayer games* ([11]-[13], [14], etc.), where multiple media streams such as computer data (e.g., information input from a keyboard), voice and video are simultaneously involved. In such scenarios, multiple players often collaborate (as a team) with each other and fight against other multiple players (belonging to other teams). When each player presents output timing different from the other players, the fairness among them, or the efficiency of the collaborative work, can be seriously damaged.
4. *Multimedia Cluster-to-Cluster (C-to-C) applications* or *multi-point to multi-point communications* ([15]-[17]), including independent but semantically related data streams (audio, video, image, text media, …) sent from end-systems located in one or more clusters[2] (sender clusters) to end-systems located in other distributed clusters (receiver clusters). For example, the sender cluster may consist of a collection of capture devices/sources (e.g., video cameras, microphones, etc.), each one producing an independent stream of data (video, audio, graphics or text media), and the receiver cluster might be a collection of display devices (e.g., screens, speakers, etc.) and computers that store and reproduce the received data streams. Other examples of such applications are 3D Tele-Immersion (3DTI) [18], computer-supported

[2] A cluster can be considered as a collection of computing and communication end-systems sharing either the same local environment or a media experience as a logical group.

collaborative environments [19], video-centered communications (e.g., surveillance systems, traffic and street monitoring, etc.), distributed multimedia presentations (DMP) which integrate correlated media streams and possess temporal requirements with respect to the presentation, ubiquitous computing environments, and more complex multi-stream, multimedia presentation environments. For example, in a 3DTI scenario, a scene acquisition subsystem could be comprised of an array of digital cameras and computing hosts set up to capture a remote physical scene from a wide variety of camera angles. Synchronously captured image sequences would be multi-streamed to a distributed 3D reconstruction subsystem at a remote location. The resulting view-independent depth streams would be used to render a view-dependent scene on a stereoscopic display in real-time using head-tracking information from the user. Overall, the application would allow remote participants to interact within a shared 3D space so everyone would feel a strong mutual sense of presence. All these C-to-C applications pose sophisticated data transport requirements due to the use of multiple, semantically related flows of information.

5. *Distributed tele-orchestra*. IDMS can enable the simultaneous display (play out) of a music orchestra at different locations, by remotely synchronizing all the correlated audio and video streams from multiple live musicians located in various remote distributed sites. The orchestra may consist of as few as a couple or a trio ([20]) of live musicians to an entire orchestra with many musicians. As a conductor (reference), one (preferably continuous) pre-recorded media stream or a metronome stream could be used, thus providing an aural cue. That reference media stream (e.g. a piano symphony) may be originated from one network site and sent to the other sites where live performers are listening to it and playing their corresponding instrument melodies in a temporally synchronized way, which will be transmitted in new individual media streams. Additionally, if needed, the metronome stream could also be forwarded as a new media stream by one of the remote sites. Note that neither the performers nor the conductor could hear the compound symphony entirely. Each performer could only hear the conductor part of the orchestra (a somewhat contrived musical experience for the performers). The correlated media streams must be delivered synchronously to the audience in

order to produce a high quality music performance in spite of delay variations and network fluctuations through the networks that carry the audio and video flows. Moreover, those media streams must be played out simultaneously at all the distributed listeners' locations. This scenario imposes very stringent synchronization requirements to achieve a high quality music orchestra, compounded by the individual melodies from distributed live musicians.

As a similar use case, in [21], authors studied the effect of group synchronization (or IDMS) control in a networked chorus. In this scenario, there was a conductor providing a standard timing, several distributed singers singing according to the standard timing and actions of the conductor, and a group of distributed listeners as an audience. Here, synchronization in a networked chorus means that singing voices and action of the conductor need to be coherently presented in each one of the singers' and listeners' terminals, respectively. The assessments results in this work proved that group synchronization can significantly improve the overall user experience (QoE) in a networked chorus.

As well, the work in [22] revolves around a socially augmented rock concert in which four friends share the music experience and enrich it through social interaction and media sharing. Some of the friends are watching a live broadcast of the concert (high-quality professional TV content), each from their own home. They could talk to each other using the IP-based communications facilities built into their TV sets (Internet) and at the same time receive a live video feed from some other friends actually attending the concert. The friends at the concert would use their smart phones to generate the stream, which could be rendered as a picture-in-picture overlay on the TVs of remote friends, giving the remote friends a view of the concert from the local audience's point of view. Also, the friends can interact with each other and comment on the shared music experience via chat or audio/video conferencing.

6. *Multi-party multimedia conferencing.* In these applications, if the output timing of speech (or video) by a participant largely varies from destination to destination, the conference itself cannot be held. Furthermore, the bigger the size of the multicast group, the more significant delay or playout differences become.

7. *Presence based games*. In such scenarios, users can win a prize when they watch a certain advertisement at a certain time. When the content is too much out of synchronization it can no longer be determined what specific content the user has been watching.
8. *Consumer-originated content and content sharing on a multimedia conference*, whose purpose is sharing some content in real-time with family, friends, colleagues or other types of "*buddies*" all over the world. An example is when browsing together through recorded digital photos and videos and commenting on the content in real-time.
9. *Conferencing sound reinforcement systems*, often used in commercial and government installations such as legislative chambers, courtrooms, boardrooms, classrooms (specially, those supporting distance learning), etc. Each participant who is using one of these systems has a microphone and a speaker. There may also be other speakers to provide reinforcement for non-speaking participants such as in an audience area or jury box. Each microphone/speaker pair is individually connected to a network and transmits digital audio over the network to the other devices through the network and receives digital audio to be reproduced through the speakers. There may be a central appliance which receives, prioritizes and mixes the microphone signals. In some systems an individual mix is created for each speaker so the speaker's own voice does not come out from his/her loudspeaker or from those immediately surrounding him/her. The objective of these systems is not that the person speaking sounds or feels amplified so much as it is to provide enough gain to enhance intelligibility. Reaching this objective helps ensure that natural person-to-person communication is retained. To this end, it is desirable that the sound through the system and from the speakers arrive 5 to 30 milliseconds after the sound arriving through the air from the person speaking. Delays in this range invoke the Haas effect which allows listeners to locate the person speaking based on the sound arriving through the air while the sound reinforcement system provides the additional gain required to achieve the desired intelligibility. It is also desirable for the sound to come out of nearby speakers at within 5 milliseconds as longer differential delays will be perceived as reverberation or echo.

10. *Networked stereo loudspeakers* in which two or more speakers are connected to the network individually. Human beings can localize sound based on inter-aural time differences, in a stereo listening situation. So, we are very sensitive to changes in latency between the (two) speakers. We perceive these changes as a shift in or instability of the "*sound stage*" during critical listening. Shifts around 10 microseconds (or even smaller) could be noticeable. If the individual speakers operate from independent network interfaces in a stereo listening setup, any changing difference in latency between the (two) speakers greater than few microseconds will affect the listening experience negatively.
11. *Phased array transducers* used in audio applications. This technique works by sending or receiving slightly different versions of a signal in a spatial sampling arrangement to produce or record spatial and directional sound fields. One example application is the conferencing microphone system that is able to electronically aim at the person speaking to improve signal to noise ratio. These microphones are also able to report the location of the speaker for purposes of automatically aiming a video camera at them. The individual transducers in such applications can be extremely sensitive to differential latency. Another example is a concert sound system called *"line arrays"* which allows technicians the control over the amount of sound sent to different places. People in the front of the audience can have the same loudness as those in the back. By preventing sound from reaching the roof and back wall of the performance space, the amount of reflected sound heard by the audience is reduced and the listening experience is improved. In these systems, accuracy in locating or emitting sound is related to differential latency through basic trigonometry. Microseconds of differential latency can translate to degrees of uncertainty. Accuracy greater than the audio sample period (about 20 microseconds for professional 48 kHz sample rate) is generally desired.
12. *Seamless switching among media devices*, e.g., where a user changes his or her TV session from a fixed television set to a mobile device or vice versa. If there is too much delay difference between content reaching the different terminals, this will spoil the switching experience as a significant portion of the content may be missed or played out twice.

13. *On-line election events*. As an example, in a pop star competition show, any vote from viewers (fans) at home sent during the show must be valid, and all the votes sent after the deadline (lines are closed) must be rejected.
14. *Game-show participation*. Starting from simple messaging to a TV show or dialing in by phone, users will become live participants in TV shows with live streaming footage through user webcams and real-time interaction between the participants and the TV show.
15. *Social TV*. This enables different groups of viewers, independently of their location and the network (and the device) they are using, to watch a TV program, while simultaneously interacting and sharing services, by using immediate chat messaging, audio/video conferencing services, or for that matter any other sort of shared experience that is yet to appear. In [2] and [8], some streaming media (IPTV or WebTV) applications providing synchronous shared experiences are presented. As an example, Watchitoo[3] is an emerging web-based application that enables not only chatting, but also audio and video conferencing while watching the same video content. What started as Internet TV has evolved into a richer mix of media for Social TV, allowing direct social interaction among people, supported by two-way communications. Social TV combining TV content with direct social and community interaction (e.g. using Facebook, MySpace…) is taking root in connected set-top boxes, web-ready TVs, and PCs. The traditional ubiquitous model (two children and mom-and-dad scenario), obsolete and overused, is being replaced by a much more dynamic family unit that is spread around the world with people moving and interacting digitally. TV is part of the shared family experience and will continue as a part of its heritage. As people are social by nature, this new TV model promises to deliver a world of content and services to any combination of devices (set-top boxes, web-ready TVs, and PCs), anywhere and anytime (the future of IPTV is connected, mobile, personal and social [23]).

    Another example is when various friends are watching a live on-line football match at separate locations (*"watching apart together"*), as reflected in Fig. 2. We could also think about the possibility of adding more friends to the session, for example, those who are travelling by train, viewing the match using smart phones (Fig. 2) and, in an extreme case, some other friends could

[3] http://watchitoo.com/

be watching the match live physically at the stadium and communicating with the others using their phones (audio/video calls or text messages). In such a case, inter-stream synchronization must be performed between the involved time-dependent media streams, such as between the multimedia content that the users are watching together (e.g. the video stream corresponding to a football match) and the associated streams corresponding to the chat messaging or audio/video conferencing services. Moreover, a significant event, e.g. a goal (see Fig. 1), should be viewed or experienced by all the home (or remote) users almost simultaneously, even in all the associated chat messaging and conferencing media streams, in order to not degrade the user experience on such interaction (IDMS). Instead, as stated before, it would be very frustrating for a home user to experience a goal later than the friends at their homes (or train) while they are chatting.

Thus, we can distinguish the different media streams involved in such interactive scenarios as primary media streams and shared experience media streams [2]. The former refer to the multimedia content the users are playing out (watching, listening, reading) together, and that must be rendered at various locations in a time synchronized manner. The latter refer to those streams of communication among the distributed users that enable the shared experience and the interaction among themselves. Both types of media streams must be globally synchronized according to their relationships.

To provide this kind of service, some platform (e.g. the one presented in [22]) involving all the friends attending the event (e.g. football match), either physically (at the stadium) or remotely (at home/train) will be needed for creating a dynamic community (also known as an ad-hoc group) in a cross-domain session through which media and interactions can be shared, synchronized, adapted, recorded, played back, and analyzed (with the consent of the users). This session would exist for the duration of the match and any related activities, such as post-match advertising. Once the group has been created, all the friends should be informed in an appropriate way, based on their context. Those using computers would receive on-screen overlay notifications, while those at the stadium would receive mobile alerts. Once the match begins, the friends could talk to each other and discuss about the match, including watching each other (videoconferencing). Friends at the stadium could send video of the match to give friends at home a view of the match from the spectators' point of view. Friends at home could also send the recorded TV edited highlights (e.g. to clarify off-side situations).

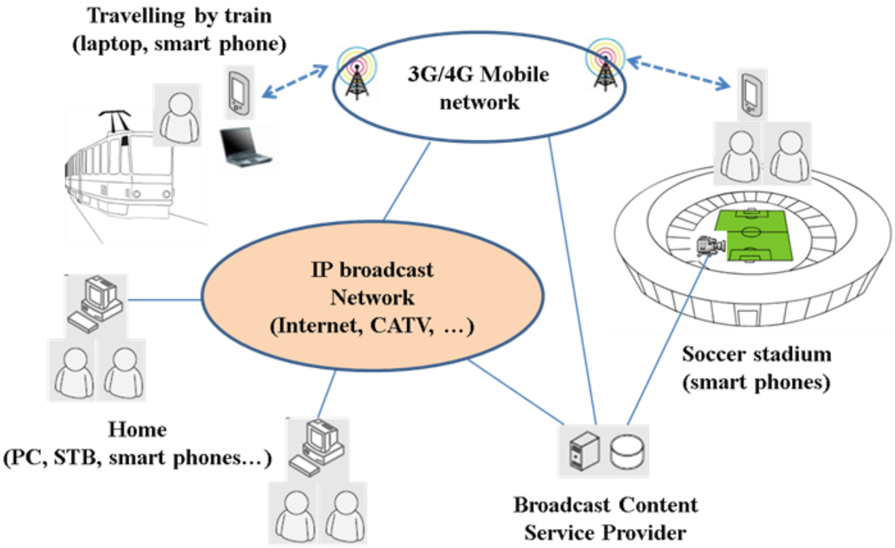

Travelling by train
(laptop, smart phone)
3G/4G Mobile network
IP broadcast Network (Internet, CATV, ...)
Home
(PC, STB, smart phones...)
Soccer stadium
(smart phones)
Broadcast Content Service Provider

Fig. 2 A generic Social TV use case.

16. *Shared service control*. This use case is similar to Social TV, and allows distributed users to experience some content-on-demand together, while sharing the trick-play controls (play, pause, fast forward, rewind). Differences in playout speed and the effect of different transit delays of MUs and of trick-play control signals would desynchronize content playout.
17. *Second screen synchronization*. Community gaming around TV content on a second screen poses different synchronization requirements. In such cases, smaller synchronization bounds might be needed compared to Social TV and soccer watching. This has many applications such as, for example, rating systems for talent shows and live interactive quiz shows. An architecture and a working implementation for using secondary screens in the interactive television environment is presented in [24].
18. *Networked video wall*. A video wall consists of multiple computer monitors, video projectors, or television sets tiled together contiguously or overlapped in order to form one large screen. Each screen only shows a part of the larger picture. In some implementations, each screen may be individually connected to the network and receive its portion of the overall image from a network-connected video server or video scaler. Screens are refreshed at 60 hertz

(every 16-2/3 milliseconds) or potentially faster, but if the refresh is not synchronized, the effect of multiple screens acting as one will be broken.

19. *Synchronous groupware*. This is a technology that facilitates teamwork, supporting the communication and coordination between geographically dispersed team members [25]. It encompasses a wide range of applications like collaborative whiteboards, text editors or Web browsers. These applications need to share a consistent common state to enable an efficient integrated collaboration.

Challenges

To the best of our knowledge, the exact ranges of asynchrony levels which could be tolerated by users for the above use case applications (i.e. the asynchrony limits that, if exceeded, are noticeable and, as a result, are annoying to users) have not been sufficiently determined yet. They should be obtained through very rigorous objective and subjective assessments (user perception tests), possibly including longer-term testing in live systems, in contrast to testing in artificial test environments. Here, we present some conclusions extracted from previous works in which some preliminary assessment results for Social TV-like scenarios have been presented, but we consider they still have to be followed up with more complete and exhaustive testing in the future. The presented ranges of tolerated asynchrony levels obtained in such Social TV-like scenarios are vastly different to some of the other use cases mentioned in the previous section (e.g. networked loud speakers, phased array transducers, etc.).

Traditionally, 150 ms has been used as a rule of thumb, a value drawn from telecommunications research. This rule states that the maximum end-to-end one-way delay when talking remotely should not exceed 150 ms. Below this value a user cannot perceive the delay in communication, and therefore cannot detect differences on synchronization of shared video content [2]. The study in [26] provides a set of allowable asynchrony values between different types of media streams that may be tolerable to human perception, but only referred to inter-stream synchronization. Additionally, some Social TV related studies exist, such as the ones in [9] and [3]. In [9], it is concluded that the requirements on inter-destination content synchronicity in interactive services may vary between 15 and 500 ms, depending on the type of service. In some cases, differences around 100 ms may already have an annoying effect on such interaction. More recently, the study in [3] aims to determine acceptable synchronization levels (i.e. asynchrony limits that are noticeable and annoying to users) for Social TV scenarios (watching on-line together a synchronized version of a video while communicating with each other). It is concluded that asynchronies (playout time differences) up to 1 second might not be perceptible by users in a distributed video watching scenario while communicating using voice conferencing services, but playout differences above 2 seconds really become annoying for most users. Concretely, voice chatters and active text chatters felt more together and noticed de-synchronization (over 1 second for voice, and over 2 seconds for active chat). However, these results are largely dependent on several factors, such as the genre of the video content, the number of users, their activity and profiles (age, sex, relationships among them –family, friends, partners, etc.– …), the communication channel, etc. Consequently, no statistically absolute user tolerance limits may be derived from these preliminary experiments, and more accurate asynchrony levels for IDMS should be achieved to avoid the user’s frustration, and thus guarantee an enjoyable shared experience in such synchronous media sharing applications.

In fact, these differences can be much larger in current content distribution networks and newer delivery paradigms ([9], [27], [28] e.g. IMS-based TV broadcast channels), mainly due to the existence of several undesirable, unpredictable, and/or uncontrollable factors in the multimedia end-to-end distribution chain (some of which can be either related to the distribution network or to the device or end-system features), such as variable capturing, coding, encryption, packetization, network (traffic load, trans-coding or format conversion, fragmentation and re-assembly of packets, multicast or dynamic routing strategies, improper queuing policies at the intermediate routers, etc.), processing, depacketization, decoding, decryption, buffering, rendering and presentation delays, or packet losses, which can seriously disturb the original media timing at the receiver side, and result in different (and time-variant) end-to-end (or playout) delays when multicasting one or several flows of information from one or more media sources to one or multiple destinations (that can be using different kinds of terminals), possibly over different delivery chains (network architectures/technologies/connections, cross-domain scenarios, coding mechanisms, etc.), as shown in Fig.3.

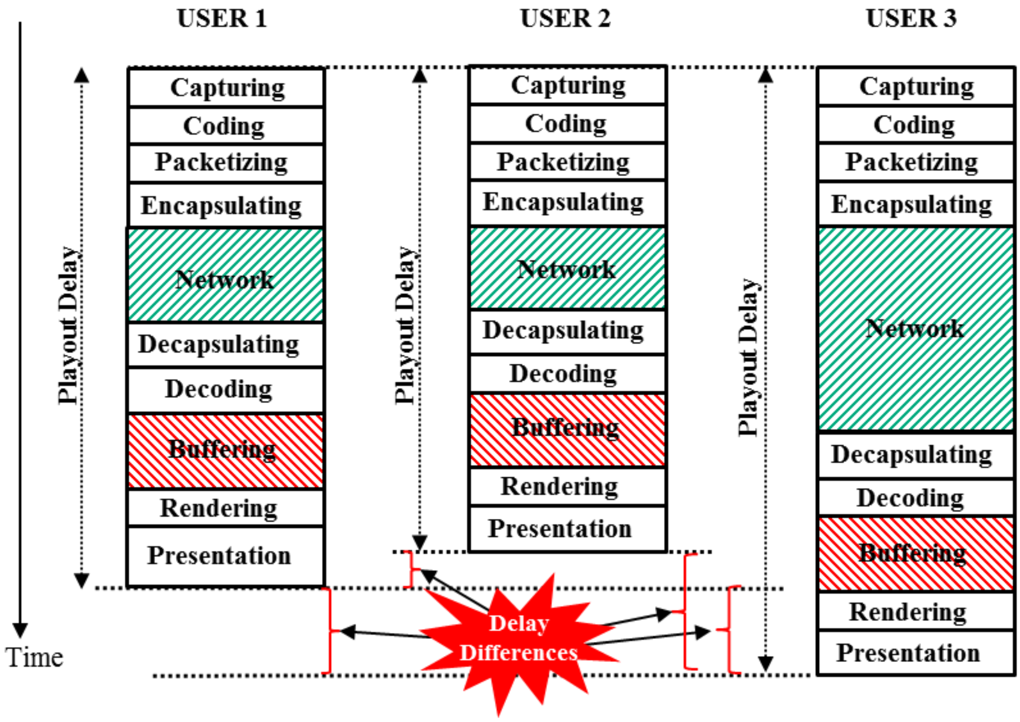


Fig.3 End-to-End (or Playout) Delay Variability: Need for IDMS.

Some of the above factors that can disturb the original timing of the incoming media streams, and can be tackled either individually or in an integrated manner, are the following:

- *Network Delay*: The MUs sent by the source experience different network delays to reach each one of the destinations. As well, network delays can vary according to the network load.
- *Network Jitter*: It denotes the varying delay that stream packets experience on their way from the sender to the receiver devices. It is mainly introduced by buffering in intermediate nodes. It refers to the delay variation of inter-arrival times of packets at the receiver because of varying network load. Jitter is commonly equalized by the use of an elastic reception buffer at the receiver side.
- *End-System Jitter*: Delay variations in presentation at the receiver because of varying (workstation) CPU load and protocol processing delays. It

refers to the variable delays arising within end-systems, and it is caused by varying system load and the packetizing and depacketizing of MUs with variable size, which are passed through the different protocol layers.

- *Clock Skew*: The clock time differences between senders and the receivers.
- *Clock Drift*: The rate of change of clock skew because of temperature difference or imperfections in crystal clocks.
- *Rate Drift*: Change in generation and presentation rates because of server and receiver load variations.
- *Network Skew*: Time difference in arrival of temporally related packets of streams, which is a differential delay among the streams.
- *Presentation Skew*: Time interval in which the temporally related packets of the streams are presented.
- *Encoding used*: If various media streams are encoded differently, the decoding times at receiver may vary considerably, specially, when using MPEG or H.264 interpolation with different Group of Pictures (GOP) sizes.

Another additional factor to take into account when using digital TVs is the display lag (i.e. the time difference between the instant at which a signal is input into a display and the instant at which it is shown by the visualization device), which may be caused by image processing routines such as scaling and enhancement. Thus, it can spoil the user experience (QoE) in gaming or Social TV scenarios. Moreover, display lag may cause a noticeable offset between the audio and the image signals. Such effect has been recently studied, and it was reported that HDTV lags can vary between 30 and 90 ms depending on the television type and of the input signal used [28].

Although presentation times are carried in media packets, buffering requirements usually do not match (different end-points may also have different de-jitter buffer sizes, which will complicate things even further) and distribution links may present different delays, so playout time discrepancies will occur. Even if a service provider tries to reduce this problem for its customers, the neighbors could access through the network infrastructure of another provider and such a delay difference is complicated to manage, unless the providers coordinate their media distribution.

Accordingly, it is concluded in [9] and [28] that existing distribution technologies do not handle the IDMS problem in an optimal way. Delay is not a serious constraint in cases where isolated users are consuming non-time-sensitive content from broadcast, content-on-demand or network-based personal video recording. Nevertheless, delay, and its variability, becomes a serious problem when an interaction between the user and the media content, or interaction between different users in the context of specific content consumption is needed, because it could be detrimental to the QoE in those synchronous social media applications and may prevent the inclusion of advanced forms of interactivity in such group shared services. Thus, additional adaptive techniques must be provided to meet the above synchronization requirements (especially IDMS) in practical content delivery networks.

As a summary, Table 1 gives a preliminary categorization of the above presented use cases assigned to different required synchronization levels and the technical requirements in order of magnitude of the maximum tolerable delay differences (asynchrony) between destinations or output devices. As there are many C-to-C applications, this general use case is not included because the requirements depend on the type of the application. The technical requirements are not meant to be exact, but give an order of magnitude of the maximum tolerable delay differences between the various destinations or output devices. These approximations, expressed with intervals and not with exact values, are derived from the functional reason for synchronization:

- *Very high synchronization* (asynchronies lower than 10 ms) is necessary for different audio outputs in a single physical location. For example, this is necessary for proper sound localization, as explained in [29]. That work explains about audio localization and the granularity of the human ear, which can recognize differences of 10 micro-seconds or less between the arrival times of sound at each ear.

- *High synchronization* (asynchronies between 10 ms and 100 ms) is required for any use case in which fairness is important. Typical response times of users should not be influenced too much by delay differences of media playout to which users respond. As explained in [30], 100 ms is a well-known upper limit for users to feel that a system is reacting instantaneously. Also, in [14] we found citations to several studies defining a delivering delay threshold around 150-200 ms to keep an enjoyable shared experience in networked multiplayer games. In such cases, when synchronization mechanisms are adopted to guarantee a consistent global view of the state of the game, the degree of interactivity may be jeopardized. Thus, sophisticated techniques need to be devised to preserve both consistency and interactivity within these bounds.
- *Medium synchronization* (asynchronies between 100 ms and 500 ms) is required in cases in which various related media items are displayed somewhat simultaneous, but in which no real-time requirements, such as e.g. lip-sync, are posed. Typical use cases here are about semi real-time additional content, or about users who are consuming content at different physical locations and do have active interaction, but not so strict as in the high accuracy scenario. For such interactive sessions, the delay should be kept in limits where it does not impact (conversational) dynamics too much, typically within the order of several hundred milliseconds, as explained in [31]. Also, the work in [3] showed that in a Social TV use case active participants start to readily notice delay differences above 500 ms.
- *Low synchronization* (asynchronies between 500 ms and 2000 ms) is required in cases where media is consumed by different users at different physical locations, but the interaction level between users is not of a very competitive nature. User tests in [3] showed that asynchronies (playout time differences) up to 1 second might not be perceptible by users in a distributed video watching scenario while communicating using voice conferencing services, but playout differences above 2 seconds really become annoying for most users. Concretely, voice chatters and active text chatters felt more together and noticed de-synchronization (over 1 second for voice, and over 2 seconds for active chat). This is why we choose the 2 second delay difference as an upper bound in the low synchronization range.

Table 1 Use cases according to their synchronization requirements

Synchronization Level	**Technical Requirement**	**Relevant use cases**
Very high	10 us – 10 ms	- Networked stereo loudspeakers - Phased array transducers - Video wall
High	10 – 100 ms	- Distributed tele-orchestra - Networked quiz shows - Networked real-time multiplayer games - Multiparty multimedia conferencing - Conferencing sound reinforcement system - Game-show participation
Medium	100 – 500 ms	- Synchronous e-learning - Synchronous Groupware - Presence based games - Consumer-originated content - On-line election events - Second screen sync
Low	500 – 2000 ms	- Seamless switching among media devices - Shared service control - Social TV

II. Related Work.

Over the last years many solutions for both intra-stream and inter-stream synchronization have been designed (e.g. [32], [33]), but not so many for IDMS, despite the increasing relevance that this kind of synchronization is acquiring in a variety of emerging distributed multimedia applications. On the one hand, [34] provides a comparative survey of many intra-stream synchronization techniques. On the other hand, in [4], the currently most exhaustive qualitative comparison between the most recent inter-stream and IDMS proposals is presented. While most of the previous work on multimedia synchronization has focused on intra-stream and inter-stream synchronization techniques, this section solely focuses on IDMS solutions for assuring concurrently synchronized playout points at different locations. Generally, three schemes are employed to perform the IDMS control (Fig.4): two centralized schemes (Master/Slave or M/S Scheme and Synchronization Maestro Scheme or SMS) and one distributed scheme (Distributed Control Scheme or DCS).

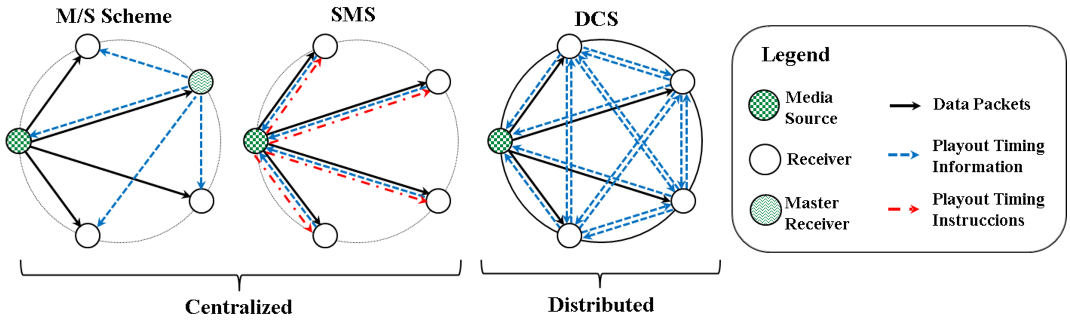


Fig. 4 IDMS Control Schemes.

Regarding centralized schemes, on the one hand, in M/S Scheme (proposed for the first time in [35], and used later in [2] and [3]), receivers are differentiated into master (one) and slave receivers (the rest). The master receiver multicasts feedback control messages about playout timing to all the other (slave) receivers. Accordingly, each slave receiver adjusts its own playout process (the output timing of MUs) to the reference playout process of the master. On the other hand, SMS (proposed for the first time in [36]) is based on the existence of a synchronization maestro or manager (that can be the source, one real or fictitious receiver or a completely separate entity), which gathers the playout information of all the active receivers and corrects their playout timing by distributing new adapted control messages. In order to do this, each receiver sends (unicast) their playout timing information to the maestro, and then the maestro, after processing such information, multicasts a new control packet including a reference playout point to which the receivers should be synchronized (in order to adjust the output timing among the destinations). Most solutions do require wall clock synchronization between the various receivers, to achieve IDMS.

SMS is performed in a similar way as the M/S Scheme. However, it should be noted that in M/S Scheme no slave destinations send any timing information control packets including their local playout timing. Moreover, in SMS, the receivers can also be classified into an M/S Scheme regarding the reference output timing, in which the playout timing of the master receiver is taken as the synchronization reference for adjusting those of all the other (slave) receivers. Besides, the master receiver role could also be dynamically exchanged between receivers according to the network conditions, allowing M/S switching technique [4].

In [37], authors presented a preliminary version of an RTCP-based IDMS approach, following an SMS, in which the source was also the maestro, and it selected a receiver as the (master) synchronization reference for adjusting the output timing of all the other (slave) receivers. Then, in [17], this IDMS proposal was extended so that the maestro could separately synchronize the playout processes of independent logical groups of distributed receivers (clusters). Moreover, several dynamic strategies for choosing a reference playout point for IDMS in each cluster were adopted: *i)* synchronization to the slowest receiver (i.e. the playout point of the most lagged receiver was selected as the IDMS reference); *ii)* synchronization to the fastest receiver (i.e. the playout point of the most advanced receiver was selected as the IDMS reference); *iii)* synchronization to the mean playout point (i.e. the IDMS reference was calculated by averaging the playout timing reported from all the distributed receivers); and *iv)* synchronization to the server nominal rate (i.e. the source acted as a virtual receiver with an ideal playout timing to which all the receivers must synchronize). In that work, the effectiveness and suitability of those policies for specific network conditions and application requirements were examined, according to the impact on the overall quality of the playout adjustments and the buffer fullness variations as the multimedia session goes on.

In DCS ([38], [11]-[13], [22] and [2]), all the receivers multicast feedback information about their playout timing to all the other receivers and each one of them selects the synchronization reference from among its own playout timing and those of the other receivers, e.g. following one of the first three master reference selection policies presented above. The fourth strategy can only be applied in SMS, and only if the maestro functionality is integrated within the media source resources. In [38], an IDMS approach using DCS is introduced for the first time, which adaptively keeps the temporal and causal relationships according to the network load. In [11], a bucket mechanism (in which users' events are delayed for a sufficiently large duration to prevent inconsistencies before being executed) is used as a DCS technique to be applied in interactive networked games. In [12], the use of *"local lag"* and *"time warp"* algorithms is proposed to avoid inconsistencies between users in replicated continuous applications, such as networked games. First, *local lag* algorithm is used to compensate for short term inconsistencies (an extra delay to sporadic events' execution is introduced, to ensure those events are received by all the peers. Second, *time warp* algorithms aim to undo inconsistencies that may still occur due to various uncontrollable factors. Time warp is the process of rolling back changes to the last known consistent state, in case inconsistencies are detected (e.g. it can be used for sending update playout actions, like jumping back or forward in distributed video watching scenarios). This solution has recently been adopted in the iNEM4U[4] platform ([22]), which provides open, intelligent, and interoperable support services for social applications. In [2], such algorithms have been adapted to achieve coherent execution of specific users' actions at all the clients, so that a consistent version of a shared video watching experience is perceived by all the users (e.g. if the primary media stream is paused at one end, then, the pause should also be executed at all other clients within bounded tolerance limits).

The work in [13] presents another DCS-based approach which takes into consideration different conversation roles in a networked game (rock, paper, and scissors) using a video conferencing system. Thus, the playout adjustments depend on the role of each player (caller or receiver), similarly as in an M/S Scheme. In [39], the importance of IDMS in web-based P2P TV systems for minimizing noticeable playout differences was revealed. Also, the study in [40] claims that IDMS improves the shared TV watching experience.

In [41] and [42], an IDMS approach, using DCS and SMS, respectively, was presented by taking into account the importance of the media objects, for its application in networked virtual environments. In those works, the concepts of *"global importance"* (importance which is judged from the point of view of all the users) and *"local importance"* (importance which is judged from the viewpoint of each user) were introduced. Both works were based on the use of the Virtual-Time Rendering (VTR) Algorithm (one of the most popular intra and inter-stream synchronization algorithms), which is applicable to networks with unknown delay bounds, makes use of globally synchronized clocks, and consists

[4] www.iNem4u.eu, Interactive Networked Experiences in Multimedia for You, FP7-ICT-2007-1-216647

of the dynamic adjustment of the MUs rendering-time, according to the network condition.

In [43], the influence of handover on several application-level QoS metrics, including the IDMS quality, by employing VTR with SMS, was examined in an integrated wired and wireless network. In [44] and [45], the previous SMS-based approach was enhanced to be efficiently used in a P2P (Peer-To-Peer) system and in a networked collaborative real-time game, respectively.

In [46], the three IDMS control schemes, also based on the VTR algorithm, were compared and evaluated in a relatively simple Multicast Mobile Ad-Hoc Network.

In all the above techniques, an end-user device receiving a media stream reports on arrival time or presentation time of media packets of that stream, and (one or several) synchronization entities are used to collect those control reports and to compute temporal discrepancies among the clients. As a result, end clients must perform playout adjustments to acquire IDMS.

Unlike the above solutions, which are end-point or terminal based, hybrid network-based approaches can also be employed, as the one proposed in [27], in which the synchronization functionality is implemented in network edge nodes (e.g. a DSLAM - Digital Subscriber Line Access Multiplexer- or CMTS - Cable Modem Termination System-, or even higher up in the network hierarchy), each managing the output timing of the equipment of its domain users. The synchronization point (that consists of a synchronization buffer and control functionality) in the network is selected so that further downstream delays are considered acceptable for the combinational service. Further, at a higher level, a synchronization manager is used to control the output timing of the edge nodes. This network-based approach is suitable if a very large number of nodes belong to the same session, as in massive multi-player on-line games or broadcast IPTV channels

III. IDMS Control Schemes Comparison.

Each one of the IDMS control schemes presented in previous Section has their own advantages and disadvantages. This Section presents an exhaustive qualitative comparison between centralized (M/S and SMS schemes) and distributed (DCS) synchronization control schemes considering some key factors such as robustness, scalability, traffic overhead, flexibility, location of control nodes, interactivity, consistency, causality, fairness, coherence and security. This comparison is based, partially, on the conclusions of several previous works (such as the ones in [2], [6], [11], [27], [41], [42] and [46]) and on our previous experience on IDMS ([4], [17], [37] and [47]).

1) *Robustness*. This refers to the ability to perform the IDMS control despite disconnections and failures of some participants. Generally, centralized schemes are less robust than distributed schemes and this is also the case here. In the former schemes, if the maestro (in SMS) or the master node (in M/S Scheme) cannot communicate with the other terminals owing to some trouble, no destination is able to carry out the IDMS control. Nonetheless, in a distributed architecture (DCS), the failure of any of the participants has a minor effect on the other participants because each one of them is independent and has locally all the necessary information to compute the overall synchronization status at any time. Hence, a server-less architecture can greatly simplify the deployment and maintenance of a distributed application (e.g. a network game).
2) *Scalability*. This refers to the ability to handle multiple concurrent participants in an IDMS session. SMS may present higher scalability constraints because it requires the maintenance of a dedicated server to which all the control information converges. If the control packets are generated at a non-adaptive rate (e.g. after the output of specific MUs), multiple destinations may send control packets almost simultaneously, thus originating a feedback-implosion problem because of the IDMS control. Consequently, as the number of participants increases, bursty traffic due to control packets can overwhelm the synchronization manager (in DCS, the synchronization functionality is implemented in all the destinations) and may degrade the output quality of the media streams (because some control and data packets may be lost). Even

though this failure mode also applies to distributed architectures (DCS), here the computational resources become saturated later at a larger group size compared to using a single centralized server. As discussed, in both SMS and DCS, the participants can be divided into independent logical sub-groups which can be separately synchronized, thus improving the above scalability constraints. In SMS, however, the maestro must process the playout control information of all the sub-groups in the session (although it may also facilitate the IDMS management, e.g. comparison of the playout processes only within each sub-group), but this technique is particularly beneficial in DCS because each distributed receiver must only process the feedback messages of those receivers belonging to the same group with whom it is sharing a media experience.

3) *Traffic Overhead.* This factor is closely related to the previous one. Regarding traffic overhead, two issues can be differentiated. The first one is the distribution of the playout timing messages from the participants to the synchronization managers (each participant in DCS). In M/S Scheme, only the master destination sends (in a multicast way) control messages for IDMS to all the slave destinations. Therefore, the network load will not be significantly increased when including IDMS control. In DCS or SMS schemes these control messages are sent in a multicast or unicast way, respectively. So, the traffic overhead may be higher in DCS than in SMS. The second issue is related to the transmission of playout setting instructions. Unlike in DCS and M/S Scheme, in which distributed receivers can directly adjust their playout timing according to the incoming control messages from other the receivers, in SMS the maestro, if it detects an asynchrony situation, must send a new control message to them, including playout setting instructions, which would slightly increase the network load a bit more. Generally, even considering this, the traffic overhead may be higher in DCS than in SMS, and higher in SMS than in M/S Scheme.

4) *Interactivity*. The lowest delays may be achieved using M/S Scheme because each slave destination can compute the detected playout asynchrony every time it receives the control messages from the master destination. Delays in DCS are a bit larger because in that case each participant must gather the overall playout status from all the other active participants (they can be

sent/received at different instants). Then, the highest delays occur when using SMS because, depending on the network topology and on the routing tree structure, the network delay may be increased up to twice (the maestro must gather the playout timing of all the receivers and, then, send back to them new control messages including IDMS setting instructions). So, desynchronized situations (over a threshold) will be detected and corrected earlier using M/S than using DCS, and earlier using DCS than using SMS.

As discussed, the report interval for the control messages should be dynamically adjusted (scaled up) if the number of distributed participants significantly increases. However, the lower report interval for the control messages, the sooner the playout timing information from the distributed participants will be available. It would obviously affect the interactivity and the frequency at which IDMS control can be performed. Consequently, the most (less) affected scheme would be DCS (M/S Scheme) because in such a case the amount of exchanged control traffic is the highest (lowest) between the considered IDMS schemes.

5) *Location of control nodes*. Centralized control schemes are more sensitive to the location of the multimedia source and of the synchronization manager [42]. Under heavily loaded network conditions, the IDMS performance (i.e. the level of synchronicity among receivers) with SMS can be slightly larger than the one with M/S and DCS schemes if the media source is selected as the maestro. This is due to the fact that IDMS control packets sent by the maestro are (or could be) sent through the same path as the MUs, e.g. video frames, encapsulated in data packets. Thus, although IDMS control messages hardly increase the network load, it could cause that some (data or control) packets may be dropped when the bandwidth availability is scarce, and, if a control packet is lost, the destination cannot get the reference output timing until receiving the next control packet. Conversely, in M/S Scheme, if the most heavily loaded destination is selected as the master, the data packets are less likely dropped on the intermediate links, as it does not need to receive control packets and their own sent control packets may be transmitted in the opposite direction to the media data packets. Therefore, in congestion situations, M/S Scheme may achieve higher IDMS quality than SMS (and also than DCS).

However, the most heavily loaded destination cannot always be known and, therefore, the master destination could not be selected accordingly [42].

6) *Consistency*. In media sharing applications, consistency is required to guarantee concurrently synchronized playout states in all the distributed participants. In centralized schemes, inconsistency between receivers' states occurs less likely, since all of them always receive the same control information about IDMS timing from the maestro (in SMS) or the master receiver (in M/S scheme). Moreover, in order to facilitate the design and implementation of this architecture, the maestro (SMS) functionality could be integrated within the multimedia source resources. SMS is usually used in distributed games to maintain a worldwide view of the game, as a single server simplifies problems related to causality and replication consistency [2]. In contrast, in a distributed scheme (DCS) there is no guarantee that the same reference IDMS timing, from among all the collected IDMS control reports, will be selected in all the distributed receivers, since each one takes its own decisions locally, leading to a more probable potential inter-receivers inconsistency. Also, if reports are sent using a non-reliable transport protocol, such as UDP, some receivers may and some other receivers may not receive certain receiver reports. This may lead to even more potential inconsistency in DCS.

7) *Coherence*. This concept refers to the ability to synchronously (and simultaneously) coordinate the media playout timing according to a reference timing for IDMS. Unlike in DCS and SMS, in which the maximum playout asynchrony (between the most lagged and the most advanced receiver) can be estimated, in M/S Scheme each receiver can only know the asynchrony between its local playout process and that of the master. Using M/S scheme, the reactive synchronization actions will not be performed simultaneously because slave receivers adjust their playout timing every time they detect an asynchrony value (regarding the playout state of the master) exceeding an allowable threshold and this situation may not be detected at the same time in all the slave receivers. As a result, despite the fact that M/S and SMS control schemes are the most appropriate in terms of consistency, SMS outperforms the other schemes (M/S and DCS) in terms of coherence. So, we can conclude that SMS is the best ranked scheme for IDMS regarding such factors.

8) *Causality*. Causality in media synchronization refers to the knowledge of the correct chronological order of actions. Therefore, the causality control is required by interactive media sharing applications to preserve the correct temporal ordering of specific events in the distributed media environment. Previous work [42] concluded that SMS is slightly superior to DCS in terms of causality and the coefficient of variation of output interval (i.e. intra-stream synchronization quality), mainly due to the minor traffic overhead. Similarly, it can be deduced that the performance in terms of causality provided by M/S Scheme is better than the one in the other IDMS schemes due to the same reason.
9) *Flexibility*. Using M/S Scheme there is no option for selecting the reference output timing since it is taken from the one reported by the master destination. Conversely, the maestro, in SMS, and the distributed receivers, in DCS, can employ several dynamic policies for selecting an IDMS reference from the collected output timings (as the ones proposed in [17]). Furthermore, as in both SMS and DCS the session members can be divided into independent subgroups (sharing the same experience). In SMS, the maestro must collect the overall synchronization status during the session (of all the sub-groups). But in DCS, although the receivers collect all the reports from all the other receivers (multicast), they will only monitor those from the receivers belonging to the same sub-group. So, DCS outperforms the other IDMS schemes in terms of flexibility.
10) *Fairness*. M/S Scheme is suitable for applications in which a single destination has a certain priority level over the others. For example, in multi-party multimedia conferencing (e.g. synchronous e-learning), the chairperson’s (e.g. the teacher’s) terminal can be selected as the master destination, which directs to the attendees’ (students’) devices the required playout adjustments to get in sync. However, this scheme cannot treat all the destinations fairly. This problem is minimized when SMS or DCS are employed, because the reference output timing is selected after a comparison among the output timing of all the destinations. As an example, the study in [6] concluded that the effectiveness of the IDMS control in competitive games, in terms of fairness between players, could be improved by adjusting the overall output timing to the latest (slowest or more lagged) one. DCS may

outperform SMS in terms of fairness because asynchrony situations, which can cause an annoying effect to de-synchronized receivers, can be corrected earlier due to the minor network and processing delays. However, for that purpose, all the distributed receivers should be coordinated to select the same reference playout point for IDMS.

11) *Security*. Another major advantage of centralized architectures is that the presence of a server makes cheating difficult. In a completely distributed architecture (DCS), each participant takes its own decisions, resulting in a lack of control of what each one is doing or if they are honest or malicious participants. In M/S Scheme, this problem can be minimized if the IDMS operation of the master receiver is under control. In SMS, the maestro can use some mechanisms to check the validity of the arriving control packets and guarantee the overall synchronization status. Hence, cheating is more difficult in centralized schemes than in DCS. In each one of the considered IDMS schemes, the reporting of an erroneous playout point, either accidental or malicious, may lead to undesired behavior. According to the adopted model, extremely advanced/delayed playout information (e.g., several seconds) would produce large adjustments of the receivers' playout processes with the consequent significant loss of real-time or continuity perception. It would obviously affect the consistency, fairness and real-time interaction of the multimedia service. Therefore, synchronization entities (maestro in SMS, or each destination in DCS and in M/S schemes) should consider inconsistent playout information, exceeding configured limits (even though it comes from the master destination in M/S Scheme), as a malfunction service and reject that information in the calculation of the necessary playout adjustments (synchronization actions).

To summarize, a ranked comparison among the existing control schemes for IDMS is presented in Table 2, regarding all the factors considered in this Section. Since each one of them has its own strengths and weaknesses, the choice between them is largely application-dependent.

Table 2 Comparison among end-point based IDMS Schemes

		Factors									
		Robustness	Scalability	Traffic Overhead	Interactivity	Consistency	Causality	Coherence	Flexibility	Fairness	Security
Scheme	M/S	3	1	1	1	2	1	3	3	3	2
	DCS	1	2	3	2	3	3	2	1	1	3
	SMS	2	3	2	3	1	2	1	2	2	1

1. Best Scheme, 2. Good Scheme, 3. Worst Scheme

As it can be appreciated in the table, M/S Scheme can provide the best performance in terms of scalability, traffic overhead, interactivity (low delays) and causality, but presents serious drawbacks if some features such as robustness, coherence, flexibility and fairness must be provided. This scheme, however, can be suited in those scenarios in which the bandwidth availability is scarce, and also in those use cases in which a single participant has a certain priority level over the others, as in synchronous e-learning scenarios (in which the terminal of the teacher or the chairman should be selected as the master reference for IDMS).

DCS is a suited option for IDMS in those use cases in which high performance in terms of robustness, fairness, flexibility, scalability and interactivity (i.e. achieving stringent synchrony levels) is desirable, despite of a slight cost in terms of traffic overhead, consistency or security (see Table 2). We have found several DCS-based solutions adapted for networked multiplayer games (e.g. [11] and [14]). So, we can conclude that DCS can be an appropriate solution for controlled environments in which bandwidth availability is not a problem, and security aspects can be ensured.

As DCS requires that the distributed receivers implement the functionality of processing the incoming IDMS reports from all the other receivers and calculating the required IDMS adjustments to keep an overall synchronization status, it implies additional complexity to the receivers' terminals, which can result in an increase of the development costs of the IDMS solutions based on this signaling scheme (to take into account as an additional DCS drawback).

An important limiting factor for the previous two IDMS schemes (DCS and M/S) is the support of multicast feedback capabilities (i.e. the ability to exchange useful information for IDMS in a point-to-multipoint way) among the distributed receivers in most media streaming technologies, e.g. those in which Single Source Multicast (SSM) is employed. In such cases only the media server can transmit data in a multicast way. So, it could prevent the deployment of an IDMS solution based on DCS or M/S Schemes in some actual large-scale environments, such as IPTV broadcast distribution channels. In other scenarios, where small groups of users are watching video content synchronously, independently of other receivers or groups of receivers, then the adoption of a DCS or M/S Scheme may be an option. Actually, the ETSI specifications (see Section V) do explicitly allow the use of a DCS for IDMS in a peer-to-peer fashion.

Finally, we can observe that SMS is the best scheme in terms of consistency, coherence, and security, which are important aspects in most of the IDMS use cases. Contrariwise, the main weaknesses of using SMS for IDMS are scalability and interactivity. The first weakness can be significantly solved by using two control mechanisms: either dividing the session into logical groups (clusters), which may facilitate the IDMS management to the synchronization manager or maestro; or dynamically adjusting the transmission interval for the IDMS reports according to the number of active receivers in the session and the available bandwidth. The second weakness is not a crucial drawback in those scenarios that do not require stringent synchronicity levels. Also, previous work has showed the feasibility of an SMS for IDMS to keep the asynchrony within allowable limits (even more stringent levels that the ones required for Social TV were accomplished) in real scenarios [37].

Also, in some media streaming technologies, such as the ones using RTP/RTCP, distributed receivers send regularly feedback messages including QoS metrics (e.g. delay, jitter, packet loss information, etc.) to the media server, who can react accordingly (e.g. by adjusting its transmission timing or the media coding mechanism). If those feedback messages are extended to include useful information for IDMS, it would facilitate the deployment of an IDMS solution (as explained in Section V). This makes SMS the most practical alternative for IDMS, especially if the synchronization manager or maestro functionality is incorporated in the media server resources.

Therefore, taking into account all the above features, it can be concluded that SMS is, in general, the best-ranked scheme for IDMS. SMS is well-suited in those scenarios in which coherence is essential (all the receivers need to be almost simultaneously synchronized to the same reference timing), the network delay is not excessively large, and the number of participants is not too high, such as networked loudspeakers, phased array transducers and sound reinforcement systems (in which a central entity responsible for mixing, filtering and prioritizing functions must be included). SMS is also adequate for on-line election events (in which all the votes must be registered in a central control entity), and for distributed shared video watching scenarios and video wall (in which feedback control reports are usually sent from the receivers to the media server for QoS monitoring purposes).

Generally, in each specific use case in which IDMS mechanisms are required, the implementer or application developer must take into consideration the context and space in which the IDMS solution is going to be deployed and the requirements that must be accomplished. Accordingly, the relative importance of the previous factors must be weighted to meet the desired goals. For instance, an implementer can choose to give more preference to interactivity than to traffic overhead, or more to flexibility and robustness than to security, or more to coherence than to scalability, etc. Also, such decisions can vary depending on the situation in which the same type of media sharing application is going to be deployed. Therefore, no definitive rules can be given, but only indicative guidelines that can be followed in the design of an IDMS solution.

As stated in Section 2, apart from the adopted control schemes that determine the role played by each participant and their communication process for IDMS, two architectural approaches for IDMS can be followed, according to the location of the synchronization entities: network-based and terminal-based. Regarding network-based solutions, only one design approach was proposed in [27] to meet the need of IDMS in advanced large-scales IPTV services. Contrarily, terminal-based solutions have been more extensively used up to date, as reflected in Section 2. Accordingly, a qualitative comparison between both approaches is also included. Network-based approaches have the following advantages [27]:

- *Scalability*. A network-based approach can scale very well. As many end clients (User Equipment or UE) can be synchronized by a single edge node, the number of synchronization messages is limited. This will also limit the needed capacity at a synchronization server, at the cost of functionality on the edge nodes. Note that the same synchronization buffer for a media stream is also shared by many UEs.
- *UEs complexity*. The network-based approaches do not require UEs to support any IDMS solution, so current legacy devices can also be employed. As an example, IPTV companies can provide their customers a (free) set top box (STB), which can save the costs for those STBs, but at the cost of functionality in the network.
- *Synchronization control*. Since the edge node is under complete control of the IPTV provider, it can guarantee the stream synchronization for streams sent to the UEs. When implemented at the edge of the network, little or no delay differences will occur between UEs. Although jitter buffer settings between UEs may vary, this will not cause significant delay differences between them.
- *Delay*. Since buffering is done in the network, channel changing delays will not increase due to IDMS control. This assumes that all broadcast channels are being buffered for a short period of time at the edge nodes. If various UEs switch to a new channel as part of a Social TV experience, the new channel should also be delivered synchronously. This may mean that the new channel is delayed for certain UEs compared to other UEs not participating in the Social TV experience.

Obviously, network-based approaches also have some disadvantages. They will not work for over-the-top IPTV service since network control is required. Control could, of course, be offered by a network provider, but experience has shown that network providers are not eager to open their networks in this manner. Moreover, this solution is much more difficult to deploy in such cases in which the end users can be divided into different physically dispersed sub-groups (clusters), which must be separately synchronized, because the same media stream should be delayed differently for each sub-group (cluster) of users. Also, any delay differences introduced behind the synchronization point, are not yet taken into account and require further study. One possible solution for this drawback could be that the synchronization functionality could be divided into the network part (e.g. edge node) and the UE [27].

Summarizing, the main advantage of end-based approaches is that they do not require any changes to the network while the main advantage of network-based solutions is that they do not require any changes to the UEs. So, the discussed solutions have different rationales and impacts on the architecture of the content delivery network. Some solutions require updates to existing reference points and corresponding protocols. Other solutions require a new functional entity and a new associated reference point. Some of them are better suited to large-scale synchronization of commodity services, while other solutions are more cost-effective for services involving (perhaps many) small groups of users [9].

IV. IDMS standardization

Standardization of IDMS has been carried out within ETSI (European Telecommunications Standards Institute) TISPAN (Telecoms & Internet converged Services & Protocols for Advanced Networking), and is currently a milestone for the IETF AVTCORE WG (Internet Engineering Task Force - Audio/Video Transport Core Maintenance Working Group). Most of the earlier IDMS solutions described in [4] define new proprietary protocols, with specific control messages, that should increase the network load. Currently, many multimedia systems make use of standardized RTP/RTCP protocols (RFC 3550). The timestamp and sequence number mechanisms provided in RTP data packets are very useful to reconstruct original media timing, to reorder packets and to detect possible packet losses at the receiver side.

IDMS involves the collection, summarizing and distribution of RTP packet arrival and playout timings. As this information can be considered as a QoS metric (it can reflect the effect of jitter, network load, packet losses, clock skews/drifts, presentation skews, CPU overload, etc.), RTCP becomes a promising candidate for carrying out IDMS. Besides using RTCP for this monitoring purpose, in IDMS also control of the play-out by receivers is needed. Although RTCP is somewhat less suited for this second purpose, since this requires application-level control and using RTCP for this control purpose can be considered a form of layer-violation, it does make sense to use a single protocol for both the reporting and the control purpose. Also, the RTCP protocol is intended to be tailored through modification and/or additions in order to include profile-specific information required by particular applications, and the guidelines for this are in RFC 5968. This makes it a suitable protocol to be extended with IDMS-specific functionality.

Both ETSI TISPAN and the IETF AVTCORE workgroup have chosen this RTCP route. This section presents the evolution of the standardization process in both organizations.

ETSI TISPAN proposal

ETSI (TISPAN) is a major European-based standardization organization with significant operator involvement. It works on new specifications for Next-Generation Networks (NGN) and its associated services, working closely together with the 3rd Generation Partnership Project (3GPP). The first TISPAN IPTV standards have mainly focused on the basic IPTV services, such as broadcast and video-on-demand, re-using as much of the generic NGN components as possible. The ETSI TISPAN Release 3 specifications on IPTV have included many advanced interactive IPTV services, such as personalization, Social TV and synchronization features. The specifications describe IPTV use cases, requirements, architecture and protocol solutions. In this section, we reflect on the main topics from each one of these parts. ETSI TISPAN has specified both an NGN- (or IMS-) based IPTV architecture and a so-called Integrated IPTV subsystem. The NGN-based IPTV is mainly using the SIP (Session Initiation Protocol) for IPTV session setup en maintenance, whereas the Integrated IPTV subsystem is based on the HTTP (Hyper-Text Transfer Protocol). The section below is based on the NGN-based IPTV solution (the Integrated IPTV subsystem is in many aspects quite similar).

Use Cases and Solution

Reference [48] contains the service layer requirements and includes a variety of advanced IPTV use cases (the *"watching apart together"* use case as a prominent example, together with gaming and remote game show participation). The specification [48] does pose IDMS and the synchronization of media streams from different sources as a requirement for providing synchronization-sensitive interactive services. These use cases are mostly in the categories of low or medium synchronization, no very high requirements are posed to delay differences between various UEs. The protocol specification gives a delay difference of between 150 ms and 400 ms as a guideline for achieving transparent interactivity, based on ITU guidelines for interactivity in person-to-person communication.

Architecture

ETSI describes the architecture for IMS-based IPTV services in [49]. Figure 5 shows its main functional entities and reference points. TISPAN IDMS is designed, based on the existing release 2 specifications for IPTV. These release 2 specifications have used the SIP protocol for setting up broadcast sessions and have used the SIP protocol in combination with the Real Time Streaming Protocol (RTSP) for setting up video-on-demand or network-PVR sessions. Both these session control protocols use the Session Description Protocol (SDP) for describing various session attributes. The IDMS mechanism introduces two new functional entities and one new reference point, depicted in Fig.6a. This new sync reference point is for exchanging IDMS control messages between *Synchronization Clients* (*SCs*) on receivers and a *Media Synchronization Application Server* (*MSAS*) in the NGN-network, and is based on RTCP. For setting up synchronization sessions between various end users, the session mechanisms from release 3 are extended with IDMS attributes, the IDMS session becoming part of the broadcast or video-on-demand sessions. Either existing media sessions can be converted in a synchronization session, or new media sessions can be set up directly with synchronization enabled.

During a synchronization session, timing information on media reception and presentation at each SC is exchanged and instructions are sent on how much an

SC should adapt the media stream playout. On the one hand, the MSAS collects synchronization status information from the SCs, calculates delay settings instructions and sends these instructions to the clients. On the other hand, the SCs report on media arrival or presentation times to the MSAS and adjust the play-out based on instructions received from the MSAS. A requirement for an SC is that it is clock-synchronized (for example, by using NTP – Network Time Protocol).

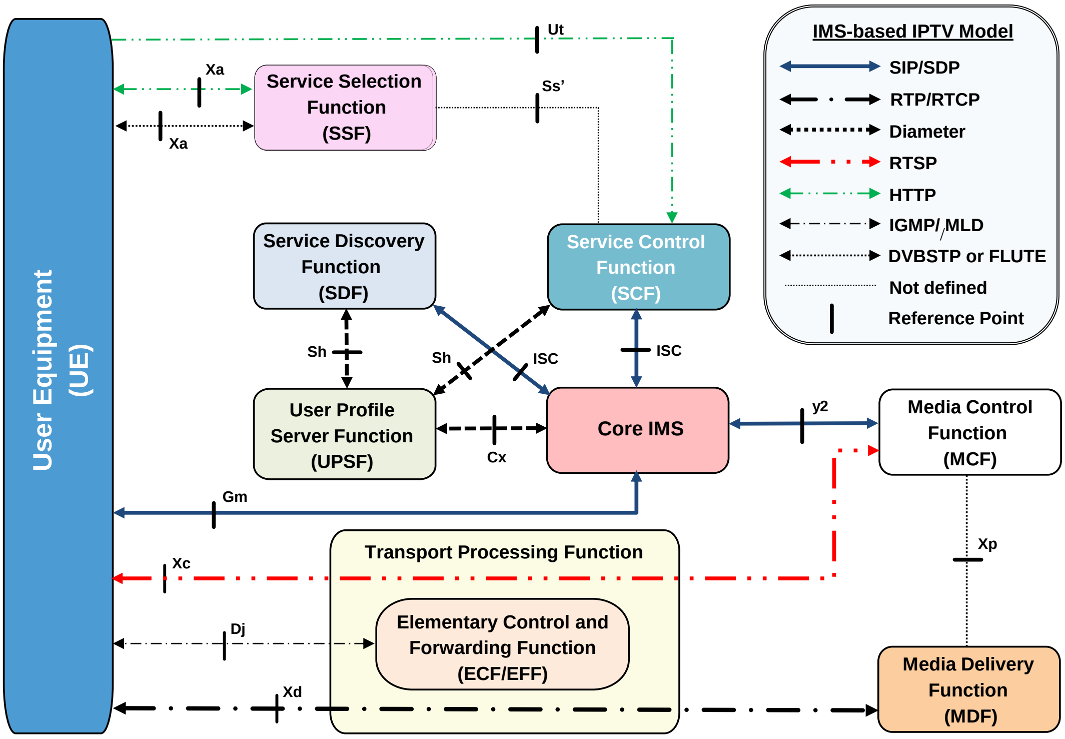
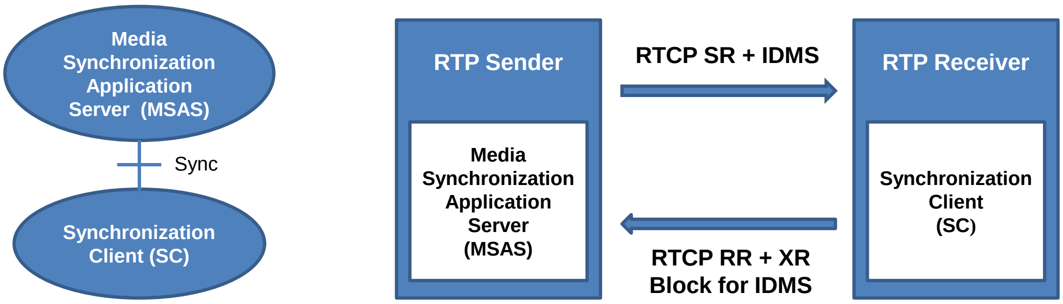

User Equipment (UE)
Ut
Xa
Xa
Service Selection Function (SSF)
Ss'
IMS-based IPTV Model
SIP/SDP
RTP/RTCP
Diameter
RTSP
HTTP
IGMP/MLD
DVBSTP or FLUTE
Not defined
Reference Point
Service Discovery Function (SDF)
Service Control Function (SCF)
Sh
Sh
ISC
ISC
User Profile Server Function (UPSF)
Cx
Core IMS
y2
Media Control Function (MCF)
Gm
Xp
Transport Processing Function
Xc
Dj
Elementary Control and Forwarding Function (ECF/EFF)
Xd
Media Delivery Function (MDF)

Fig. 5 ETSI TISPAN functional entities and reference points in the IMS-based IPTV architecture [49]


Media Synchronization Application Server (MSAS)
Sync
Synchronization Client (SC)
RTP Sender
Media Synchronization Application Server (MSAS)
RTCP SR + IDMS
RTCP RR + XR Block for IDMS
RTP Receiver
Synchronization Client (SC)

(a) ETSI TISPAN [50]

(b) IETF Internet Draft [47]

Fig. 6 Functional entities and reference point for IDMS

The algorithms to calculate the synchronization settings instructions from collected synchronization status information have not been specified, but left to vendor-specific implementations. This allows vendors to differentiate their solution from that of other vendors.

ETSI TISPAN does allow various implementations of the IDMS functional architecture, as described in the specifications. The basic implementation is of an SMS scheme, where the SC is implemented in the receiver and the MSAS is implemented in the network. The ETSI specifications specify the MSAS as a functional entity separate from the Media Distribution Function (MDF), the ETSI term for media source, but implementations can co-locate the MSAS function there. In another implementation, SCs still reside in the User Equipments (UEs, ETSI term for receiver) but the MSAS is also co-located with the SC in one of the UEs. In another implementation, the SCs are implemented as part of the network nodes, as described earlier in this paper [27]. In both mappings, the session-related part of the MSAS is part of the Service Control Function, or exists as a dedicated IMS application server.

ETSI TISPAN, additionally, specifies an IDMS solution for the modification or re-origination of streams, which may be the case when one IPTV implementation serves both HD streams and SD streams, using transcoding. Such modifications or re-originations may change the RTP timestamp offset between different streams and thus can cause problems for IDMS. Additional measures are then required, such as placing an additional media-stream modifying SC’ within the functional entities where media streams are modified. This SC' can then deliver correlation information to the MSAS, containing the timing relation between various streams, e.g. between an HD and an SD stream.

This ETSI TISPAN IDMS architecture shares some properties of other recent application-layer service capabilities for IPTV, such as solutions for retransmission (RET) or forward error correction (FEC). Many IPTV operators are currently looking at or implementing such QoS enhancement technologies, on top of their current legacy IPTV solutions. The IDMS solution specified by ETSI TISPAN can, similarly to the RET and FEC technologies, be added to an existing

IPTV solution. Many set-top boxes allow software modifications to be performed remotely through the use of remote management protocols such as TR-069, and can thus be equipped with an IDMS client without having to provide a new physical set-top box to end-users. So, even though the solution is part of the ETSI TISPAN IPTV release 3 specifications, the part on IDMS can also be implemented and used separately from other features of these specifications.

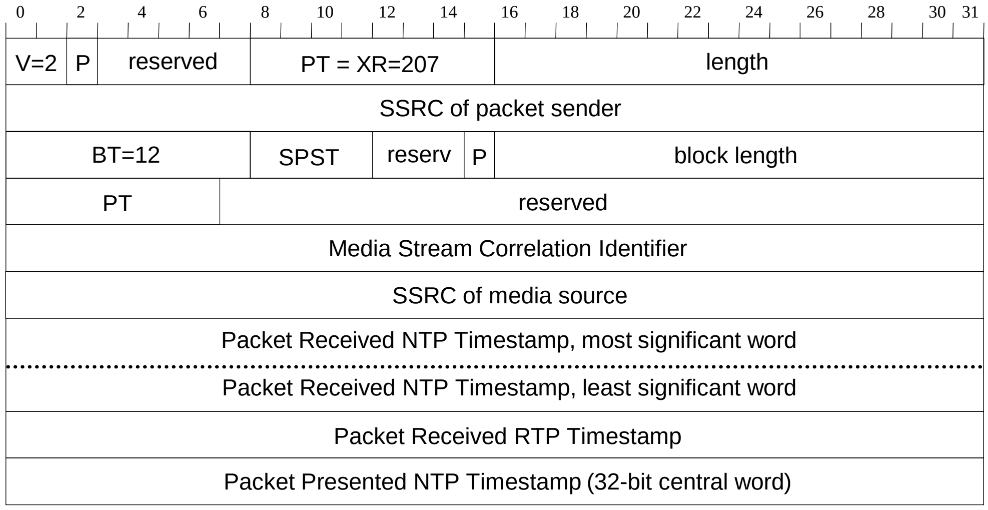
Protocols

The ETSI IDMS protocol is specified as a two-part solution in [50]. The one part is the setup, maintenance and teardown of synchronization sessions among the users involved in a synchronous shared media experience. These sessions are set up using SIP and SDP (Session Description Protocol), using the G_m and *ISC* reference points, for broadcast, or using a combination of SIP and RTSP (Real Time Streaming Protocol), also using SDP, for content on-demand. The exception to this is the network-based synchronization. Since network nodes are not involved in the media sessions, this synchronization setup requires the network nodes to be pre-configured with regard to IDMS. The synchronization session information is contained in the SDP media description. This SDP contains the following items:

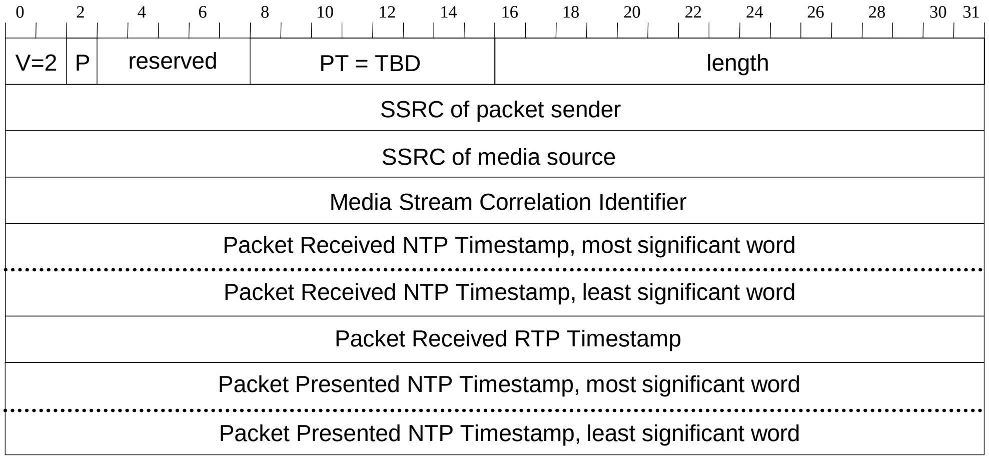
- The *address of the MSAS* to be used for the synchronization session. This is allocated by the Service Control Functions (SCFs) and will usually be the same for all UEs in a synchronization group. Alternatively, various MSAS's may be hierarchically or otherwise coupled to allow for SCs in a certain synchronization group to use a different MSAS.
- A *SyncGroupId*, which specifies the synchronization group. The SyncGroupId can be allocated by the SCFs or it can be indicated by the UE. This is similar to the use of a conference-ID in conference calls, where each user has to enter the same conference-ID to become part of the same conference call.
- In case of content on-demand, *the SSRC (Synchronization Source) of the media stream*. It can be used to correlate various RTCP messages, since in unicast media streams, the SSRCs of the various streams will be different, where in the broadcast scenario using IP multicast, every UE receives the media stream with the same SSRC.

Synchronization sessions can be ended in various ways. The various SCs can leave a synchronization session by using a SIP re-INVITE containing the media description but omitting the synchronization parameters. If only one SC remains in a synchronization session, the MSAS will terminate that session by sending a similar re-INVITE to that last remaining SC. Alternatively, a synchronization session can be ended if an SC ends the entire media session.

After configuration of network elements or synchronization session setup for UEs, synchronization messages can be exchanged between SCs and their MSAS. SCs send synchronization status information to the MSAS, indicating the arrival time and/or presentation of media packets to the MSAS. The MSAS sends synchronization settings instructions to the SCs. After debating the various protocol options for exchange of these control packets, such as using SIP, HTTP and RTCP, ETSI TISPAN chose RTCP as the protocol for this communicating of status and delay information. Although ETSI TISPAN does support the use of MPEG Transport Streams (TS) directly on top of UDP, since RTCP is used, IDMS in ETSI TISPAN requires the use of RTP as transport protocol for the media. A new RTCP XR block type has been specified for the purpose of synchronization (Figure 7a). An IANA registration has been performed based on the ETSI TISPAN specifications, making the RTCP XR block available to a wider community.


0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
31
V=2
P
reserved
PT = XR=207
length
SSRC of packet sender
BT=12
SPST
reserv
P
block length
PT
reserved
Media Stream Correlation Identifier
SSRC of media source
Packet Received NTP Timestamp, most significant word
Packet Received NTP Timestamp, least significant word
Packet Received RTP Timestamp
Packet Presented NTP Timestamp (32-bit central word)

a) IDMS RTCP XR Block, in both ETSI TISPAN ([50]) and Internet Draft ([47])


0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
31
V=2
P
reserved
PT = TBD
length
SSRC of packet sender
SSRC of media source
Media Stream Correlation Identifier
Packet Received NTP Timestamp, most significant word
Packet Received NTP Timestamp, least significant word
Packet Received RTP Timestamp
Packet Presented NTP Timestamp, most significant word
Packet Presented NTP Timestamp, least significant word

b) RTCP Packet Type for IDMS (IDMS report), in Internet Draft ([47])

Fig. 7 RTCP packets for IDMS

This new block type contains the default RTCP XR headers, followed by the new block type. The new block type contains the SyncGroupID in the Media Stream Correlation Identifier, it contains the SSRC of the media source, it contains an RTP timestamp as a reference to which RTP packet the report belongs, and it contains at least the packet received time and optionally the packet presented time. Although packet presentation times will allow for a higher level of synchronization, the use cases in ETSI TISPAN do not pose such high-level requirements. Therefore the packet received times, which are much easier available in a receiver, are the basis of the ETSI TISPAN IDMS solution.

For synchronization status information, the use of this block type is straightforward. For synchronization settings instructions, an XR report should be interpreted as a status information report of the synchronization reference point (e.g. the one of the most lagged SC). The MSAS can either match the most lagged receiver, but could also insert additional delay to be able to deal with future delay variations, or use some other mechanism. IDMS requires all SCs in each group to match this reference point.

IETF Internet Draft on IDMS [47]

Besides standardization in ETSI TISPAN, standardization of the RTCP-based IDMS protocol is currently being carried out within the Internet Engineering Task Force (IETF), in the AVTCORE working group [47]. This is the core group that is responsible for the RTP and accompanying RTCP protocol. Even though ETSI

TISPAN has done the first work on standardizing RTCP usage for IDMS, it is more suitable to continue this work within the IETF, where most RTCP extensions are developed. Also, the ETSI proposal is a dedicated solution for use in large-scale IPTV deployments, with low to medium level synchronization requirements. Other services such as Internet-based video services may also benefit from IDMS, and other use cases require higher levels of synchronization, and are not supported by the ETSI solution.

A first informational ID on IDMS [51], dated September 2010, presented the work done in ETSI to the IETF AVT group, with the purpose of having a discussion on the need for work in this area in the IETF. There was enough support for work on IDMS, and the work was accepted as a standards-track working group item within the AVTCORE WG. The current version is draft-brandenburg-avtcore-rtcp-for-idms-03 [47]. This work uses the ETSI TISPAN IDMS specification as a base, and extends on that, while arranging for interoperability between the two sets of specifications.

Use Cases

The work in the IETF is mainly based on the same Social TV use case as in ETSI but the goal in the IETF is to have a more general applicable IDMS solution. Not only should IDMS work for services other than IPTV, it should also support more accurate synchronization and be applicable to other use cases than Social TV, such the ones presented in Section 1. Use cases explicitly mentioned in the ID are for example a video-wall and networked loudspeakers. Such use cases, where synchronization of media presentation in a single physical location has to be achieved, require synchronization levels in the sub-millisecond range.

Architecture

In the ID, the functions of SC and MSAS are defined as part of the RTP receiver and RTP sender, respectively (Figure 6b). Optionally, the MSAS can also be part of a receiver. The ID does keep to the terminology introduced by ETSI TISPAN, but in this sense limits the implementation options. ETSI architectures are normally functional architectures. By specifying functions and reference points, ETSI solutions aim for scalability. Implementors of their specifications can choose to have all functions implemented separately, or to combine several functions in a single implementation. For IDMS, ETSI has specified both the SC and the MSAS as separate from for example the media delivery functions or the user equipment. The RTP/RTCP framework specified by the IETF is defined more from the viewpoint of a media sender and a media receiver, and the ID on IDMS is in line with this.

Within the IETF, a policy is maintained to use XR blocks only for monitoring and reporting purposes and not for control purposes. Therefore, the IETF has specified a separate IDMS report block for carrying delay settings instructions. This signifies an important change compared to the ETSI solution, where a single RTCP XR block is used for both reporting and control purposes, using the SPST parameter to indicate the usage.

Protocol

The protocol in the ID is based on the protocol as specified by ETSI. The ID describes the use of the ETSI specified XR block for reporting on RTP packet arrival time and presentation time, which is contained in the ID for informational purposes. RFC 5968 states that the only valid reason to create a new RTCP packet type is if the required functionality would not be appropriate as part of one of the current packet types (such as XR blocks). Thus, for sending synchronization settings instructions to receivers, a new RTCP packet type is introduced, called RTCP IDMS report (Figure 7b). This report contains mostly the same elements as the ETSI TISPAN specified XR block. Some headers can be removed, because it is now a separate RTCP report block, and the packet presentation time element has been changed, see also below. The use of this IDMS report can be declared using the new SDP parameter "*rtcp-idms*", specified in the ID.

Because the use cases included in the IETF ID (and other use cases presented in Section I) have requirements in the high and very high synchronization levels, the firstly proposed 32 bit presentation timestamp [51] does not offer the level of granularity needed. For use cases such as network stereo loudspeakers or phased array transducers, effects may be noticeable with shifts of 10 microseconds or smaller. For this purpose, the last version of the ID [47] introduced a 64 bit presentation timestamp as part of the IDMS report block. This signified a definite change compared to the ETSI protocol, needed to support such very high synchronization levels.

ETSI posed a requirement of the use of NTP for synchronizing the wallclocks of the various receivers. In a managed operator IPTV deployment this is sufficient, since the operator will also provide the NTP servers. In the Internet environment, it is known that, although the NTP protocol can provide very accurate clock synchronization, the use of NTP may not lead to very accurate clock synchronization. The main reason for this is the use of different NTP servers by different receivers. NTP servers are not always set up correctly, and can thus provide wrong clock time to receivers. A second cause of clock deviation is clock skew within receivers. Also, not all receivers may support NTP for clock synchronization, but may support other protocols for this same purpose.

To help receivers sort out these timing issues, the ID refers to a new SDP attribute called *"clocksource"*, specified in [52], which is derived from the IDMS Internet Draft. This attribute allows receivers to declare if they support clock synchronization, which clock sources they support for this and which was used latest for synchronization. This can be used as an indication to the clock accuracy for a given receiver, and also allows receivers in a synchronization group to choose a common clocksource. Currently the defined sources are local (meaning no support for synchronization exists), NTP, GPS, GAL and PTP (Precision Time Protocol). This is an extendable list to be registered with IANA, so future clock synchronization technologies can be added as well.

Interoperability between the IETF specifications and the ETSI specifications is arranged for in the ID. The XR block for reporting on RTP packet arrival and presentation times in the ID is fully compatible with the ETSI defined XR block. Further, if all receivers and the media sender involved in an IDMS session support the new IETF-defined IDMS report for synchronization settings instructions, they must use that. Receivers may still support the ETSI specified XR block for this purposes as well, but only as a backwards compatibility mechanism with ETSI.

This solution prevents a real forking of the RTCP-based IDMS solution, and will help in the adoption of a single solution by the industry.

One other detail has been dealt with in the ID: the issue of leap seconds, also referring to [52]. Some time sources, such as NTP time, operating system clocks and other UTC (Coordinated Universal Time) references include leap seconds (though the ITU is studying a proposal which could eventually eliminate leap seconds from UTC). A leap second is a positive or negative one-second adjustment to the Coordinated Universal Time (UTC) time scale that keeps it close to mean solar time. If synchronization sessions are ongoing when a leap-second is introduced, receivers should be careful not to report too close to this occurrence. Any reports too close to a leap second introduction can be misinterpreted because the clocks of senders and receivers of such reports can be misaligned. Also, if the time-source of some receivers is immediately aware of the leap-second, whereas others use a time-source that is not, a error of 1 second is introduced in the synchronization. This awareness of leap seconds and thus this error between various receivers' clocks can occur over a longer period of time, it may take several days or longer before every receiver has adjusted for a leap second. This leap second problem can be avoided by using a clock reference with a timescale which does not include leap seconds, such as IEEE 1588, GPS and other TAI (International Atomic Time) references.

V. Conclusions and future research.

In this paper we have focused on a multimedia synchronization type, called IDMS, that has been gaining popularity in recent years, specially due to the rise of social networking applications. The importance of IDMS has been emphasized and, although Social TV is the most prominent use case in which IDMS is useful, up to 19 use cases in which IDMS is needed have been presented and ranked depending on their synchronization requirements. The most popular schemes proposed by researchers in the last years to achieve IDMS have been presented and compared qualitatively showing their advantages and disadvantages. Moreover, as a proof of the importance of IDMS, the standardization efforts from ETSI TISPAN and IETF organizations have been summarized, in which the authors have been, and still are, participating actively. Also, standardization of

IDMS will help the uptake of implementations and of the interoperability between various implementations, ensuring a more widespread use of IDMS in practice.

Future research on media synchronization, among which IDMS, is ongoing. New streaming protocols are developed and put to use, such as HTTP Adaptive Streaming, new delivery methods such as segmented video delivery are under research, and many so-called second screen applications are being developed. All these will require synchronization, and applying synchronization to all these new technologies will require future research.

AKNOWLEDGEMENTS

This work has been financed, partially, by Universitat Politècnica de Valencia (UPV), under its R&D Support Program in PAID-05-11-002-331 Project and in PAID-01-10, and by TNO, under its Future Internet Use Research & Innovation Program. Authors also want to thank Kevin Gross who provided us with some of the use cases included in Section I.2.

REFERENCES

[1] Kernchen R, Meissner S, Moessner K, Cesar P, Vaishnavi I, Boussard M, Hesselman C. (2010) Intelligent Multimedia Presentation in Ubiquitous Multidevice Scenarios, IEEE Multimedia, vol.17, no.2, pp.52-63, April-June 2010.

[2] Vaishnavi I, Cesar P, Bulterman D, Friedrich O, Gunkel S, Geerts D (2011) 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.

[3] Geerts D, Vaishnavi I, Mekuria R, Van Deventer O, Cesar P (2011) Are we in sync?: synchronization requirements for watching on-line video together, CHI '11, New York (USA), May 2011.

[4] Boronat F, Lloret J, García M (2009) Multimedia group and inter-stream synchronization techniques: A comparative study, Inf. Syst. 34, 1, 108-131, March 2009.

[5] Chen M (2003) A low-latency lip-synchronized videoconferencing system, SIGCHI Conference on Human Factors in Computing Systems, CHI’03, ACM, 464-471, New York, 2003.

[6] Ishibashi Y, Tasaka S, Ogawa H (2003) Media Synchronization Quality of Reactive Control Schemes, IEICE Transactions on Communications, Vol.E86-B, No.10, 3103-3113, October 2003.

[7] Ademoye OA, Ghinea G (2009) Synchronization of Olfaction-Enhanced Multimedia, IEEE Transactions on Multimedia, vol.11, no.3, pp.561-565, April 2009.

[8] Cesar P, Bulterman DCA, Jansen J, Geerts D, Knoche H, Seager W (2009) Fragment, tag, enrich, and send: Enhancing social sharing of video, ACM Trans. Multimedia Comput. Commun. Appl. 5, 3, Article 19, 27 pages, August 2009.

[9] Van Deventer MO, Stokking H, Niamut OA, Walraven FA, Klos VB (2008) Advanced Interactive Television Service Require Synchronization, IWSSIP 2008, Bratislava, June 2008.

[10] Premchaiswadi W, Tungkasthan A, Jongsawat N (2010) Enhancing learning systems
by using virtual interactive classrooms and web-based collaborative work. In: Proceedings of the IEEE Education Engineering Conference (EDUCON 2010), Madrid, Spain, pp 1531-1537.

[11] Diot C, Gautier L (1999) A Distributed Architecture for Multiplayer Interactive Applications on the Internet, IEEE Network, vol. 13, nº 4, pp. 6-15, July/August 1999.

[12] Mauve M, Vogel J, Hilt V, Effelsberg W (2004), Local-Lag and Timewarp: Providing Consistency for Replicated Continuous Applications, IEEE Transactions on Multimedia, Vol.6, No.1, February 2004.

[13] Hosoya K, Ishibashi Y, Sugawara S, Psannis KE (1009) Group synchronization control considering difference of conversation roles, IEEE 13th International Symposium on Consumer Electronics, ISCE '09, 948-952, May 2009.

[14] Roccetti M., Ferretti S., Palazzi C. (2008) The Brave New World of Multiplayer Online Games: Synchronization Issues with Smart Solution, 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC), pp. 587-592, May 2008.

[15] Ott DE, Mayer-Patel K (2007) An open architecture for transport-level protocol coordination in distributed multimedia applications. ACM Trans. Multimedia Comput. Commun. Appl. 3, 3 (Aug. 2007), 17.

[16] Boronat F, Montagud M, Guerri JC (2009) Multimedia group synchronization approach for one-way cluster-to-cluster applications, IEEE 34th Conference on Local Computer Networks, LCN 2009, 177-184, Zürich, October 2009.

[17] Boronat F, Montagud M, Vidal V (2011) Smooth Control of Adaptive Media Playout to Acquire IDMS in Cluster-based Applications, IEEE LCN 2011, pp. 617-625, Bonn, October 2011.

[18] Huang Z, Wu W, Nahrstedt K, Rivas R, Arefin A (2011) SyncCast: synchronized dissemination in multi-site interactive 3D tele-immersion, In Proc. of MMSys, USA, February 2011.

[19] Kim S-J, Kuester F, Kim K (2005) A global timestamp-based approach for enhanced data consistency and fairness in collaborative virtual environments, ACM/Springer Multimedia Systems Journal, Vol. 10 (3), pp. 220-229.

[20] Schooler E (1993) Distributed Music: A Foray into Networked Performance, International Network Music Festival, Santa Monica, CA (Sept 1993).

[21] Y. Miyashita, Y. Ishibashi, N. Fukushima, S. Sugawara, and K. E. Psannis (2011) QoE assessment of group synchronization in networked chorus with voice and video, in Proc. IEEE TENCON'11, pp. 393-397, Nov. 2011.

[22] Hesselman C, Abbadessa D, Van Der Beek W et al (2010) Sharing enriched multimedia experiences across heterogeneous network infrastructures, IEEE Communications Magazine, Vol.48, no.6, pp.54-65, June 2010.

[23] Montpetit M, Klym N, Mirlacher T (2011), The future of IPTV - Connected, mobile, personal and social, Multimedia Tools and Applications Journal, Vol. 53 (3), 519-532, 2011.

[24] Cesar P, Bulterman DCA, Jansen J (2009) Leveraging the User Impact: An Architecture for Secondary Screens Usage in an Interactive Television Environment, ACM/Springer Multimedia Systems, Vol. 15 (3), pp.127-142.

[25] Lukosch S. (2003) Transparent Latecomer Support for Synchronous Groupware, In Proceedings of 9th International Workshop on Groupware (CRIWG), Grenoble (France), pp. 26-41, September 2003.

[26] Steinmetz R (1996) Human Perception of Jitter and Media Synchronization, IEEE Journal On Selected Areas In Communications, Vol.14, No.1, 61-72, January 1996.

[27] Stokking H, Van Deventer MO, Niamut OA, Walraven FA, Mekuria RN (2010) IPTV inter-destination synchronization: A network-based approach, ICIN'2010 , Berlin, October 2010.

[28] Mekuria RN (2011), 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.

[29] Pitt, Ian, CS2511 Usability Engineering Lecture Notes, Localisation of Sound Sources, http://web.archive.org/web/20100410235208/http:/www.cs.ucc.ie/~ianp/CS2511/HAP.html

[30] J. Nielsen (1994). Response Times: The Three Important Limits. Available: http://www.useit.com/papers/responsetime.html.

[31] ITU-T Rec. G.1010 (2001). End-user multimedia QoS categories. International Telecommunication Union, Geneva, Switzerland.

[32] Biersack E, Geyer W (1999) Synchronized delivery and playout of distributed stored multimedia streams, ACM/Springer Multimedia Systems, Vol. 7 (1), pp. 70-90.

[33] Xie Y, Liu C, Lee MJ, Saadawi TN (1999) Adaptive multimedia synchronization in a teleconference system, ACM/Springer Multimedia Systems, Vol. 7 (4), pp. 326-337.

[34] Laoutaris N, Stavrakakis I (2002) Intrastream synchronization for continuous media streams: a survey of playout schedulers, IEEE Network Magazine, 16 (3), 30-40, 2002.

[35] Ishibashi Y, Tsuji A, Tasaka S (1997) A Group Synchronization Mechanism for Stored Media in Multicast Communications, Proc. of the INFOCOM '97, Washington April 1997.

[36] Ishibashi Y, Tasaka S (1997) A group synchronization mechanism for live media in multicast communications, IEEE GLOBECOM'97, pp. 746–752, November 1997.

[37] Boronat F, Guerri JC, Lloret J (2009) An RTP/RTCP based approach for multimedia group and inter-stream synchronization, Multimedia Tools and Applications Journal, Vol. 40 (2), 285-319, June 2008.

[38] Ishibashi I, Tasaka S (1999) A distributed control scheme for group synchronization in multicast communications, Proc. of International Symposium Communications, Kaohsiung (Taiwan), pp. 317-323, November 1999.

[39] Lu Y, Fallica B, Kuipers FA, Kooij RE, Van Mieghem P (2009) Assessing the Quality of Experience of SopCast, Int. J. Internet Protoc. Technol, Vol.4 (1), March 2009.

[40] Shamma DA, Bastea-Forte M, Joubert N, Liu Y (2008) Enhancing online personal connections through synchronized sharing of online video, ACM CHI'08 Extended Abstracts, Florence, April 2008.

[41] Ishibashi Y, Tasaka S (2002) A distributed control scheme for causality and media synchronization in networked multimedia games, Proc. 11th International Conference on Computer Communications and Networks, Miami (USA), pp. 144-149, October 2002.

[42] Ishibashi Y, Tomaru K, Tasaka S, Inazumi K (2003) Group synchronization in networked virtual environments, Proc. Of the 38th IEEE International Conference on Communications, Alaska (USA), pp. 885–890, May 2003.

[43] Tasaka S, Ishibashi Y, Hayashi M, (2002) Inter–destination synchronization quality in an integrated wired and wireless network with handover, IEEE GLOBECOM 2002, pp. 1560- 1565 vol.2, Nov. 2002.

[44] Kurokawa Y, Ishibashi Y, Asano T, (2007) Group synchronization control in a remote haptic drawing system, Proc. of IEEE International Conference on Multimedia and Expo, Beijing (China), pp. 572-575, July 2007.

[45] Hashimoto T, Ishibashi Y (2006) Group Synchronization Control over Haptic Media in a Networked Real-Time Game with Collaborative Work, Netgames'06, Singapore, October 2006.

[46] Nunome T, Tasaka S (2005) Inter-destination synchronization quality in a multicast mobile ad hoc network, Proc. of IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, Berlin (Germany), pp. 1366-1370, September 2005.

[47] Brandenburg R. van, Stokking H, Van Deventer MO, Boronat F., Montagud M., Gross K. (2012), RTCP for inter-destination media synchronization, draft-brandenburg-avtcore-rtcp-for-idms-03.txt, IETF Audio/Video Transport Core Maintenance Working Group, Internet Draft, March 9, 2012.

[48] ETSI TS 181 016 V3.3.1 (2009-07), Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Service Layer Requirements to integrate NGN Services and IPTV.

[49] ETSI TS 182 027 V3.5.1 (2011-03), Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IPTV Architecture; IPTV functions supported by the IMS subsystem.

[50] ETSI TS 183 063 V3.5.2 (2011-03), Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IMS-based IPTV stage 3 specification.

[51] Brandenburg R. van, et al. (2010), RTCP XR Block Type for inter-destination media synchronization, draft-brandenburg-avt-rtcp-for-idms-00.txt, IETF Audio/Video Transport Working Group, Internet Draft, September 24, 2010.

[52] Williams A., et al. (2012), RTP Clock Source Signalling, draft-williams-avtcore-clksrc-00, IETF Audio/Video Transport Working Group, Internet Draft, February 28, 2012.