Implementation and Evaluation of an M/S Scheme for Inter-Destination Multimedia Synchronization (IDMS)

Mario Montagud, Fernando Boronat

Integrated Management Coastal Research Institute, Universitat Politècnica de Valencia

C/ Paraninfo 1, 46730, Grao de Gandía, Valencia (Spain)

Tel: +34+962 849 341 E-mail: mamontor@posgrado.upv.es; fboronat@dcom.upv.es

Received: November 11, 2011 Accepted: November 30, 2011 Published: December 18, 2011

DOI: 10.5296/npa.v3i3.1063 URL: http://dx.doi.org/10.5296/ npa.v3i3.1063

Abstract

Nowadays, the media consumption model is changing from passive isolated activities towards dynamic group shared experiences. People separated in space can already interact and collaborate within the context of simultaneous media content delivery (e.g. they can launch a real time conversation during a shared video watching). However, to stimulate the acceptance and usability of those socially aware interactive media services, synchronous communications (i.e. consistent playout states) across separated locations must be guaranteed. Inter-Destination Multimedia Synchronization (IDMS) has become a key research topic to enable geographically dispersed consumers to experience a satisfying feeling of togetherness. Several (distributed and centralized) approaches to perform IDMS have been adopted by the research community up to date. In this paper, a Master/Slave (M/S) Control Scheme for IDMS is implemented and tested, thus analyzing its strengths and weaknesses for distributed networked environments. Simulation results prove the effectiveness and suitability of the proposed M/S Scheme in those scenarios in which scalability, consistency, interactivity and low control traffic overhead features must be provided.

Keywords: Inter-Destination Multimedia Synchronization, M/S Scheme, RTP/RTCP, Simulation.

1. Introduction

The media consumption paradigm is undergoing a radical transformation. Nowadays, emerging ubiquitous multimedia applications allow direct social interaction and communication services within the context of simultaneous media content delivery. In those scenarios, geographically distributed users must be almost simultaneously aware of specific events (e.g. a goal in an on-line football match) to guarantee a group shared media experience with a satisfying feeling of togetherness. Otherwise the involved conversation and interaction patterns would be broken. Therefore, concurrently synchronized playout points must be ensured in each one of the separated users. This process is known as Inter-Destination (also Group or Multi-Point) Multimedia Synchronization (IDMS) [1], and it 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, etc.). IDMS is becoming essential in a variety of synchronous media sharing applications, such as: as e-learning, networked multi-player games, 3D tele-immersion (3DTI), and Social TV.

Up to date, several (centralized and distributed) approaches to carry out IDMS have been employed by the research community. In this paper, we have modified our preliminary RTCP-based IDMS approach ([2] and [3]) to adopt and follow a Master/Slave (M/S) Scheme. In this centralized approach there is a designated master destination whose playout timing is used to adjust those of the slave receivers. Receivers are coarsely synchronized at the beginning of the playout and each time an out of synchronization[1] situation (exceeding an allowable threshold) is detected. The strengths and weaknesses of an M/S Scheme for IDMS are qualitatively discussed, considering some key factors. Simulation results show the effectiveness of the proposed M/S Scheme for IDMS (using two different reactive sync techniques) in those social media sharing applications in which scalability, low traffic overhead and, especially, interactivity and strict sync requirements must be provided.

The structure of the paper is as follows. In the next section some related works are discussed. Section 3 analyzes the main advantages and disadvantages of an M/S Scheme for IDMS. In Section 4, some challenges to perform IDMS are introduced. Section 5 introduces the effect of playout rate imperfections over the media sync. Next, in Section 6, the implementation of the proposed M/S Scheme and its integration in our RTCP-based IDMS approach are described. Section 7 gives some performance results, and Section 8 is a conclusion and a discussion of future work.

2. Related Work

Several approaches to perform IDMS have been adopted by the research community up to date. Depending on the location of the sync functionality, they can be classified into terminal-based and network-based approaches.

In the former, an end-user device must report on arrival/presentation times of media

[1] From now on, we will abbreviate ‘synchronization’ as ‘sync’.

packets belonging to a specific media stream, and (one or several) sync managers are used to collect those control reports and to compute temporal discrepancies among the involved clients. As well, end-user devices must perform playout adjustments to acquire IDMS according either to their own calculations or as a result of the reception of a new control packet. In the later, the sync functionality is implemented in network edge nodes. Also, at a higher level, a sync manager must be used to control the output timing of the edge nodes.

In [1], the most exhaustive qualitative survey of earlier IDMS works was published. Generally, three end-based control schemes have been identified to perform the sync control (Fig.1 and Fig.2): two centralized (Master/Slave or M/S and Synchronization Maestro Scheme or SMS) and one distributed (Distributed Control Scheme or DCS). In this section their basic operation, as well as a set of IDMS solutions in which those control schemes have been employed are presented. Readers are referred to [1] for a more complete description and classification.

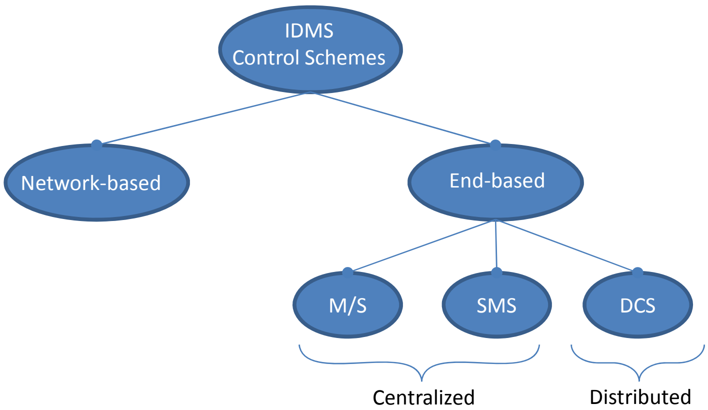

Figure 1. IDMS Schemes Classification.

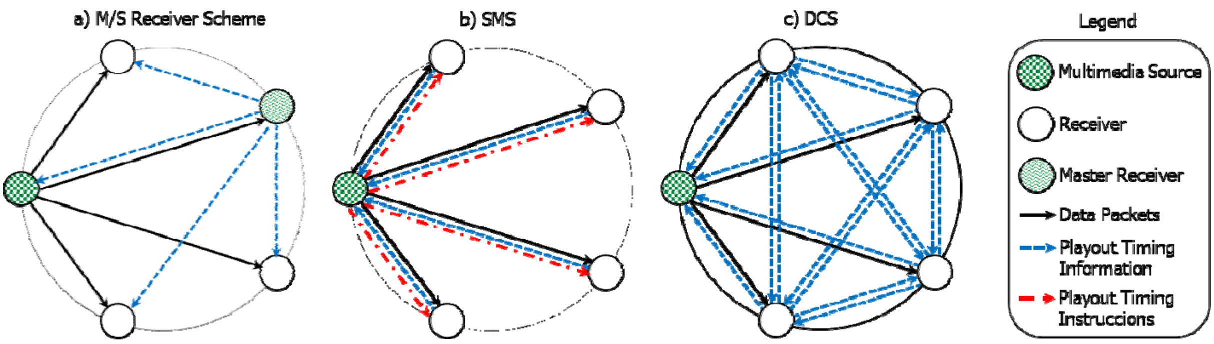

Figure 2. End-based IDMS Control Schemes.

In M/S Scheme (Fig.2a), receivers are differentiated into master (one) and slave receivers (the rest). Only 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 Media Units or MUs, e.g. video frames or voice samples) to the reference playout process of the master. This signaling scheme was introduced for the first time in [4], and more recently applied in [5] and [6]. In [5], an evolved IDMS algorithm has been adapted to achieve coherent execution of a specific user's actions in 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 be executed at other clients within bounded tolerance limits). The study in [6] is focused on testing the effect of de-synchronization over the user experience in Social TV scenarios. It was concluded that asynchronies (playout time differences) up to 1 second might not be perceptible by users while communicating using either chat messaging or voice conferencing services.

SMS (Fig.2b) is based on the existence of a synchronization maestro or manager (that can be the source or one real or fictitious receiver), which gathers the playout information of all the active receivers in a media session and corrects their playout timing by distributing new adapted control messages. In order to do this, each receiver sends (unicast) the 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. Note that in an M/S Scheme only the master receiver sends (multicast) control messages about playout timing (not all of them) and the necessary playout adjustments are directly calculated (apart from performed) by the (slave) receivers (there is no any sync manager or maestro).

Moreover, in SMS, the receivers can be also classified into an M/S scheme regarding the reference output timing, in which the playout timing of the master receiver is selected as the sync reference for adjusting those of all the other (slave) receivers. The master receiver role could also be dynamically exchanged between receivers according to the network conditions, thus allowing M/S switching technique [1]. In [2], authors presented a preliminary version of an RTCP-based IDMS approach. It employed an SMS in which the source was the maestro, and it considered a fixed receiver as the sync (master) reference during the session. Then, in [3], this proposal was extended so that the maestro could separately synchronize the playout processes of independent logical groups of distributed receivers (clusters). Moreover, several dynamic policies for selecting a master reference playout point in each cluster were adopted, and their suitability and effectiveness for specific network conditions and application requirements was examined.

In DCS (Fig.2c), all the receivers multicast feedback information about their playout timing and each one of them decides the sync reference among its own playout timing or those of the other receivers. This distributed control architecture has been adopted in several previous IDMS works, such as [5], [7]-[9].

3. Advantages/Disadvantages of an M/S Scheme

The above IDMS control schemes have their own advantages and disadvantages. In this Section, the main properties of an M/S Scheme in the context of IDMS are discussed and analyzed, taking into consideration some key factors such as robustness, reliability, scalability, traffic overhead, flexibility, location of control nodes, interactivity, consistency, fairness, coherence and security.

3.1 Advantages

M/S Scheme is the best end-based IDMS scheme in terms of *interactivity* because each slave destination can compute the detected playout asynchrony every time it receives a new control message 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. So, out of sync situations will be detected and corrected earlier using M/S Scheme than using DCS and SMS, thus minimizing the IDMS error.

In DCS and SMS, control messages (including playout timing info) are sent by all the receivers in a multicast and unicast way, respectively. If M/S Scheme is used, slave receivers will only receive control packets from the master destination and, therefore, the network load will not be significantly increased. So, the *traffic overhead* may be higher when using DCS than when using SMS (multicast vs unicast), and higher when using SMS than when using M/S Scheme (control messages sent from all the receivers vs only from the master receiver).

Closely related to the previous feature, SMS and DCS may present more stringent *scalability* constraints than M/S Scheme because in those schemes multiple destinations would send control packets almost simultaneously, which could originate a *feedback-implosion* problem because of the IDMS control. This problem may appear earlier in SMS than in DCS. This is because, despite the session members can be classified into different logical groups (or clusters), the sync server (maestro) must still gather the control messages from all the active destinations in the session. In DCS, instead, each destination must only process the feedback messages from those participants belonging to the same cluster with whom it is sharing a media experience.

To minimize such a problem, the report interval 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 to the *interactivity* and the frequency at which the 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 end-based IDMS control schemes.

In centralized schemes, since all the active destinations always receive the same playout control information from the maestro (in SMS) or from the master destination (in M/S Scheme), *inconsistency* between the states of session members occurs less likely. In contrast, in DCS there is no guarantee that the same output timing will be selected as the reference one in all the distributed destinations, leading to a probable inter-decision inconsistency.

Another major advantage of centralized schemes is that the presence of a server or a designated master node for the sync control makes cheating difficult, because the validity of the arriving control packets is easier to be monitored (*security*). In a distributed scheme, however, each peer takes its own decisions, without any control of what they are doing or if they are honest peers or not.

3.2 Disadvantages

Drawbacks of M/S Scheme also exist. Next, some of them are discussed. The first one of them is *robustness* (or *reliability*). In centralized schemes, if the sync manager (maestro, in SMS, or master node, in M/S) 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 no effect on the other participants since each participant is independent, and has locally all the necessary information to compute the overall sync status at any time.

Another important inconvenient of an M/S Scheme is *fairness*. It can be suitable for applications in which a single destination has priority 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') nodes the required playout adjustments. 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 participants (belonging to a specific cluster).

As a consequence, using M/S Scheme there is no option for selecting the reference output timing since it is taken from the one of the master destination (null *flexibility*). Conversely, SMS and DCS can employ several dynamic policies for selecting an IDMS reference from the all the collected output timings (as the ones proposed in [3]). Furthermore, the session members can be divided into independent sub-groups or clusters, which can be separately synchronized [3].

Unlike in DCS and SMS, in which the maximum playout asynchrony (the one between the most lagged and the most advanced receivers) can be estimated, in M/S Scheme each participant can only know the asynchrony between its local playout process and that of the master destination. Using M/S scheme, the reactive sync actions will not be performed simultaneously because slave receivers can only adjust their playout timing every time they detect an asynchrony value (regarding the master destination) exceeding an allowable threshold. Thus, despite the fact that M/S and SMS control schemes are the most appropriate in terms of *consistency*, SMS outperforms M/S Scheme and DCS in terms of *coherence*, which denotes the ability to synchronously (and simultaneously) coordinate the playout of media content in group shared experiences.

To conclude this section, the influence of the *location of the control node*s (the source and master destination) is examined. This issue is more relevant in centralized control schemes. Under heavily loaded network conditions, the IDMS error with SMS can be slightly larger than the one with M/S Scheme and DCS if the source is selected as the maestro. This is because the control packets sent by the maestro are (or can be) sent through the same path as the media data packets. Thus, some (data or control) packets may be dropped and, then, 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, 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, those control packets are less likely dropped on the intermediate links. Therefore, in such cases, M/S Scheme may achieve a higher IDMS quality than SMS. However, the most heavily loaded destination cannot be always known and, therefore, the master destination cannot be selected accordingly [10].

All the above properties have been summarized as strengths and weaknesses of an M/S Scheme for IDMS in Table 1. In particular, on the one hand, M/S Schemes are suited in those media sharing applications in which strict sync requirements, interactivity, low traffic overhead, scalability, consistency and security issues must be provided. On the other hand, M/S Scheme does not assure robustness, fairness and flexibility. Besides, the other end-based control schemes (SMS and DCS) can ensure better performance in terms of coherence, and are less affected to the location of the control nodes.

Table 1. Suitability of an M/S Scheme for IDMS.

M/S Scheme	
Strenghts	***Weaknesses***
Interactivity	Robustness
Traffic Overhead	Fairness
Scalability	Flexibility
Consistency	Coherence
Security	Location of Control Nodes

4. IDMS Challenges

Multimedia sync is a key issue in current content distribution networks and newer delivery paradigms (e.g. IMS-based TV broadcast channels), mainly due to the existence of several uncontrollable factors, some of which can be either related to the distribution network or to the end-systems' features, such as variable capturing, coding, encryption, packetization, network (e.g. background traffic load, trans-coding or format conversion, fragmentation and re-assembly of packets, dynamic routing strategies, improper queueing policies, 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 delays when multicasting one (or several) flows of information from one (or more) media sources to one (or multiple) destinations, possibly over different delivery chains (network architectures, cross-domain scenarios, etc.).

On the one hand, the experiments in [11] and [12] showed that delays of up to five seconds can occur in television and Internet/Web technologies when using different types of end-terminals. On the other hand, the studies in those works pointed that the allowable asynchrony levels (i.e. asynchrony limits that, if exceeded, can already be noticeable and annoying to users) can vary between few milliseconds to 1 second, depending on the specific use case in which IDMS is needed. Accordingly, it was concluded that existing distribution technologies do not handle the sync problem in an optimal way. The above delay differences become a serious constraint when interaction between the user and the media content, or between different users in the context of specific content, is needed, because it could be detrimental to the Quality of Experience (QoE) in those synchronous social media applications and may prevent the inclusion of such advanced forms of interactivity. Therefore, additional adaptive techniques must be provided to meet the above media sync requirements (especially IDMS) in practical content delivery networks.

5. Playout Rate Imperfections

Apart from a proper playout buffering policy, the availability of a precise and reliable local timing mechanism is a desirable feature to ensure both local and global media sync. A perfect receiver playout rate (μ MU/s) would mean the consumption of the incoming MUs with the same nominal rate (θ MU/s) as they were generated by the source ($\mu=\theta$). Nevertheless, real local playout timing mechanisms could present a deviation trend or skew given by $\pm\gamma$, that is often expressed as a ratio in parts per million (*ppm*), and also a nonlinear time variant drift, $\omega(t)$, which can randomly oscillate over time between values bounded by a maximum factor of $\pm\varepsilon$ *ppm*. This fluctuation is very close related to the clock resolution, aging, oscillator stability, voltage changes, surrounding temperature and other environmental variables (e.g., noise). As a result, the instantaneous playout rate of the *i-th* receiver can be formulated as:

$$\mu_i(t) = \theta\cdot(1+\gamma_i+\omega_i(t)) \quad \text{[MU/s]} \tag{1}$$

These undesirable factors, obviously, could affect to the receivers' playout asynchrony since their local timing are not generally in perfect agreement. If the receiver playout rate is faster than the sender generation rate, the receiver may suffer underflow as the session goes on. Conversely, the receiver can become flooded with MUs if its playout rate is slower than the generation rate of the sender.

6. RTCP-based IDMS using M/S Scheme

Most of the existing IDMS solutions ([1]) define new proprietary protocols, with their own specific control packets, with a consequent increase of the network load. Instead, we have decided to use standard protocols, such as RTP/RTCP (specified in [13]), and their extension for IDMS purposes.

Currently, RTP and RTCP protocols are extensively used in many communication and entertainment systems that involve streaming media, such as Voice over IP (VoIP), videoconferencing applications, Video on Demand (VoD), or TV services over IP (IPTV). On the one hand, RTP provides the following main features: *(i) payload type identification*, to allow applications to identify the format of the payload carried in the RTP media stream; *(ii) sequence numbers*, to allow RTP receivers to detect missing and out of order packets; and *(iii) timestamps*, to allow RTP senders and receivers to calculate delay jitter, synchronize their clocks, and reconstruct the original media timing. On the other hand, the data transport is augmented by a RTCP to allow QoS (e.g. jitter, delay, or loss rate) monitoring of the data delivery in a manner that is scalable to large multicast networks, and to provide minimal control and identification functionality. RTP receivers and senders provide reception quality feedback by sending out RTCP Receiver Report (RR) and Sender Report (SR) packets [13] respectively, which may be augmented by eXtended Reports (XR) [14]. Subsequently, service providers (e.g., IPTV) can trust on RTCP for troubleshooting and for fault tolerance management. For example, networked devices can connect to an IPTV distribution channel for retrieving available content on demand and, additionally, they could integrate other IP-enabled communication services such as VoIP, chat and video conferencing, thus enabling Social TV scenarios.

These protocols are useful for us because modifications and/or extensions to them (e.g. by defining new header extensions, report blocks or packet types) are allowed in order to include profile-specific information required by particular applications (RFC 3550 [13], RFC 3611 [14], and RFC 5968 [15]). As IDMS involves the distribution of feedback reports about receipt and playout timing, and this kind of information can be considered as a QoS metric (it can reflect the effect of jitter, network load, packet losses, clock skews/drifts, CPU overload, etc.), RTCP becomes a promising candidate for carrying out IDMS.

So, we decided to design a new IDMS approach taking advantage of the RTP/RTCP capabilities, by means of including/collecting playback timing in/from new defined and extended reports, which may facilitate implementation/deployment in typical multimedia applications.

Next, the main properties of our RTCP-based IDMS approach are outlined:

- The synchronization problem is tackled by dividing it in two main phases: i) *Coarse Sync,* to ensure that all the receivers start the playout process at the same time [2] and [3]; and ii) *Fine Sync,* to maintain the playout processes between all of them synchronously throughout the streaming session lifetime.

- It is based on the availability of a global time reference for all the participants involved in a sync session (e.g., provided by NTP, GPS, PTP or other solutions).

- The transmission rate for the feedback information does not need to be computed, as is required by most of the solutions in [1], because the report interval is dynamically adjusted according to the active number of participants and the session bandwidth, as specified in [13].

In this paper, our RTCP-based IDMS solution has been modified to adopt an M/S Scheme for the sync control. In our previous SMS-based IDMS approach ([2] and [3]) all the destinations sent (unicast) messages about playout timing to a centralized maestro. If the maestro, after processing such information, detected an out of sync situation, it sent (multicast) to all the destinations a new control message including playout setting instructions. In the new version of our IDMS approach presented in this paper, only a designated destination (master) will multicast feedback messages about its own playout timing and, after receiving that control report, all the (slave) receivers will directly calculate (apart from perform) the necessary the sync adjustments.

In this section, the main techniques that have been designed and adapted to acquire IDMS in the proposed M/S Scheme are described.

6.1 Fine Sync (Playout Timing Control)

During the session, receivers regularly send RTCP RR to inform about QoS [13]. Additionally, for IDMS purposes, a new RTCP XR block type (Fig.3a) has been adopted to provide feedback about receipt and presentation times for RTP packets in a specific *i-th* receiver: i) the original RTP timestamp[2] of the MUs being played (t'_i); ii) its reception instant (r_i); and iii) its playout time (p_i). This control message will be sent in a compound packet only by the designated master destination in the proposed M/S Scheme for IDMS.

Each slave receiver must register the playout timing information included in that XR message and, immediately, it must compute the playout time discrepancy (asynchrony) regarding the master destination, by comparing its current playout delay (d_{slave}) to the one reported by the master (d_{master})[3]. If the detected asynchrony (Δ), see Eq.2, exceeds an allowable threshold ($\Delta \geq \tau_{max}$) that slave receiver must adjust its playout process to get in sync. It can be done by following two possible methods. The first one is based on simple reactive actions such *'skips/pauses'* (aggressive adjustments), while the second one makes use of Adaptive Media Playout or AMP (smooth adjustments) to achieve IDMS.

$$\Delta = (d_{master} - d_{slave}) \qquad (2)$$

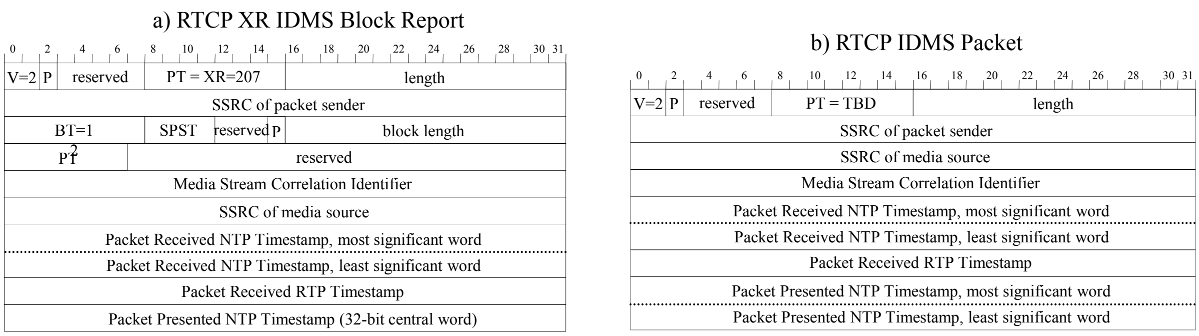

Figure 3. RTCP reports/packets for IDMS.

[2] Note that the notation of the RTP Timestamp (t') differs from the one of the transmission time (t) since the former refers to the RTP sampling (local clock) instant, while the later refers to the absolute reference (wall clock) time.
[3] For each *i-th* receiver its playout delay can be computed as $d_i = p_i - t'_i$.

Network Protocols and Algorithms
ISSN 1943-3581
2011, Vol. 3, No. 3

The timing diagram for the RTCP messages exchange in this centralized version of our RTCP-based IDMS approach is illustrated in Fig.4, whilst the flow chart of the designed M/S Scheme for IDMS (implemented in each slave receiver) is sketched in Fig.5.

6.2 Aggressive Playout Adjustment (Skips & Pauses)

If $\Delta>0$, that slave receiver is advanced. So, it must *'pause'* (stop playing) its playout process during Δ seconds to synchronize, causing a probable *freezing effect*. Consequently, its playout delay will be increased (i.e. the playout time for the next MU will be delayed). Otherwise, if $\Delta<0$, that slave receiver is lagged. In that case, it must *'skip'* (jump or move forward) a certain number of MUs until the detected asynchrony is reduced to a lower value than the service/presentation time for one MU. This way, its playout delay will be reduced (i.e. the playout time for the next MU will be advanced). Those reactive playout actions will result in an overall sync status (within acceptable limits), as shown in the Evaluation Section.

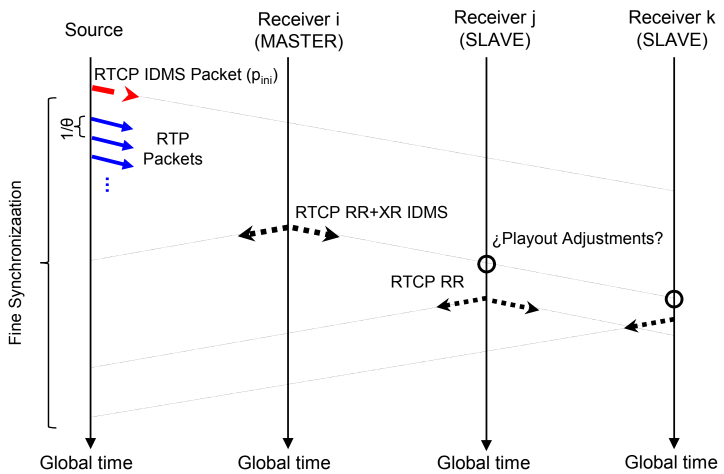

Figure 4. RTCP messages exchange for IDMS (M/S Scheme).

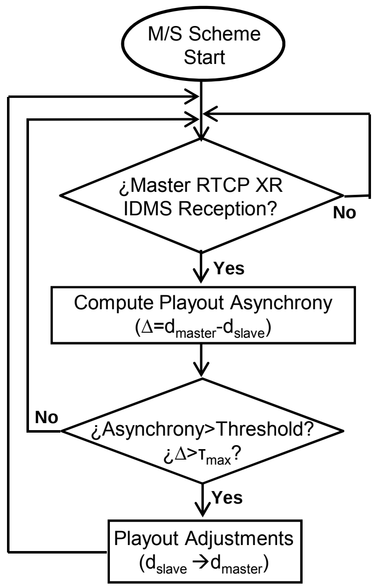

Figure 5. Flow Chart of the Designed M/S Scheme for IDMS.

6.3 Smooth Playout Adjustment (Adaptive Media Playout)

The above reactive playout adjustments could originate a noticeable degradation of user perception as regards the quality of the incoming media stream because, on the one hand, some important information may not be presented to the users (due to the *skipped* MUs) and, on the other hand, a sensation of loss of continuity may also be noticed (due to the *paused* MUs). Thus, an AMP technique has been adopted to minimize the occurrence of the above long-term playout discontinuities.

Previous works on AMP have been mostly focused to mitigate the effect of network (and end-systems) dynamics in order to improve the intra-stream sync quality (according to the buffer fullness level) of audio and video streaming applications (e.g. [17] and [18]) and, occasionally, for inter-stream sync purposes [19]. However, in [3], authors proposed to extend the use of AMP for IDMS purposes, since this technique can enable distributed receivers to adjust their playout timing (playing the media faster/slower than normal), within perceptually tolerable ranges, in order to smoothly acquire an overall sync status every time an asynchrony threshold between their playout states is exceeded. In this paper, the above AMP solution, which was designed to be used in an SMS-based IDMS approach, has been adapted to be applied in the newly proposed M/S Scheme for IDMS.

The operation of the AMP algorithm is as follows. Initially, the playout controller deals to play out the buffered MUs at a non-adaptive playout rate (μ), as they were generated by the source. Once each *n-th* MU finishes its presentation period, the next *(n+1)-th* MU must be played out, and the buffer occupancy must be updated. Every time an out of sync situation is detected by a slave receiver (after receiving an XR message from the master), the target playout point (d_{IDMS}=d_{master}) is registered and the AMP process is triggered. At this point, the AMP approach will attempt to either fasten or loosen the video playout rate in order to minimize Δ, by means of smoothly adjusting its local playout delay to the one of the IDMS master reference (d_{IDMS}).

However, to perform the AMP scheme we must consider the allowed ratio within which the video playout speed can be varied without degrading the media quality. Previous related subjective tests have shown that playout speed variations of up to 25% are often *unnoticeable* and, depending on the content and the frequency of the adjustments, variations up to 50% are sometimes *acceptable* ([17] and [18]). Thus, we assume in our simulation tests that video playout adjustments up to 25% lead to unnoticeable quality impairments, and we define a playout factor (φ) to specify this variation ratio.

On the one hand, high values (near the maximum limit, i.e. $|\varphi_{max}|\approx 0.25$) of the playout factor will result on a nearly immediate sync status (depending, of course, of the allowed τ_{max}). On the other hand, low values of the playout factor would origin unnoticeable rate adjustments, but the overall sync status will be reached later. In this work, for simplicity, a linear adjustment policy has been adopted. The number of MUs involved in the AMP period must be enough to allow slave receivers to correct the detected playout asynchrony ($\Delta\approx\tau_{max}$) without exceeding a pre-specified playout factor ($|\varphi|\leq 0.25$). That number is given by Eq.3 (if the slave receiver is advanced) or by Eq.4 (if the slave receiver is lagged):

$$N_{\varphi}^{advanced} \geq \left\lceil \frac{\tau_{\max}}{\frac{1}{\mu\cdot(1-\varphi)} - \frac{1}{\mu}} \right\rceil \quad (3)$$

$$N_{\varphi}^{lagged} \geq \left\lceil \frac{\tau_{\max}}{\frac{1}{\mu} - \frac{1}{\mu\cdot(1+\varphi)}} \right\rceil \quad (4)$$

Such process is illustrated in Fig.6. The AMP process will be finished once the target playout point (d_{IDMS}) is reached, i.e. once the playout delay of the slave receiver matches with the one of master receiver (d_{slave}=d_{master}), and will not be performed again until a new out of sync situation is detected.

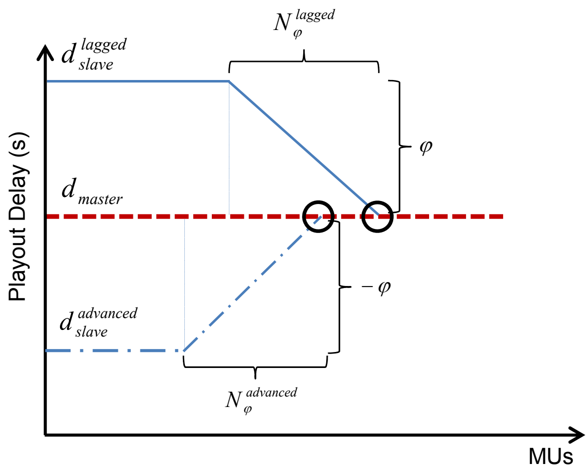

Figure 6. AMP Adjustments.

7. Evaluation

Modelling and simulations were conducted using NS-2. We tested the proposed M/S Scheme for IDMS in a multicast scenario with four distributed receivers, with variable delays to the media server (Table 2). The source transmitted a media stream with a specific rate of θ=25 MU/s. Moreover, different intensive background traffic was configured over the network topology in order to cause jitter variability. We set larger receivers' playout rate deviations (Table 2) than reasonable values in inexpensive oscillators, which can vary between 10-100 ppm [20], in order to force higher asynchronies among themselves, and to test if they were successfully handled by the our IDMS approach when using the proposed M/S Scheme.

The duration of the simulations was set to 10 minutes. As in an M/S Scheme, each receiver can only estimate the playout asynchrony regarding the master destination and, according to the studies in [11], an overall asynchrony of 100 ms can already be perceivable and annoying in some use cases in which IDMS is needed, the value of τ_{max} (maximum allowable asynchrony threshold) was set to 50 ms. This way, the overall playout asynchrony should always be kept below 100 ms, since the worst case would occur when the playout point of two specific receivers are extremely lagged and advanced, respectively, regarding the one of the master destination (without considering possible extreme congestion situations).

Table 2. Receivers' Parameters.

Receiver	*Mean RTT (ms)*	*Rate Skew (%)*	*Rate Drift (%)*
R1 (LAN 1)	~10 ms	γ_1 = 0.03 %	ε_1=0.02 %
R2 (LAN 2)	~125 ms	γ_2 = -0.02 %	ε_2=0.02 %
R3 (LAN 3)	~288 ms	γ_3 = -0.05 %	ε_3=0.02 %
R4 (LAN 4)	~88 ms	γ_4 = 0 %	ε_4=0 %

7.1 Simulation Results

Figure 7a illustrates the playout delay evolution of 3 receivers (R1, R2, and R3) to acquire IDMS when our RTCP-based solution was enabled, using M/S Scheme (as sync control scheme) and aggressive adjustments (as a sync reactive technique). It can be observed that the asynchrony between the playout states of those receivers progressively increased mainly due to the configured deviations in their local playout rates (Table 2). In such a case, R1 (the fastest one, $d_{master}=d_1 \leq d_2 \leq d_3$) was selected as the IDMS master reference during the session lifetime. As a result, every time slave receivers (R2 and R3) detected an asynchrony between their local playout point and that of the master destination (R1) exceeding the allowed threshold (τ_{max}=*50 ms*), they adjusted their playout timing, by skipping just one MU[4] (see zoom view), to get in sync. However, there were not any *'pauses'* in the receivers' playout processes during the session. Unlike in our previous SMS-based IDMS approach ([2] and [3], in which the playout adjustments where performed almost simultaneously as a result of a reception of a new control message (including playout setting instructions) from the maestro, in the proposed M/S Scheme, the playout adjustments are directly calculated (and performed) by slave receivers every time they detect an out of sync situation. Therefore, those adjustments would not be performed simultaneously by all the slave receivers (*coherence*), as can be appreciated in the figure.

Despite the different Round Trip Time (RTT) values for each receiver, measured from each RTCP RR sent by them during the simulations (Table 2), we can notice in all the graphs in Fig.7 that all of them were perfectly synchronized at p_{ini} (it is emphasized in the zoom view of Fig.7a), because the source indicated to them a global initial playout delay (d_{ini}) of 500 ms in the initial phase of the IDMS approach (*Coarse Sync*) [2].

Figure 7c illustrates the same process, but using AMP as reactive technique. In such a case, slave receivers (R2 and R3) were more closely and fine-grained synchronized than using aggressive adjustments because they smoothly fasted up their playout rate to minimize the detected asynchrony. Thus, the number of adjustments to acquire IDMS during the session was reduced for both lagged receivers.

[4] Only entire MUs can be skipped, each one with a duration of $1/\theta$=40 ms.

We can also notice in both figures a significant reduction in the playout delay in all the receivers as the session advanced in time, which caused an inherent progressive emptying of their buffer occupancy. Thus, if the playout rate of the master is significantly faster than the source nominal rate, the playout buffers may suffer underflow if the multimedia session had a long duration, and it would require the use of novel adaptive techniques to avoid such a situation, without needing to significantly increase the p_{ini}.

Figures 7b and 7d show the evolution of the playout processes of the above receivers when R2 was selected as the IDMS reference, using aggressive and smooth adjustments, respectively. In such a case, lagged slave receivers (R1) had to fast up their media playout rate to synchronize every time τ_{max} was exceeded. Conversely, advanced slave receivers (R3) had to slow down their playout rate to minimize the detected asynchrony. As in the previous case, lagged receivers were more closely synchronized when using AMP.

Figure 7e illustrates the same process when R3 (the slowest one) was selected as the master. In such a case, the evolution of the playout processes was similar when using aggressive and smooth adjustments, and the graphs for both cases were undistinguishable (unless a zoom view in the adjustment periods is done). This is because advanced receivers paused specific MU the exact value of to the detected asynchrony (Δ) in order to minimize it. Using M/S Scheme, if the master receiver is significantly slower than the multimedia source, the playout buffers may progressively become flooded with MUs (leading to probable overflow situations) as the multimedia session goes on (it can be observed in Fig.7e a progressive increase of the playout delay in all the receivers). So, large capacities of the playout buffers would be required, with the probable degradation of the real-time perception.

Generally, using smooth adjustments, long-term playout discontinuities (skips/pauses) were avoided, although the total number of adjusted MUs was higher than the total number of *skipped/paused* MUs when aggressive adjustments were employed. However, the percentage of adjusted MUs involved in the AMP in each one of the simulated cases was very low, so those adjustments might be unnoticeable by users and also they added a minimal computational load to the receivers.

To conclude the evaluation, a new ideal receiver (R4) with a perfect playout rate (without deviations) was joined to the multicast session (see Table 2). The playout controller of that receiver was able to play out the incoming MUs with the same rate as they were generated by the multimedia source. As a result, we measured the buffer delay for the incoming MUs in that receiver when each one of the receivers was selected as the master. We can observe in Fig.7f that the buffer delay for that receiver (R4) was kept quite uniform during the multimedia session (the appreciated fluctuation is due to the jitter delays) until new playout adjustments were triggered. At this moment, R4 had to adjust its playout timing in order to correct the estimated playout time discrepancy, according to the collected playout point of the master receiver. As expected, when R2 and R3 (slow receivers) were selected as the IDMS master references, the buffer delay for the incoming MUs was progressively increasing. This means an inherent filling of the playout buffer that could even overflow if the multimedia session had a long duration. On the contrary, when R1 (the fastest one) was selected as the master, the buffer delay in R4 was progressively decreasing during the session lifetime, because the playout rate of the master receiver was higher than the source nominal rate ($\gamma_{master}=\gamma_1>0$). As a consequence, the buffer occupancy was gradually emptying. Finally, when R4 was selected as the master reference for IDMS, we can observe a smooth evolution of its buffering delay because it presented an ideal playout timing. So, the buffer fullness level was moderately stable during the multimedia session.

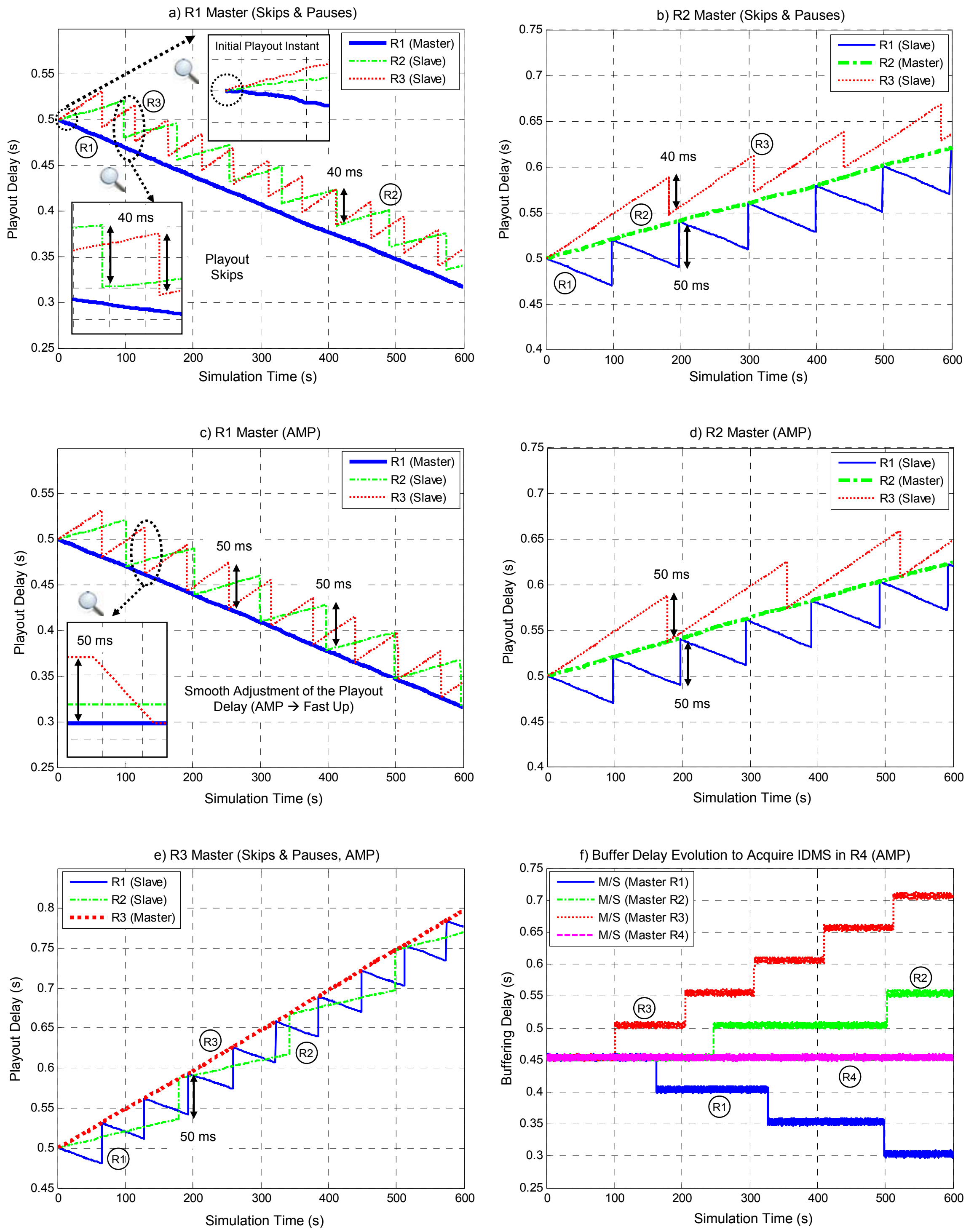

Figure 7. Playout Delay Evolution to Acquire IDMS (M/S Scheme).

So, it can be concluded that using M/S Scheme, if the master is advanced/delayed, the playout delay for all the receivers will progressively increase/decrease as the session goes on, with the consequent progressive filling/emptying of the playout buffers that could even overflow/underflow (if no additional control is used).

Regarding the traffic overhead added by our RTCP-based IDMS approach using M/S Scheme, the number of RTCP XR reports (288 bits each one) sent by the master receiver during the 10-minute session in each one of the simulations was around 2% of the total RTP data packets sent by the server (~15000). These amounts are similar to our measurements in [2]. Additionally, the source must send an RTCP IDMS (224 bits) to coarsely synchronize the playout processes of all the distributed receivers at the beginning of each stage in which the session can be divided (one single stage in our simulations). Consequently, the traffic overhead is very low, because we have not defined a new proprietary protocol either, but we have taken advantage of the RTP/RTCP feedback and extension capabilities for designing an M/S Scheme for IDMS. Also note that, according to [13], the maximum fraction of the total amount of control traffic added by RTCP must be limited to 5 % of the RTP session bandwidth, so the control traffic overhead added by our IDMS approach will be always significantly lower than this percentage.

8. Conclusion

In this paper, our RTCP-based IDMS approach has been extended and adapted to integrate an M/S Scheme. The main strengths and weaknesses of that control scheme for IDMS has been qualitatively discussed and analyzed. Simulation results have proved the effectiveness of that M/S Scheme to keep an overall sync status (consistency/coherence) in a group shared media experience, and its suitability for those applications in which high performance in terms of scalability, control traffic overhead and, especially, interactivity must be provided. Also, an M/S Scheme can be suited in those scenarios in which strict sync levels must be ensured and one of the participants may have priority over the others (for example, synchronous e-learning applications, in which the teacher's node could be the master).

As future work, we plan to: i) perform a more exhaustive comparison between the existing (end-based and network-based) IDMS schemes; and ii) implement our RTCP-based IDMS approach in actual video sharing applications in order to perform real-world trials, measurements and exhaustive subjective tests.

Acknowledgement

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.

References

[1] Boronat F., Lloret J., and García M., "Multimedia group and inter-stream synchronization techniques: A comparative study", *Information Systems*, Vol.34, 1, 108-131, March 2009, http://dx.doi.org/10.1016/j.is.2008.05.001.

[2] Boronat F., Guerri J.C., and Lloret J., "An RTP/RTCP based approach for multimedia group and inter-stream synchronization", *Multimedia Tools and Applications Journal*, Vol. 40 (2), 285-319, June 2008, http://dx.doi.org/10.1007/s11042-008-0208-1.

[3] Boronat F., Montagud M., Vidal V., "Smooth Control of Adaptive Media Playout to Acquire IDMS in Cluster-based Applications", *IEEE LCN'11*, 617-625, Bonn, October 2011.

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

[5] Vaishnavi I., Cesar P., Bulterman D., Friedrich O., Gunkel S., Geerts D., "From IPTV to synchronous shared experiences challenges in design: Distributed media synchronization", *Signal Processing: Image Communication*, Vol. 26, Issue 7, 370-377, August 2011, 006 http://dx.doi.org/10.1016/j.image.2011.01.006.

[6] Geerts D., Vaishnavi I., Mekuria R., Van Deventer M. O., Cesar P., "Are we in sync?: synchronization requirements for watching online video together", *CHI '11*, New York (USA), May 2011, http://doi.acm.org/10.1145/1978942.1978986.

[7] Diot C., Gautier L., "A Distributed Architecture for Multiplayer Interactive Applications on the Internet", *IEEE Network*, vol. 13, nº 4, pp. 6-15, July/August 1999, http://doi.acm.org/10.1109/65.777437.

[8] Mauve M., Vogel J., Hilt V., Effelsberg W., "Local-Lag and Timewarp: Providing Consistency for Replicated Continuous Applications", *IEEE Transactions on Multimedia*, Vol.6, No.1, February 2004, http://doi.acm.org/10.1109/TMM.2003.819751.

[9] Hosoya K., Ishibashi. Y, Sugawara S., Psannis K.E., "Group synchronization control considering difference of conversation roles", *IEEE 13th International Symposium on Consumer Electronics*, ISCE '09, 948-952, May 2009, doi: http://dx.doi.org/10.1109/ISCE.2009.5156876.

[10] Nunome T., Tasaka S., "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, http://dx.doi.org/10.1109/PIMRC.2005.1651663.

[11] Van Deventer M. O., Stokking H., Niamut O. A., Walraven F. A., Klos V. B., "Advanced Interactive Television Service Require Synchronization", *IWSSIP 2008*, Bratislava, June 2008.

[12] Stokking H.M., Van Deventer M.O., Niamut O.A., Walraven F.A., Mekuria R.N., "IPTV inter-destination synchronization: A network-based approach", *ICIN'2010* , Berlin, October 2010, http://doi.acm.org/10.1109/IWSSIP.2008.4604379.

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

[14] Friedman T., Caceres R., Clark A., "RTP Control Protocol Extended Report (XR)", RFC 3611, November 2003.

[15] Ott J., Perkins C., "Guidelines on Extending the RTP Control Protocol (RTCP)", RFC

5968, September 2010.

[16] Brandenburg R. V., Stokking H., V. Deventer M.O., Boronat F., Montagud M., Gross K., "RTCP for Inter-Destination Media Synchronization", IETF Internet Draft, October 2011.

[17] Su Y., Yang Y., Lu M., Chen H., "Smooth Control of Adaptive Media Playout for Video Streaming", *IEEE Transactions on Multimedia*, Vol.1 (7), November 2009, http://doi.acm.org/10.1109/TMM.2009.2030543.

[18] Chuang H., Huang C., Chiang T., "Content-Aware Adaptive Media Playout Controls for Wireless Video Streaming", *IEEE Tran. on Multimedia*, Vol.9 (6), 2007, doi: 10.1109/TMM.2007.902884.

[19] Ishibashi Y., Tasaka S., Ogawa H., "Media Synchronization Quality of Reactive Control Schemes", *IEICE Transactions on Communications*, Vol.E86-B, 10, 3103-3113, October 2003.

[20] Ferrari F., Meier A., Thiele L., "Accurate Clock Models for Simulating Wireless Sensor Networks", *SIMUTools 2010*, Torremolinos (Spain), March 2010, http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8693.