On the use of Adaptive Media Playout for Inter-Destination Synchronization

Mario Montagud, *Student Member, IEEE,* and Fernando Boronat, *Member, IEEE*

***Abstract*—Inter-Destination Multimedia Synchronization (IDMS) is essential in most of the emerging social multimedia applications. In this paper we present a novel Adaptive Media Playout (AMP) scheme that aims to acquire an overall synchronization status between distributed receivers by means of smoothly adjusting their playout timing. Simulation results show that the proposed solution minimizes long-term playout discontinuities (skips and/or pauses) which are subjectively more annoying to users than small variations in the media playout rate.**

***Index Terms*—Adaptive Media Playout, Inter-Destination Synchronization, Multimedia Systems, RTP/RTCP, Simulation.**

I. Introduction

DISTRIBUTED multimedia applications usually include intra-stream and inter-stream synchronization solutions. Nevertheless, a new type of synchronization is essential in a variety of emerging multimedia applications, such as Social TV or multi-playing online games. It is called Inter-Destination Multimedia Synchronization (IDMS), and involves the simultaneous synchronization of the playout processes at different receivers ([1] - [4]).

The maintenance of the temporal relationships within or among different media streams may be disturbed by the following factors: *delay, jitter, loss rate, clock skews/drifts, etc*. Thus, streaming services usually rely on playout buffers to protect themselves from feasible distortion of the original media timing. Let t_n, $r_{n,i}$ and $p_{n,i}$, be the time when the *n-th* Media Unit (MU) is sent, received, and played by the *i-th* receiver, respectively. We call *Initial Playout Instant* ($p_{ini,i}$) to the playout time of the first MU (MU_{ini}) for each *i-th* receiver. Next, the playout controller will schedule the playout of the successive MUs at $p_{n+1,i}=p_{n,i}+s_{n,i}$, where $s_{n,i}$ refers to the service time for the *n-th* MU (if we assume a CBR traffic of θ MU/s, $s_{n,i}$ should be equal to $1/\theta$ seconds). The *playout delay* for the *n-th* MU is given by $d_{n,i}=p_{n,i}-t_n$ and it should be maintained uniformly for each couple of *n-th* and *k-th* MUs, i.e. $(p_{n,i}-p_{k,i})\approx(t_n-t_k)$. However, buffering policies do not suffice in severely congested environments, where playout interruptions could occur due to underflow/overflow situations. Hence, Adaptive Media Playout (AMP) solutions have emerged to smoothly regulate the inter-presentation time among MUs, while reducing long-term playout discontinuities (skips/pauses), which are subjectively more annoying to user perception on the media quality than short-term discontinuities (small variations in the playout rate) [5].

Additionally, distributed receivers could lose synchronization due to possible imperfections in their local playout rates that could present a deviation trend or skew ($\pm\gamma$ parts per million *-ppm-*) and also a nonlinear time variant fluctuation or drift ($\omega(t)$, which is typically bounded by a maximum value of ε *ppm*) over the sender nominal rate [6]. As a result, the instantaneous playout rate (in MU/s) of the *i-th* receiver can be formulated as: $\mu_i(t)=\theta\cdot(1+\gamma_i+\omega_i(t))$ (1).

The above factors would cause an increasing asynchrony between the receivers' playout states that must be controlled and corrected if IDMS must be satisfied. However, the feasible playout adjustments performed by each receiver to synchronize could occasion a noticeable degradation of the media quality. Thus, in this work we present the coordination of an enhanced IDMS approach and a novel AMP scheme that aims to alleviate the effect of playout interruptions while acquiring an overall synchronization status between distributed receivers by means of smoothly adjusting their playout timing.

II. IDMS Approach

In [1], authors presented a qualitative comparison of many IDMS proposals. Most of them defined new proprietary protocols. Instead, our IDMS approach ([2]) is based on simple extensions to RTP/RTCP [7], following the guidelines specified in [8], which may facilitate implementation and deployment in current multimedia services [4]. It tackles the synchronization goal by dividing it into two main phases: initial phase, to ensure that all the receivers initiate the play out of the media stream at the same time (p_{ini}); and second phase, to maintain the playout processes between distributed receivers in a synchronized way throughout the streaming session lifetime. Further details can be found in [2]. We extended RTCP RR to include the local playout point of each *i-th* receiver (i.e., the MU being played *-MU_i-* and its playout time *-p_i-*). Once the source has collected the playout information of all the active receivers, it will run a simple algorithm so as to obtain a synchronization reference and calculate the maximum playout time discrepancy between all of them. If the detected asynchrony exceeds an allowed threshold (τ_{max}), the source will multicast a new defined *'action message'*, called RTCP APP ACT, which includes a target playout point (i.e., a MU sequence number *-MU_{ACT}-* and

This work was supported by Universidad Politècnica de València (UPV), under its R&D Support Program in PAID-01-10 Project, and by Generalitat Valenciana, under its R&D Support Program in GV 2010/009 Project.

The authors are with IGIC Institute and UPV, 46730, Grao de Gandia (Spain) (e-mail: mamontor@ posgrado.upv.es, fboronat@dcom.upv.es).

the global time -p_{ACT}- at which that MU should be played), to make the receivers adjust their playout timing. We have extended our IDMS approach [2], in which a fixed receiver was selected as the synchronization reference, by introducing four new dynamic master selection policies: i) synchronization to the slowest receiver; ii) synchronization to the fastest receiver; iii) synchronization to the mean playout state; and iv) synchronization to the source nominal rate. Immediately after receiving an *action message*, the playout controller in each receiver will deduce its state by comparing the target playout point with the local one. Consequently, it will adjust its playout process 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 AMP to achieve synchronization (smooth adjustments).

III. AMP Scheme

Let us consider the *i-th* receiver is playing a specific MU -MU_i- at p_i instant (local playout point). That receiver would consume the successive MUs with a (possibly deviated) playout rate of μ_i MU/s. So, it would play out the MU_{ACT} at p'_{ACT} instant which (possibly) does not match with p_{ACT} instant (synchronization target). Let $\Delta_{n,i}$ denote the asynchrony, for each *n-th* MU, between the evolution of the local playout point of the *i-th* receiver and the target playout point:

$$p_{ACT} = p'_{ACT} + \Delta_{n,i} = \left[p_i + \frac{1}{\mu_i}(MU_{ACT} - MU_i) \right] + \Delta_{n,i} \qquad (2)$$

If $\Delta_{n,i}>0$, the *i-th* receiver playout process is advanced to the synchronization target. So, it must *'pause'* its playout process during $\Delta_{n,i}$ seconds to synchronize. Otherwise, if $\Delta_{n,i}<0$, the receiver playout process is lagged. In that case, it must *'skip'* a certain number of MUs to minimize the detected asynchrony.

Our proposed AMP scheme aims to minimize the above long-term playout discontinuities. The flow chart of this algorithm is sketched in Fig. 1. Initially, the playout controller manages the play out of the buffered MUs at a non-adaptive rate given by $\mu_{n,i}=1/(s_{n,i})$. Each receiver includes its current local playout point (MU_i, p_i) in each RTCP RR EXT it sends to allow the source to gather the overall playout status. If asynchrony greater than τ_{max} is detected by the source, a new RTCP APP ACT will be sent. Once it is received by participants, the target playout point is registered and the AMP process is triggered. At this point, the AMP approach will attempt to either fasten or loosen the playout rate in order to distribute $\Delta_{n,i}$ among all the remaining MUs to reach the target playout point. It can be done by means of increasing or decreasing the playout time of each *n-th* MU a value of $\delta_{n,i}=(\Delta_{n,i})/(MU_{ACT}–MU_i)$ seconds (i.e. $s_{n,i}+\delta_{n,i}$). However, to perform the AMP, we must consider the allowed ratio within which the playout speed can be manipulated without degrading the user perception on the media quality. Subjective tests have shown that playout speed variations of up to 25% are often *unnoticeable* and, depending on the content, variations up to 50% are sometimes *acceptable* ([5] and [9]). Thus, we will assume that playout adjustments up to 25% lead unnoticeable quality impairments, and we define a playout factor for each *n-th* MU ($\varphi_{n,i}$) to specify this variation ratio, whose optimal value (4) is computed, combining (2) and (3) as:

$$p_{ACT} = p_i + \frac{1}{\mu_i(1+\varphi_{n,i})}(MU_{ACT} - MU_i) \ (3); \quad \varphi_{n,i} = \frac{1}{1+(\delta_{n,i}/s_{n,i})} \ (4)$$

Note that if the calculated $\varphi_{n,i}$ is higher than 25%, it will be limited to that maximum scaling ratio (i.e. $|\varphi_{max}|\leq 0.25$). In such cases, the receiver could not achieve a fine synchronization. It may occur when τ_{max} is set too high or when there are not enough buffered MUs to smoothly distribute the detected asynchrony between them. So, a proper election of p_{ini}, τ_{max}, MU_{ACT} or the master selection algorithm must be accomplished to minimize noticeable playout interruptions (skips/pauses or buffer outage). The AMP process will be finished once the target playout point is reached (i.e. MU_{ACT}, $\varphi_{n,i}$ and $\delta_{n,i}$ will be set to zero) and will not be performed again until the reception of a new RTCP APP ACT packet.

IV. Experimental Results

A. Simulation Models, Scenario and Setup

Modeling and simulations were conducted using NS-2. We tested our approach in multicast scenario with three distributed receivers (Table 1). The multimedia stream consisted of a CBR traffic with a specific rate of 200 kb/s (θ=25 MU/s). We additionally configured background traffic over the network topology in order to cause jitter variability. The receivers' playout parameters were configured as summarized in Table 1. Typically, the requirements on inter-destination content synchronicity may vary between 15 and 500 ms, depending on the type of multimedia service offered. In most applications, differences around 100 ms may already have an annoyance effect [3], so τ_{max} was set to a lower value of 80 ms.

B. Simulation Results

Despite the different Round Trip Time (RTT) values for each receiver, measured from each RTCP RR sent by them (Table 1), we can notice from the upper graphs in Fig. 2 how all the receivers were perfectly synchronized at the p_{ini}.

Fig. 2 illustrates the playout delay evolution in all the receivers when the fastest receiver was selected as the synchronization reference. In that case, every time τ_{max} was exceeded, the source sent an *action message* to make the receivers adjust their playout timing according to the collected playout information of the fastest one (R1). We can observe from the upper graph how, using aggressive adjustments, slower (slave) receivers had to skip a certain number of MUs to synchronize. The summary of the reactive playout adjustments in all the receivers, for each one of the adopted master selection policies, is summarized in Table 2.

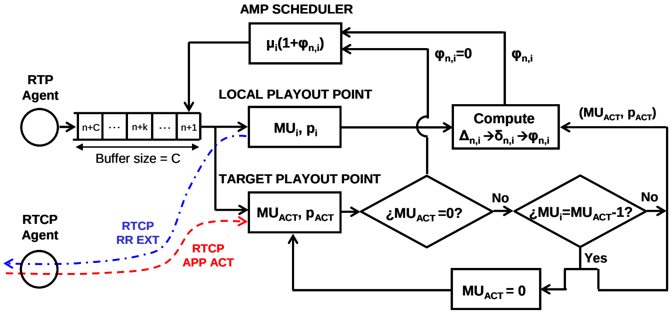

Fig. 1. AMP Scheme

The lower graph illustrates the same process by enabling the AMP mechanism. It can be observed how lagged receivers (R2 and R3) were more closely and fine-grained synchronized. Thus, the number of *action messages* sent by the source was reduced. Generally, using smooth adjustments, long-term discontinuities were avoided, although the total number of adjusted MUs was greater than the total number of skipped/paused MUs when aggressive adjustments were employed. However, in none of the master selection algorithms the percentage of adjusted MUs was higher than 1% of the total MUs (fifth column in Table 2). The playout rate deviations of R2 and R3 were intentionally changed at the midpoint of the simulation in order to convert R2 as the slowest one, so as to reflect M/S switching capabilities [1]. The upper graph in Fig. 3 illustrates the playout delay evolution when the synchronization to the source nominal rate was selected. This policy was introduced in order to avoid underflow/overflow situations during the multimedia session (if network conditions are quite stable). We can observe from this graph how the playout delay evolution was kept quite uniform for all the receivers during the session (the buffer occupancy variation was bounded to a lower value than $\pm\tau_{max}$), which is a desired feature in real-time multimedia services. In such a case, accurate receivers will not have to make significant adjustments in their playout timing. In addition, the lower graph in Fig. 2 illustrates the playout rate variation for all the receivers when this policy was employed. This figure corroborates that the playout rate was varied within perceptually tolerable ranges in order to acquire IDMS. The last column in Table 2 reflects the maximum playout factor for all the adopted master selection policies ($|\varphi_{max}|\leq 0.25$).

V. Future Work

Our future work will address the following issues: i) a more exhaustive and a subjective assessment of the proposed AMP scheme; and ii) design of a dynamic AMP scheme to provide both buffer fullness level monitoring and IDMS control.

References

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

[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.

[3] V. 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.

[4] ETSI TISPAN, "IMS-based IPTV stage 3 specification", TS 183 063 v3.4.6 (2010-12).

[5] Su Y., Yang Y., Lu M., Chen H., Smooth Control of Adaptive Media Playout for Video Streaming, IEEE Transactions on Multimedia, Vol.1, No. 7, November 2009.

[6] Ridoux J. and Veitch. R., "Principles of Robust Timing over the Internet", Communications of the ACM, Vol. 53 (No. 5), May 2010

[7] Schulzrinne H., Casner S., Frederick R., and V. Jacobson, "RTP: a transport protocol for real-time applications", RFC-3550, July 2003.

[8] Ott J., Perkins C., Guidelines on Extending the RTP Control Protocol (RTCP), RFC 5968, September 2010.

[9] Chuang H., Huang C., and Chiang T., Content-Aware Adaptive Media Playout Controls for Wireless Video Streaming, IEEE Transactions on Multimedia, Vol.9, No. 6, October 2007.

TABLE I
Receivers' Parameters

Receiver	Mean RTT	Rate Skew (%)	Rate Drift (%)
R1 (LAN 1)	~44 ms	γ_1 = 0.05 %	ε_1 = 0.03 %
R2 (LAN 2)	~125 ms	γ_2 = -0.03 % → -0.07 %	ε_2 = 0.03 %
R3 (LAN 3)	~208 ms	γ_3 = -0.1 % → -0.05 %	ε_3 = 0.03 %

TABLE II
Playout Adjustments

Receiver		Aggressive Adjustment		AMP	
	Master Selection Algorithm	- Skipped (%) / + Paused (Δ_{max}) MUs	Buffer Fullness Variation	Adjusted MUs (%)	φ_{max}
R1	Fastest	0 / 0	-151.2 ms	-	-
	Slowest	0 / +4 (82.2 ms)	+177.8 ms	61(0.9%)	-0.16
	Mean	0 / +5 (54.7 ms)	+78.1 ms	45(0.7%)	-0.09
	Source	0 / +5 (23.7 ms)	$\leq\lvert\tau_{max}\rvert$ ms	43(0.6%)	-0.08
R2	Fastest	- 7 (0.1%) / 0	-131.3 ms	57(0.9%)	+0.23
	Slowest	0 / +3 (29.1 ms)	+236.8 ms	64 (1%)	-0.08
	Mean	0 / +1 (10.9 ms)	+159.5 ms	47(0.7%)	+0.06
	Source	- 3 (0.04%) / 0	$\leq\lvert\tau_{max}\rvert$ ms	45(0.7%)	+0.11
R3	Fastest	- 8 (0.1%) / 0	-107.7 ms	52(0.8%)	+0.24
	Slowest	0 / +1 (14.5 ms)	+235 ms	62(0.95%)	-0.03
	Mean	- 2 (0.02%) / 0	+139.2 ms	45(0.7%)	+0.1
	Source	- 4 (0.05%) / 0	$\leq\lvert\tau_{max}\rvert$ ms	43(0.6%)	+0.11

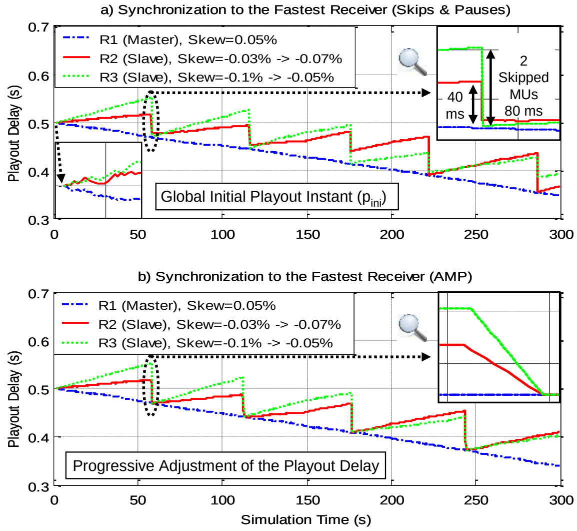

Fig. 2. Playout Delay Evolution to acquire IDMS.

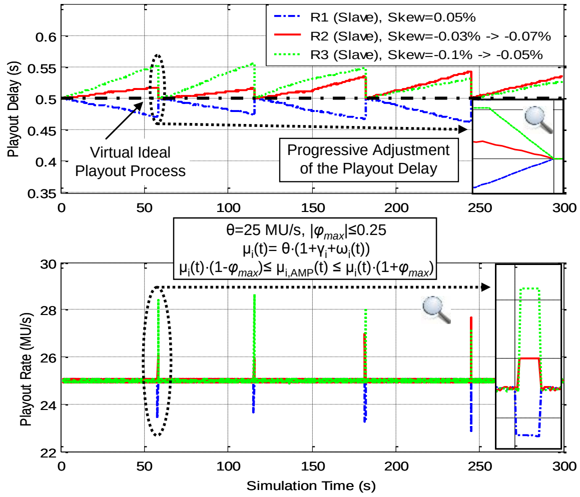

Fig. 3. Playout Rate Variation to acquire IDMS.