Improving Clock Synchronization for MPEG-2 Services over ATM Networks

R. Noro and J.P. Hubaux

Telecommunications Services Group, TCOM Laboratory
Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland
e-mail:{noro, hubaux}@tcom.epfl.ch
URL: http://tcomwww.epfl.ch/

Abstract. A stable clock indication is needed for the processing and playout of MPEG-2 streams. ATM networks can support the transfer of such streams but contribute significantly (as well as other system operations) to delay and delay variation. The MPEG-2 standard specifies strong temporal constraints for processing and presentation. MPEG-2 is also based on the assumption of a common time base shared among sender and receiver. Clock synchronization covers this requirement and is traditionally performed using PLLs. Unfortunately, their linear nature cannot simultaneously guarantee rapidity and good clock reconstruction. The non-linear technique we propose in this paper is able to offer such guarantees and also can handle higher amounts of delay variations. Such technique is not restricted to the scope of MPEG-2 clock synchronization and can also succeed in the reconstruction of time bases for generic distributed applications. It does not require any specific device, like very precise oscillators or filters.

1 Introduction

The MPEG-2 standard is becoming one of the most popular standards for the storage and transmission of high-quality audiovisual material. The standard specifies an audio and a video coding layer as well as a system layer, which defines the syntax and operations for transferring MPEG-2 streams over B-ISDN networks.

MPEG-2 allows several rate control methods for video: the more common are the *Constant Bit Rate* (CBR) control and the *Variable Bit Rate* (VBR) control. The latter is gaining more and more attention from the Multimedia Network community and has several advantages over CBR: reduced end-to-end transfer delay, better quality of video and simpler buffer management in the encoder/decoder. Nevertheless VBR traffic suffers a higher *Cell Loss Ratio* (CLR) and *Cell Delay Variation* (CDV).

MPEG-2 integrates a timing model that assures the synchronized presentation of audio/video streams and prevents buffer underflows/overflows, providing that sender and receiver share a common time base. This is obtained through a clock synchronization mechanism that locks the receiver's clock (slave) on the sender's one (master): the clock synchronization can fail if significant delay variations are introduced by the network and/or protocol's layers. In this paper we propose a non-linear clock synchronization method that is able to handle strong delay variations and does not require particular hardware devices at the receiver side.

The paper is organized as follows. In Section 2 we give a brief overview of MPEG-2 and its timing model, in Section 3 we describe how MPEG-2 streams map onto ATM networks and in Section 4 we explain the traditional clock synchronization approach. In Section 5 our non-linear technique for clock synchronization is discussed and a comparison with classical techniques is proposed. Finally, in Section 6 we derive some conclusions and further developments for the proposed method.

2 MPEG-2 timing model

The MPEG-2 standard is subdivided into three main parts: audio [1], video [2] and system [3]. Each audio/video stream is independently coded and an *Elementary Stream* (ES) is produced for each medium. Private data can also be streamed with independent syntax.

At the higher level, the *Logical Units* [4] of information are defined as *Presentation Units* (PUs) and correspond to *Access Units* (AUs) in the compressed domain. Due to the predictive nature of MPEG-2, equally sized PUs may generate varying size and differently ordered AUs. The ES are then organized in a sequence of *Packetized Elementary Stream* (PES) packets. Each of these packets is composed by an *header* and a *payload*, filled by contiguous bytes of an ES. A PES packet contains one or more complete AUs and its length is usually variable.

For the transmission over error-prone supports, such as ATM networks, the PES packets are further split into 188-byte, fixed length *Transport Stream* (TS) packets. These packets are suitable for the multiplexing of several media streams into a single multiplexed stream. The TS multiplexer labels the packets belonging to the same elementary stream with a unique *Program Identifier* (PID) field. The density of the packets with a given PID in the multiplexed stream is medium-dependent: for example, video packets can be six times more dense than corresponding audio packets. Fig. 1 illustrates the logical architecture of MPEG-2.

The timing in MPEG-2 standard is designed to support both *inter-* and *intra-stream synchronization* [5].

Intra-stream synchronization is based on two types of *Time Stamps* (TSs). Indeed, in the header of a PES packet we can include the *Decoding Time Stamp* and *Presentation Time Stamp* (DTS and PTS) of the first AU contained in that packet. They indicate the time at which the AU must enter the decoder and the time at which it must be delivered to the presentation device. DTS and PTS can be different because display order can differ from decoding order (e.g., P frames of video streams). The AUs not labeled with DTS and PTS are displayed according to the known frame/sample rate and the organization of the ES: any-

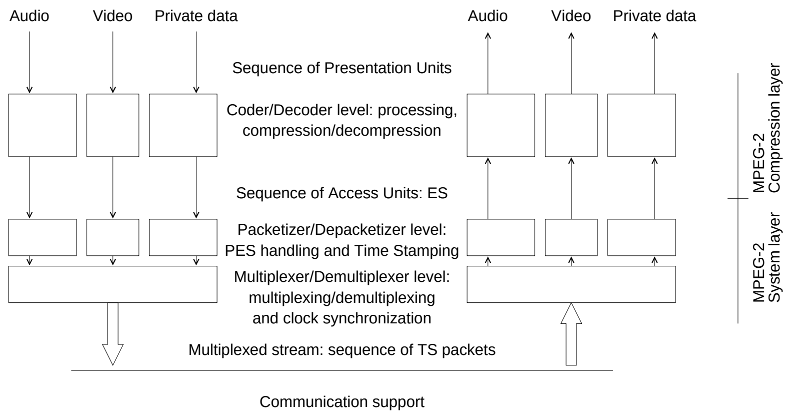


Fig. 1. MPEG-2 compression and system layers.

way, DTS/PTS fields must be encoded in the single stream every 700 ms (at least). Intra-stream synchronization is then possible if a common clock is shared among the sender, the receiver and the multiplexed streams.

Inter-stream synchronization covers this task. The sender's clock is periodically indicated to the receiver by means of the *Program Clock Reference* (PCR) field (at least one every 100 ms, if the TS structure is used). The receiver must lock its local clock on the incoming PCR in order to supply a "clean" time base for decoding and presentation purposes. The MPEG-2 standard specifies the following features for the system clock:

– *System Clock Frequency*: 27 MHz.
– *Frequency Drift*: 540 Hz.
– *Rate of Change* in the *System Clock Frequency*: 75×10^{-3} Hz/s.

Locking of clock is usually achieved through *Phase Locked Loops* (PLLs) mechanisms. In this paper we address the problem of clock synchronization comparing traditional PLLs with a new, non-linear approach that seems more appropriate for use on top of jittered connections, such as ATM.

3 Mapping of MPEG-2 streams over ATM

We consider in this section the approach taken by the ATM Forum in mapping MPEG-2 streams onto ATM cells for audiovisual services. The ATM Forum specifies that a *Single Program Transport Stream* (SPTS) is carried by one single *Virtual Channel Connection* (VCC), where a SPTS is a multiplexed stream of media sharing the same time base.

Between the TS level (sequence of 188-bytes long packets, belonging to a SPTS) and the ATM cells level (sequence of 5+48-bytes long cells, belonging to a unique VCC) two interfaces are present, as plotted in Fig. 2.

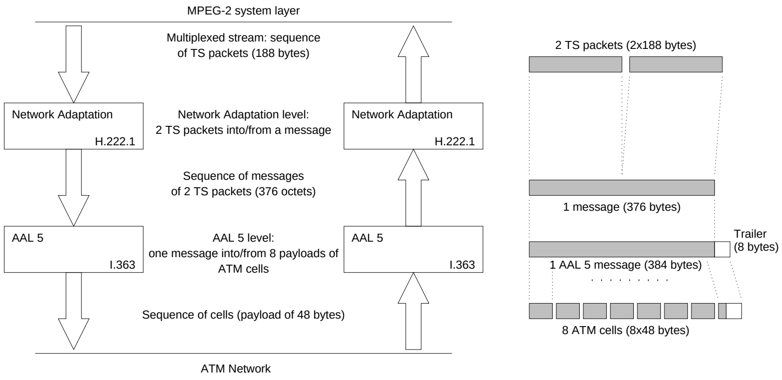


Fig. 2. ATM Forum mapping of audiovisual services onto ATM.

First, a *Network Adaptation* layer merges two TS packets into a message of 376 bytes (or conversely at the receiver side).

Second, this message is passed to the *ATM Adaptation Layer* (AAL) type 5 that adds a trailer, splits the message in chunks of 48 bytes and maps these chunks onto the payload of ATM cells. Reverse operations are performed at the receiver side.

The specification of ATM Forum is designed for CBR MPEG-2 [6]: nevertheless, it is expected that a close approach will be chosen for the next VBR MPEG-2 specification. Moreover, it is noticeable that the presented scheme delegates de-jittering and synchronization tasks to the application and error correction issues to the physical layer.

4 Clock synchronization by means of PLLs

Mapping of MPEG-2 Transport Streams onto ATM networks introduces delay: the processing and transfer operations contribute to this factors of de-synchronization.

The introduced delay has two components. First, a *Fixed Delay* that can be predicted, evaluated and/or bounded at run-time: this delay affects in the same manner all the elements of the stream (i.e., video, audio, DTS, PTS and PCR indications). It turns out that it can be easily compensated by delaying all the DTS/PTS indications of the same, suitable value. The system presents a latency that usually is less than a second. Second, a *Statistical Delay* or *Jitter* with an unpredictable distribution around his long-term average value of zero.

The jitter influences the clock reconstruction scheme at the receiver: indeed, if not equalized, early arrivals of packets with PCR indications cause buffers to be wasted at higher rate than required, PUs to be discarded due to the incoherent PTS indication and so on. Late arrivals, conversely, cause overflows of buffers and reduction of presentation rate. The PLL mechanisms allow clock synchronization despite the presence of noisy PCR.

PLLs was originally designed for phase-locking purposes of modulated signal carriers in the early thirties [7]. In the classical implementation (Fig. 3) the receiver employs a local oscillator to reproduce the incoming signal.

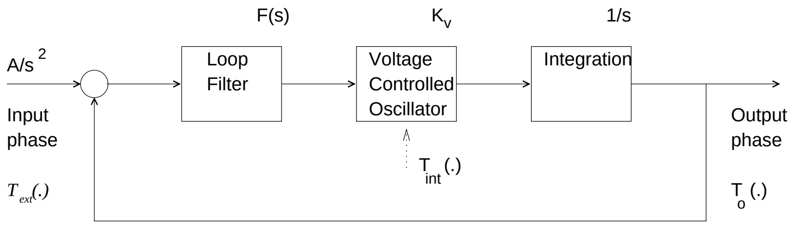


Fig. 3. Classical PLL.

For coherent phase-locking, the local frequency should be "close enough" to that of the incoming signal. The frequency shift is driven by the loop filter via a voltage applied to the *Voltage Controlled Oscillator* (VCO). The phase error fed a low-pass filter that modulates the frequency to minimize the phase error. The phase error is averaged over a given period of time to minimize the effects of noise in the incoming signal. Only a little information is needed by the local oscillator to lock on a stable input signal.

4.1 Mathematical model of PLLs

Mathematically, the analog PLL is modeled by the following *closed-loop transfer function* in the Laplace domain (refer to Fig. 3):

$$H(s) \;=\; \frac{T_o(s)}{T_{ext}(s)} \;=\; \frac{K_v F(s)}{s + K_v F(s)} \;. \tag{1}$$

By means of this transfer function, it can be demonstrated that an input ramp $T_{ext}(s) = \frac{A}{s^2}$ is perfectly tracked if the loop filter $F(s)$ is characterized by (at least) one pole located in the origin. Therefore, the simplest way to do this is to set a loop filter with the following transfer function:

$$F(s) \;=\; K_p + \frac{K_i}{s} \;, \tag{2}$$

with a *proportional* (K_p) and an *integrative* (K_i) action. The PLL is then described by a *second order loop*:

$$H(s) \;=\; \frac{2s\xi\omega_n + \omega_n^2}{s^2 + 2s\xi\omega_n + \omega_n^2} \;, \tag{3}$$

where:

$$\omega_n \;=\; \sqrt[2]{K_i K_v} \;,\; \xi \;=\; \frac{1}{2}\sqrt[2]{\frac{K_v}{K_i}}K_p \;, \tag{4}$$

are, respectively, the *natural frequency* and the *damping factor* of the loop.

When using PLL in the case of MPEG-2 clock synchronization, we move in the *discrete domain*. Samples of the input ramp enter the discrete PLL at given arrival times and are affected by delay and delay variations (i.e., the noise factors). With some algebra, it can be shown that the analog PLL described above maps onto a discrete PLL. Computationally, this can implemented in the following way:

$$
\begin{aligned}
T_o(k) &= T_o(k-1) + V(k-1)(T_{int}(k) - T_{int}(k-1)) \ , \\
V(k) &= K_v[prop(e(k)) + int(e(k))] \ , \\
prop(e(k)) &= K_p e(k) \ , \\
int(e(k)) &= int(e(k-1)) + K_i e(k) \ ,
\end{aligned}
$$

where $T_o(\cdot)$ is the locked clock, $T_{int}(\cdot)$ is the free internal clock value and $e(\cdot) = T_{ext}(\cdot) - T_o(\cdot)$ is the error, assuming that PCRs are carried as $T_{ext}(\cdot)$. The index k indicates the arrival of the $k-th$ PCR.

4.2 Simulation results of PLLs

We notice that the dynamical behavior of this PLL is driven by the pair K_p, K_i (alternatively ω_n, ξ): they determine the amount of residual jitter after locking and the locking speed of the PLL. For different cases of K_p, K_i, we simulate the transmission of PCR indications (at 50 ms intervals) affected by a fixed delay of 4 ms and a normally distributed jitter of average zero and 1 ms variance. The initial difference between frequencies is 2000 *parts per million* (ppm). In Fig. 4 we plot the residual frequency error (in ppm) for three cases of K_p, K_i (corresponding values of ω_n and ξ are derived).

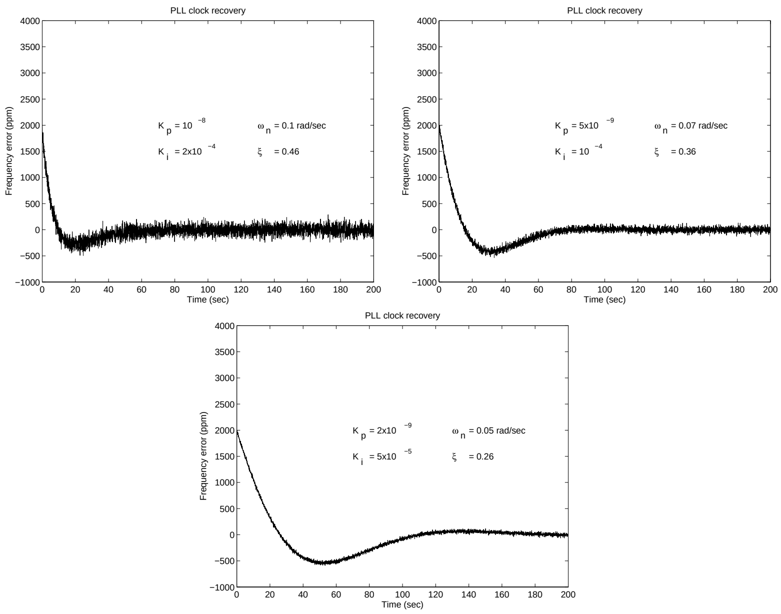


Fig. 4. Dynamic behavior of classical PLLs.

These PLLs reveal a symmetrical behavior: either we have a short transient phase and a poor clock reconstruction or transient is longer and clock is better locked. This is due to the linearity of PLLs: short transient is obtained through a large filter band, hence noise enters the system, while good phase locking implies a narrow band, hence long transient.

5 The non-linear technique for clock synchronization

The linear characteristics of classical PLLs do not fullfill both the requirements of fast and robust locking. Moreover, PLLs are not optimized for clock locking and/or recovery (i.e., when an input ramp enters the system), because they were originally designed to perform phase locking of periodic signals. The method itself and the needed circuitry can be bypassed by the proposed technique.

5.1 Mathematical approach to the non-linear technique

The alternative approach that we propose is based on a non-linear scheme for clock recovery. We first observe that any pair of clocks (i.e., sender and receiver clocks) can be related through the following expression:

$$T_{ext}(\cdot) \;=\; a\, T_{int}(\cdot) + b \;, \tag{5}$$

at any instant, where a and b (respectively the *frequency factor* and the *offset* parameters) are fixed, provided that the two clocks are stable in frequency (otherwise $a(\cdot)$ and $b(\cdot)$ move in time). Our method is able to directly estimate these two parameters. It interpolates with a straight line the points in the plane $(T_{int}(\cdot), T_{ext}(\cdot))$, corresponding to the arrival times of PCRs and the PCR indications it-selves (Fig. 5), affected by delay and jitter.

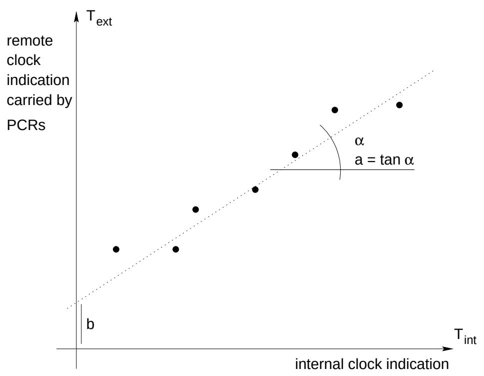


Fig. 5. Interpolation of PCR indications.

It is non-linear in the sense that we use a min-square interpolation. Slope and offset of the straight line are the tracked parameters a and b and are evaluated as follows:

$$(\bar{x}, \bar{y}) = (\frac{\sum_{i=0}^{n} x_i}{n}, \frac{\sum_{i=0}^{n} y_i}{n}) \quad \text{is the baricentric point,}$$
$$cotan(2\alpha) = \frac{\sum_{i=0}^{n}[(x_i - \bar{x})^2 - (y_i - \bar{y})^2]}{2\sum_{i=0}^{n}[(x_i - \bar{x})(y_i - \bar{y})]} \ ,$$
$$a = tan(\alpha) \ , \ b \ = \ \bar{y} - a\bar{x} \ .$$

Where, for the sake of simplicity, we have indicated with x and y the values along, respectively, the axes $T_{int}(\cdot)$ and $T_{ext}(\cdot)$.

5.2 Implementation of the algorithm

In order to be implemented, the algorithm requires to be recursive and computationally light. It can be tailored to follow the closed loop approach, then resulting in two interconnected blocks as in Fig. 6.

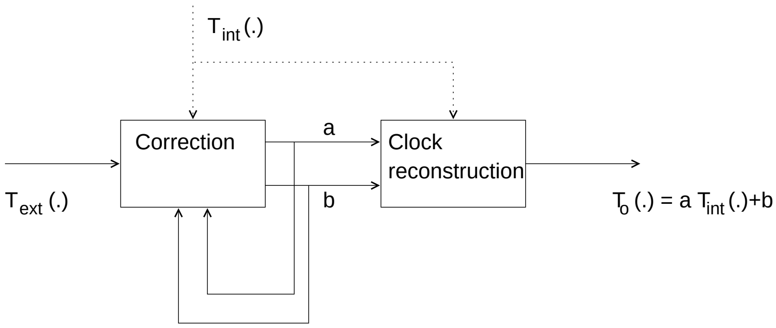


Fig. 6. Separated functionalities of the non-linear technique.

Reconstruction block recovers the remote clock by means of the internal clock and the pair a, b. Correction block updates a and b according to the incoming PCRs and the previous values: it can be designed to handle any length of the PCR trace (1 to ∞), thanks to recursion. If we assume that clocks have close frequencies ($a \simeq 1$), we can simplify the evaluation of α. On the receipt of the $n-th$ PCR indication $(x(n), y(n))$, we update the system in the following way:

$$(\bar{x}^{(n)}, \ \bar{y}^{(n)}) = (\frac{(n-1)\bar{x}^{(n-1)} + x(n)}{n}, \frac{(n-1)\bar{y}^{(n-1)} + y(n)}{n}) \ ,$$
$$\text{the updated baricentric point,}$$

$$\alpha^{(n)} = \frac{\pi}{4} + \beta^{(n)} ,$$
$$\beta^{(n)} \simeq -\frac{\sum_{i=0}^{n} x(i)^2 - \sum_{i=0}^{n} y(i)^2 - n(\bar{x}^{(n)^2} - \bar{y}^{(n)^2})}{4\sum_{i=0}^{n} x(i)y(i) - 4n\bar{x}^{(n)}\bar{y}^{(n)}} ,$$

referring to n as the current PCR, to $n-1,\ n-2, \ldots,\ 0$ as those previously received.

The performance of the present technique is compared to the classical PLL in the same test conditions: PCR are sent every 50 ms, affected by 4 ms of fixed delay and a normally distributed jitter with zero average an 1 ms of variance. In Fig. 7 are plotted the residuals frequency errors for three different cases of window length (i.e., the length of the PCR trace used to estimate a and b). Here we guess an initial frequency equivalent to the local one and we adapt it by filling the PCR trace with the incoming indications.

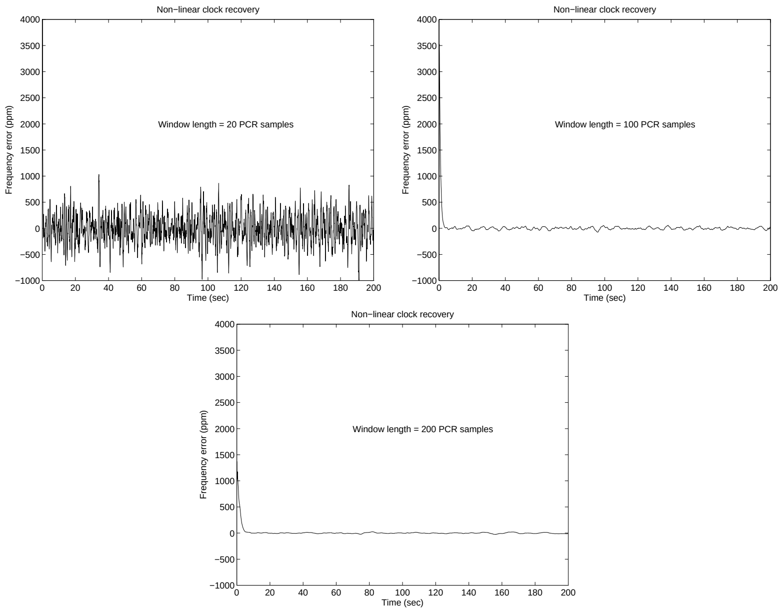
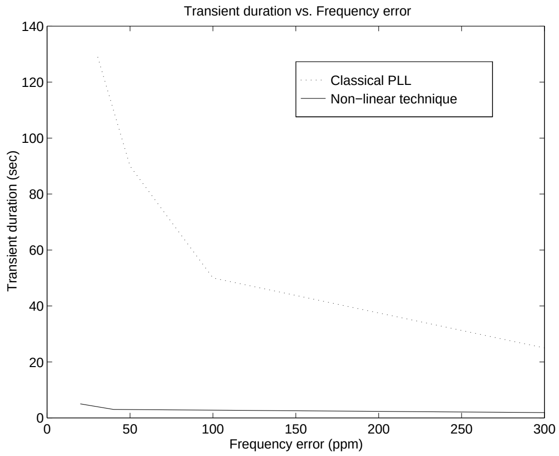


Fig. 7. Dynamical behavior of non-linear clock recovery.

The non-linear clock recovery scheme allows a fast-locking of the receiver clock. Moreover, the locked clock is really more jitter-free and tracks very close the sender's clock.

A comparative analysis of PLL and non-linear technique is plotted in Fig. 8. We consider the transient duration and the residual frequency error as the features of the two clock recovery schemes. It is worth to notice that, with an



Test conditions: PCR rate 50 ms, delay 4 ms, jitter variance 1 ms

Fig. 8. Compared performances of PLLs and non-linear technique.

equal residual frequency error, we can easily reduce the transient duration by a factor ten: this represents, therefore, a more efficient way to satisfy the timing requirements of MPEG-2.

6 Conclusion

Clock synchronization of MPEG-2 streams with jitter (where jitter is introduced by network and network protocols) is possible with PLLs: they can lock the local frequency onto the remote one or speed up the transient period associated with locking, but not both simultaneously.

PLL's behavior depends on the values of the parameters K_p and K_i (i.e., the loop filter designed to absorb jitter). PLL is a linear scheme for clock synchronization: increasing jitter is roughly the same as increasing the frequency error in a proportional manner.

In this case, a non-linear scheme is more suitable for absorbing jitter and synchronize clocks: non-linearity simultaneously satisfies quick-locking and clock-fidelity. Our technique has one degree of freedom in the design, the window length (i.e., the length of the PCR trace), and embeds a min-square algorithm to map the local clock frequency onto the sender's frequency. It is worth to notice that possible long-term drifts of the local frequency are completely absorbed by this non-linear scheme: indeed, the algorithm compensates $a(\cdot)$ with an equivalent value in the opposite direction.

The advantages of using this non-linear technique can be found in the case of interactive applications (i.e., [8]), in particular if a strong amount of jitter is introduced by communication support. If we plan to use traditional PLLs, we are faced with an initial latency of the system (transient of one or more seconds can appear) or to a poor quality of the presentation (too much residual jitter, so it is likely that we discard Presentation Units) or to an higher delay (equalization of residual jitter by delaying the presentation). This negatively influences the degree of interactivity of the application itself. Instead, using the proposed non-linear technique, latency is reduced by a factor ten, the clock has less jitter and the application more meets the interactivity requirements [9].

Moreover, thanks to the additional advantage of being implementable in software without requiring particular computational power, the efficiency of such scheme should not be limited to the case of MPEG-2 clock synchronization.

Other advantages are that it can be implemented in software without requiring particular computational power and can also be used on top of very "noisy" communication supports.

Further work should go in the direction to explore the benefits in terms of presentation quality, when used to synchronize multimedia streams, but also to precisely synchronize multiple end systems for distributed application with time constrained media, thanks to its long term jitter-independence.

Acknowledgement

The authors would like to thank S. Koppenhœfer for his technical comments and for the improvement of the editorial quality of this manuscript.

References

1. ISO/IEC, *Information technology - Generic coding of moving pictures and associated audio information - Part 2: Audio*, Oct. 1994. DIS 13818-2.
2. ISO/IEC, *Information technology - Generic coding of moving pictures and associated audio information - Part 3: Video*, Oct. 1994. DIS 13818-3.
3. ISO/IEC, *Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems*, Oct. 1994. DIS 13818-1.
4. G. Blakowski and R. Steinmetz, "A Media Synchronization Survey: Reference Model, Specification, and Case Studies," *IEEE JSAC*, vol. 14, pp. 5–35, January 1996.
5. R. Steinmetz and K. Nahrstedt, *Fundamentals in Multimedia Computing and Communications.* Englewood Cliffs, NJ, USA: Prentice - Hall, 1995.
6. ATM Forum, *Video on Demand Specification*, 1.1 ed., Oct. 1996.
7. F. M. Gardner, *Phaselock Techniques.* New York, USA: John Wiley & Sons, Inc., 1979.
8. A. Basso, O. Verscheure, and R. Noro et al., "A Multimedia Architecture for Medical Tele-Imaging over ATM," in *CAR '96*, Paris, France, June 1996.
9. R. Noro, "Un mécanisme distribué de synchronisation appliqué à un cas de téléconsultation," in *JDIR '96*, Paris, France, September 1996.