Title: Multimedia Playout Synchronization Using Buffer Level Control

Authors: Daniel Köhler, Harald Müller

Language: English

Citation: Köhler, D.; Müller, H.: Multimedia Playout Synchronization Using Buffer Level Control. - In: Multimedia: Advanced Teleservices and High-Speed Communication Architectures, Proc. of the 2nd Int. Workshop IWACA '94, Heidelberg, September 26-28, 1994. Ed.: R. Steinmetz. Berlin: Springer, 1994, pp. 167-180.

Multimedia Playout Synchronization Using Buffer Level Control

Daniel Köhler, Harald Müller

Lehrstuhl für Kommunikationsnetze, Technische Universität München,
D-80290 München, Germany

Abstract. This paper presents a new approach to the intrastream synchronization problem. The proposed technique is based on controlling the receiver buffer. The basic idea is to find realistic models for the control loop components and to apply automatic control methods to choose an appropriate control algorithm and to derive stability criteria and transient behavior of the whole system. The theoretical results are shown to have good accordance to real, measured system behavior. The synchronization method has been implemented and first experiences could be gathered with a video phone application.

1 Introduction

Many multimedia applications, like remote camera, video conferencing, and video on demand, require the delivery of continuous audio and video information in real-time. When using asynchronous networks for data transmission, timing information of the media units produced gets lost and a mechanism is necessary to ensure continuous and synchronous playback at the receiver side. The so-called intrastream or playout synchronization is one key concern in multimedia synchronization. In this paper, we present a playout synchronization technique that is based on receiver buffer control. By realistic modeling of the playout components, a suitable control algorithm can be determined and stability criteria and transient behavior of the system can be theoretically derived when using automatic control methods. This allows for the design and dimensioning of a well behaving system without having to rely on experiments.

The synchronization method was implemented using the PC-based multimedia system Action Media II (DVI). The first application to use the proposed technique is a video phone that allows bidirectional communication over local and high-speed metropolitan area networks (MAN) including audio and full motion video.

In Sect. 2, a classification of other intrastream synchronization solutions is given and the basic principles of the proposed technique are described. Section 3 gives a detailed description of the control loop components and how they can be modeled. Stability criteria and transient behavior of the system, including a well chosen regulator, are theoretically derived. In Sect. 4, some details concerning the implementation of the proposed synchronization technique using the Action Media II system are explained. Section 5 gives an impression of a sample video phone application that was realized based on the proposed synchronization method.

2 Playout Synchronization

The key concern of this paper is playout synchronization, which is also termed intrastream synchronization. A continuous media stream consisting of media units is produced synchronously at the sending station and transmitted over an asynchronous packet network, where jitter is introduced. On the other side, one or more receivers have to play out the continuous media units in a synchronous way again. When no synchronization measures are taken, the clocks of sender and receiver will be slightly different and buffer overflows or starvations will occur during the transfer process. Uncontrolled buffering compensates jitter effects to a certain extent, but cannot compensate clock asynchronism over longer periods of time.

2.1 Classification of Playout Synchronization Solutions

Solution space for playout synchronization consists of three almost orthogonal design criteria with two main choices in each dimension. The first decision is, whether the systems have an explicit common understanding of time or not. In the former case, some kind of clock synchronization takes place. The presentation time of a media unit can be calculated from an absolute or relative time stamp carried with every unit. If no clock synchronization takes place, playout synchronization can be achieved based on buffer control mechanisms.

The second criterion is the location of synchronization actions. They can be performed either at the sender or the receiver of continuous media information. Sender control always requires some kind of feedback.

The third dimension distinguishes the methods that are used to correct asynchronism. This can be done by speeding up or slowing down presentation or production speed of media units, or by stuffing. The second method is well known from bit or byte synchronization and means deleting or inserting media units in our context.

2.2 Related Work

Escobar et al. [5,6] suggest a clock synchronization method. The transmitted media units contain time stamps that allow the receiver to determine presentation time. A very similar technique using the notion of a common LTS (logical time system) for several media streams, is introduced by Anderson et al. [2]. Corrective actions are performed by skipping and pausing. The mechanism is mainly applicable to single-site workstations.

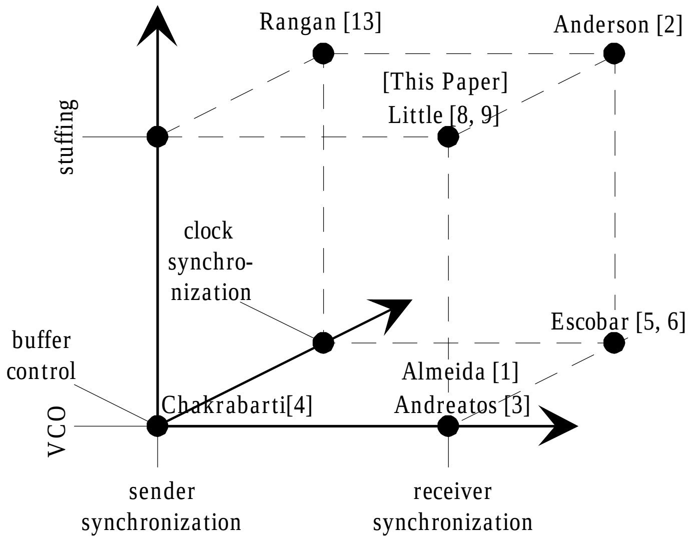

Fig. 1. Classification cube for intrastream synchronization and existing solutions

Another kind of clock synchronization used in conjunction with sender control, is proposed by Rangan et al. [13]. They assume less powerful receiver stations who feed back the number of the currently displayed media unit to the sender. There, corrective actions on each data stream can be taken, if necessary. To regain synchronization, media units are deleted or duplicated. Basic assumption here is a limited and known amount of delay jitter. This technique is mainly used as interstream synchronization for synchronous retrieval from networked file servers.

Chakrabarti et al. [4] suggest a solution, where the receiver clock drives the periodic, synchronous playout of media units. Depending on the amount of used buffer space in the receiver, the sender frame rate is controlled. As with other feedback methods, increasing network delay slows reactivity and stability cannot be proved in the case of unpredictable delay. The parameters of the control algorithm are derived experimentally.

PLL (phase locked loop) solutions are mainly known from bit synchronization. The basic principle is to compare a buffer level at the receiver to a nominal value. A loop filter forms the input voltage for a VCO (voltage controlled oscillator), which generates the buffer readout clock. Frequency usually drifts only by small amounts of some ppm. Almeida et al. [1] describe a PLL mechanism for bit synchronization of AAL services with synchronous timing relation. The byte level of the receiver FIFO is used to calculate the byte read clock and several loop filters are compared but not theoretically analyzed. A similar approach is taken by Andreatos et al. [3], who transmit scan lines of uncoded video signals over a 140 Mbit/s network with bounded delay jitter. A PLL is used to recover the line clock for the receiver. Control parameters of the PI controller are determined experimentally and neither stability nor dynamic behavior of the system is theoretically analyzed.

Little et al. [8,9] propose a receiver-based synchronization technique, which mainly realizes interstream synchronization, but some aspects are related to intra-stream synchronization. When the level of the receiver queue reaches a high or low threshold, frames are dropped or duplicated. Since the playout process is stated to be purely synchronous, stuffing is used instead of clock adjustment. Given that either sender or receiver clock speed is higher, the queue level will always tend to stay at one threshold, meaning either lower disturbance immunity or higher delay. Choosing the thresholds closer to each other leads to very frequent corrective actions and consequently errors. The correction function that controls frame drop and duplication is arbitrary, but only a constant rate-based function was investigated. The mechanism is based on the assumption of guaranteed network resources and has to react only on reservation violations or frame losses.

2.3 Proposed Synchronization Technique

The proposed synchronization mechanism is intended to act in an interconnected LAN and high-speed MAN environment, i.e. available bandwidth or delay cannot be guaranteed and transmission is based on best effort. Jitter effects are introduced by network transmission and by non-real-time operating systems in use. Figure 2 shows a typical distribution of frame arrival spacing at the receiver, that was measured on one LAN segment during normal traffic conditions. Every 33 ms, a frame containing multiplexed audio and video information is produced at the sending station, which explains the resulting peak. The slow reaction of the messaging system at the sender causes the process that reads out and transmits coded frames often to find two frames in the buffer. This explains peaks at very small frame distances around 5-10 ms and at 66 ms. The gaussian-like distortion of the peaks is caused by the operating system and mainly network jitter effects. Media units are transmitted without error correction and thus can be lost.

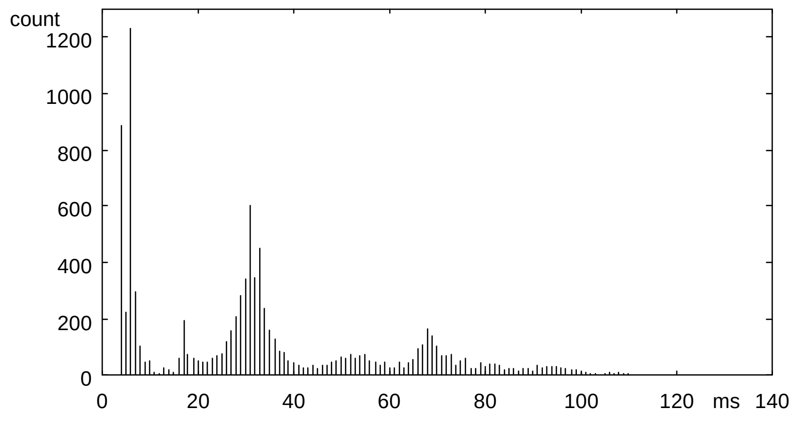

Fig. 2. Measured frequency distribution of the arrival spacing of received frames; frame rate 30 frames/s; transmitted over single ethernet segment.

We have chosen a receiver control that runs independently from the sender. This has the advantage that unpredictable delay cannot influence synchronization, and the method naturally extends to several receivers synchronizing independently to the same sender. Typical playback systems for continuous media and especially for video decompression with hardware support perform in a synchronous manner, i.e. after initialization frames are consumed at a fixed rate out of a buffer. The main purpose of the buffer is to compensate jitter effects introduced between frame production and delivery at the receiver. Because presentation time of the media units cannot easily be dictated and due to complexity of clock synchronization, our synchronization technique is based on controlling the level of the receiver frame buffer. The aim of the control is to keep the amount of media units in the buffer at a nominal value in the long term. The buffer control must not compensate short buffer fluctuations caused by jitter effects. In contrast to other solutions we count the amount of frames, not number of bytes, in the buffer. This is reasonable, since frames are produced synchronously but have variable frame sizes. Assuming constant delay, when the amount of frames in the buffer is held constant, sender and receiver are synchronized. Taking into account not only compressed frames but all media units in the receiver system up to the one currently being displayed, also compensates decompression jitter.

Since the mechanism should also be applicable to bidirectional communications, one of the main goals is to keep the delay caused by buffering and thus the buffer level as low as possible. On the other hand, network jitter dictates a certain amount of buffer for compensation. This contradiction suggests that deviations from an ideal buffer level to either side lead to disadvantages.

The main idea behind the proposed synchronization technique is the application of automatic control methods for buffer control. These allow the investigation of stability and transient analysis of the whole system. With these results the appropriate control algorithms and their parameters can be derived, as will be shown in detail in the following section.

As mentioned above, there are two principle choices for the regulating unit: clock adaption and stuffing. The human ear is extremely sensitive to speed changes when playing back audio information; variations of one percent or more can be perceived. Choosing even smaller tuning ranges makes the control mechanism react too slowly to buffer level changes in our case, since there are only few frames in the buffer. On the other hand, only few perceivable errors are introduced by frame stuffing. Insertion and discard of audio frames can only be perceived as soft crackling when done frequently and with several frames at once. There is no change in pitch at all. Video frame stuffing is not perceivable by the user. For these reasons our synchronization technique uses frame stuffing. Note that in case of transmission errors, some kind of method is already necessary to replace lost or incorrect frames, so the additional implementation effort for the frame stuffing mechanism is low.

3 Modeling the Control Loop

Due to the previously mentioned reasons, uncontrolled operation of the receiver leads to buffer overflow or starvation. The latter appears even more frequently in the case of uncompensated frame losses. These situations produce noticeable disturbances which have to be compensated by playout synchronization measures. As mentioned before, our solution is based on controlling the average level of the receiver frame buffer to a nominal value. This secures stable and fluent playout of the data and yields low delays.

In this section we describe a model for the control loop components that approximates the real playout system very close. A realistic model is essential to investigate system behavior and dimension the control components by application of automatic control engineering.

3.1 Components

Figure 3 shows the components of the control loop. The frame buffer is filled by incoming frames from the network. Due to jitter effects, the arrival spacing of the frames varies temporarily. However, the average is constant and is forced by the sender's clock. The buffer is emptied with a constant rate, driven by the receiver's playout clock.

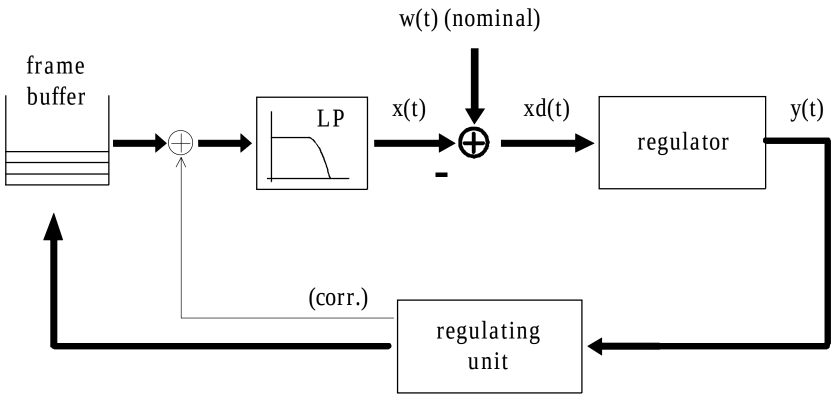

Fig. 3. Principle of the control loop

The level of the frame buffer is measured and filtered by a low pass component, before further processing is done. The buffer level includes compressed frames, frames currently being decompressed and already decompressed frames waiting to be displayed. This avoids introducing additional jitter due to variable decoding times.

Low pass filtering is necessary for two reasons. First it does an averaging of the measured buffer level since it passes only low frequent level changes. The sender's clock can only be derived from the average arrival spacing. High frequent changes caused by temporal delay and jitter must therefore be eliminated. The second reason for the filter is to keep the sampling theorem, which is important later when looking at the digital system.

A comparator subtracts the filtered buffer level *x(t)* from the nominal value *w(t)*. Usually, *w(t)* is a constant, ideal value. The calculated deviation *xd(t) = w(t) - x(t)* becomes the input for the regulator which calculates a corresponding output value *y(t)*. The regulator is the only part which is not given by a system component but can be chosen in a more or less optimal manner, as we will see below. According to the chosen control function, the output value of the regulator gives the amount of deviation, which has to be corrected in order to adjust the buffer level to its nominal value. The actual corrective action is performed by the regulating unit by inserting or discarding frames. The shown correction output does not appear in the theoretical model but is an implementation detail described later.

The regulating unit, the frame buffer and the low pass filter can be grouped together to form the controlled system. The control loop therefore divides into two main sections: the controlled system and the regulator. To investigate the loop behavior, the transfer functions of both sections are needed. They are given in the complex variable domain, which has the advantage that the transfer function of the closed loop can easily be calculated using algebraic operations. Stability of the loop can be investigated in the complex domain and inverse Laplace transformation yields temporal behavior of the system.

Modeling the controlled system is probably the most difficult task when applying control engineering methods. In general, no exact description of a real world system is possible. Therefore, a good approximation has to be found, which describes the system behavior well enough. No measurements can be made with the isolated controlled system in our case, as it does not operate in a stable way without control. So we have to take a look on the three parts on their own.

The transfer function of the frame buffer can be described by a proportional (P) component with amplification factor K=1. This explains out of the fact that inserting or deleting a frame immediately causes the buffer level to change by ± 1.

The low pass filter is a classical well known first order delay component (PT_1) with the transfer function:

$$F_{LP}(s) = K_S \frac{1}{1 + s \cdot T_1} \ ;$$

K_S: low pass amplification, always set 1
T_1: low pass time constant

Finally, the regulating unit can also be modeled by a P component with K=1, if we assume that it can add or discard frames without delay. This assumption is true as long as the regulator dictates small changes of the buffer level in the area of a few frames (small-signal response), but becomes worse at high changes. As mentioned, the regulating unit acts by inserting or discarding frames to or from the data stream. Frames can only be discarded once when being received. To achieve symmetrical behavior and not to introduce visual or audible artifacts, doubling is done only once for each frame. Thus, corrective actions of more than one frame cannot take place immediately, but have to be taken in steps. This becomes even worse, if intercoded frames occur within the stream and not every frame can be doubled or discarded. Figure 4 shows the assumed response of the regulating unit to a step of height n_o at time t=0. It is assumed that intraframes appear at fixed time intervals T_{SP}, which is a multiple of the frame time $T_F = 1/f_F$. Only one frame can be doubled every time interval T_{SP}, which results in the stepwise ascent until n_o is reached. This behavior can be approximated in a worst case manner by a PT_1 component with time constant T_2. This leads to a PT_2 behavior of the whole controlled system.

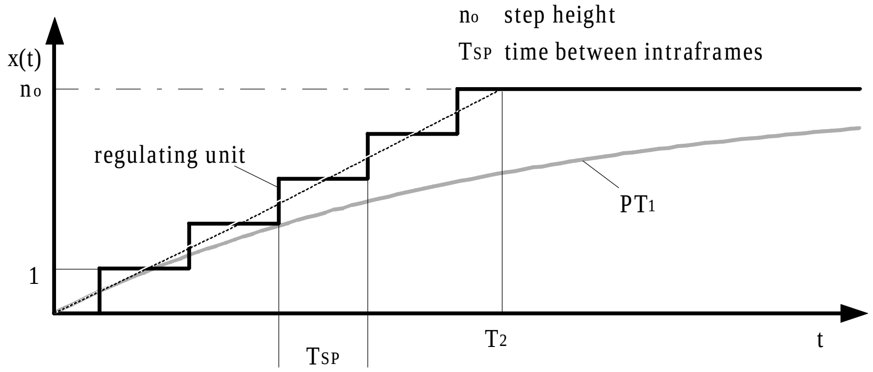

Fig. 4. Assumed step response of regulating unit and approximation with PT_1 component

3.2 Regulator Selection

With the knowledge about the controlled system, an appropriate regulator has to be found. We first assume that the contolled system has a first order delay (PT_1). This is a good approximation while the system is running and the buffer level changes within bounds of few frames (small-signal response at the operating point). To calculate stability in case of sudden high level changes as they can appear at connection setup or due to sudden network overload, the more exact model of a second order delay (PT_2) behavior is used for the controlled system (large-signal response).

After examination of several regulator types [7], an integrating (I) regulator turned out to be the best solution. It does not need a permanent deviation, but always regulates the actual value exactly to the nominal value. Further, the integrating behavior suppresses ripple of the input value and therefore supports the low pass filter in extracting the average filling level. Together with a PT_1 system, the I-regulator always yields stable system behavior. The transfer function of the I-regulator is:

$$F_r(s) = \frac{y(s)}{xd(s)} = \frac{K_i}{s}$$; K_i: integrating constant [1/s].

3.3 Reference Transfer Function

The reference transfer function describes the behavior of the input value *x(t)* in response to changes of the nominal value *w(t)*. Note that all functions are given in the complex variable domain. For the given PT_1-I loop, the reference transfer function is:

$$F_w(s) = \frac{x(s)}{w(s)} = \frac{F_r(s) \cdot F_s(s)}{1 + F_r(s) \cdot F_s(s)} = \frac{K_i K_s}{K_i K_s + s + s^2 T_1} ;$$

In case of modeling the controlled system with an PT_2 behavior, the reference transfer function becomes:

$$F_{w2}(s) = \frac{K_i K_s}{s(1 + sT_1)(1 + sT_2) + K_i K_s} ;$$

T_2 is the time constant of the regulating unit. Its value depends on input step height, frame rate and period between intraframes. Stability in case of disturbances can be derived from the disturbance transfer function, which has the same denominator but consinsts only of $F_s(s)$ in the numerator compared to the above formulas. As the dynamic behavior of the loop depends only on the denominator of the transfer function, the stability criteria for reference and disturbance transfer are the same.

3.4 Damping and Stability Criteria

Small-Signal response. First we will analyze the system around the operating point with PT_1 behavior of the controlled system. In this case, the damping *D* of the loop can be derived from the coefficients in the denominator of the reference transfer function $F_w(s)$:

$$D = \frac{1}{2 \cdot \sqrt{K_i K_s T_1}} ;$$

The damping value describes the dynamic behavior of the system, the following cases can be distinguished:

1. $D > 1$: aperiodic behavior.
2. $D = 1$: aperiodic borderline case, quickest response without overshoot.
3. $0 < D < 1$: damped oscillation.
4. $D = 0$: constant oscillation.
5. $D < 0$: rising amplitude oscillation.

As K_i, K_S and T_1 are positive values, it is evident that D is always positive and the system is stable. K_S and T_1 are given by system components, whereas the integrating constant K_i can be chosen to dictate loop behavior.

Large-Signal Response. With the controlled system being modeled as a PT_2 component, the transfer function of the closed loop becomes more complex. One way to investigate stability in this case is using the Hurwitz criteria, which constrains:

1. All coefficients a_0 to a_n of the denominator of the reference transfer function have to be greater zero.
2. The determinant derived from a_0 to a_n and its subdeterminants must be greater zero.
3. If these constraints are met, the system is stable.

Application of these rules leads to the following stability criterion:

$$K_i K_S < \frac{1}{T_1} + \frac{1}{T_2}$$

In this relation, K_S is chosen to 1 (low pass amplification) and T_2 depends on the step height and the interval between intraframes. As $1/T_2$ is always positive, a secure stability criterion is at least $K_i < 1/T_1$.

4 Implementation

Finally, the modeled system is realized in a digital environment. The low pass filter, the regulator and the regulating unit are realized in software. The buffer level is an integer value which must be scanned periodically and can only vary in steps of whole frames. Besides the extraction of the buffer level average, the low pass filter has to act as an anti aliasing filter for the following control loop components. We chose a scan frequency of 4 Hz at which the buffer is scanned and all calculations are done. The sampling theorem therefore forces a filter cut off frequency of 2 Hz, which leads to the value of T_1 = 0.5 s. The low scan frequency is sufficient as we want an average regulation of the buffer level. Further, it does not afford much performance at the host system.

The low pass filter was formed by a 2nd degree FIR (finite impulse response) filter which is easy to implement and has stable behavior. Its output at instant k is:

$$y_{LP}(k) = a_0\,[x(k) + x(k\text{-}2)] + a_1\,x(k\text{-}1);$$

The constants a_0 and a_1 have been calculated by a filter software [10] and rounded to 0.25 respectively 0.5. The integrating behavior of the regulator is approximated by a trapezoid calculation. Its output at instant k is calculated by:

$$y(k) = x_d(k-1) + Ki \cdot T_a \cdot \frac{x_d(k) + x_d(k-1)}{2} \;;\; T_a\text{: scan period (250 ms).}$$

The regulating unit finally decides, whether and how much frames need to be discarded or added. But some care must be taken to implement the control loop right. As can be seen in Fig. 3, the regulating unit offers a correction output whose value is added to the scanned buffer level value before it is fed into the low pass filter. This is necessary because only whole frames can be added or discarded. In case the integrator 'wants' e.g. only a half frame correction, this is done by the correction output of the regulating unit, which simulates a level change as long as it is below a whole frame. Without this measure the regulator would always integrate up and down periodically with the amplitude of one frame. This process takes place within some seconds, which is not critical for the loop, but would introduce additional and obviously unnecessary disturbances.

Figure 5 shows the theoretically derived step response of the closed control loop for two different damping values compared to the measured response of the real system. The regulating unit was modeled as a PT_1 component. Note that jitter effects present in the measured curves produce statistical deviations.

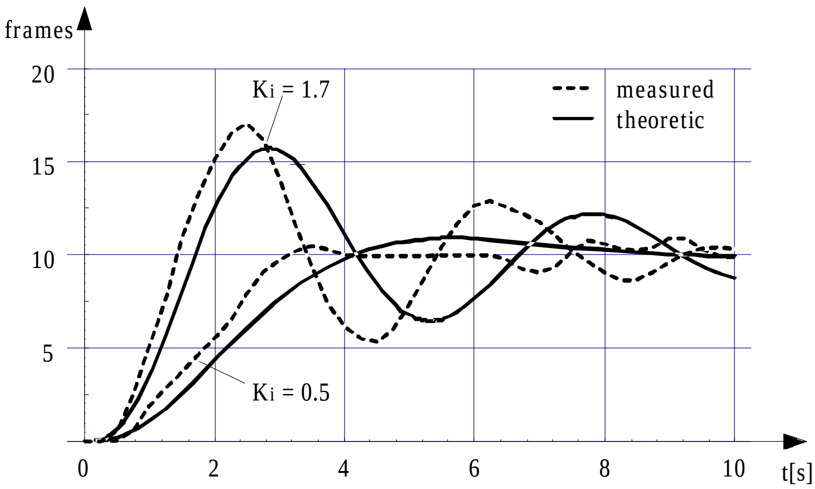

Fig. 5. Step response of the pT_2-I control loop and the real system to a step of 10 frames

The curves with integrating constant $K_i = 0.5$ are near the aperiodic borderline case and show almost no overshoot. The response of the real system is slightly quicker than theoretic behavior. This can be explained by the approximation of the stepwise ascent by a PT_1 component, which increases slower (see Fig. 4). With $K_i = 1.7$, the system comes closer to the stability bound. In this area, damped oscillation occurs. Again, the real system has a faster response and hence a higher oscillation frequency. The oscillation amplitude is reproduced almost exact. We also took measurements with higher K_i values and could verify the stability bound. For values with $K_i > 1/T_1 = 2$ [1/s], the system proceeds to oscillations increasing in amplitude. These results justify the choice of the PT_1 approximation for the regulating unit as basis for derivation of stability and dynamic behavior.

5 Application Example - Video Phone

The described control loop based playout synchronization is used in an application for bidirectional audio and video communication [11]. The video phone application was implemented under the operating system OS/2. Taking advantage of its multitasking capabilities, task priorities and graphical user interface, it is possible to run other applications in parallel. So, communication capabilities can easily be extended by e.g. data transfer.

For encoding, decoding and display of audio and video information, the IBM Action Media II system (DVI) is used. It offers the possibility to concurrently compress and decompress audio and full motion video with a frame rate up to 30 frames/s. For video coding, the real time video (RTV) algorithm is used, which offers a full color image resolution of 128x120 pixel, when doing coding and decoding in parallel. The algorithm uses both inter- and intraframe coding. The resulting data rate depends on the choice of several coding parameters and is between approximately 350 kbit/s and 1.2 Mbit/s in each direction. Audio coding is done either with a PCM or ADPCM algorithm and yields quality comparable to digital telephony.

For transport of the real-time audio and video information, the user datagram protocol (UDP) is used. As it is an unsecure datagram protocol, measures against frame corruption, loss and reordering have to be taken by the application program. Signaling for call management is separated from data transport and uses a special protocol. Besides call setup and release, parameters of the coding algorithms in both directions can be negotiated. This allows the user to change video or audio quality and respectively data rate even during an existing communication.

In the current implementation, the user can manually change the control loop parameters, i.e. the nominal buffer level and the regulator parameter K_i. Future investigations will include automatic adaption of these parameters to the current network situation, which can e.g. be estimated by online measurements.

At this point, we will sum up some experiences with the video phone in general and under different network conditions [7]. The control loop synchronizes the receiver within a few seconds at connection setup and keeps it stable during the communication. In usual LAN environments, corrective actions of the regulator are very seldom with intervals of some tens of seconds. In most cases the insertion and discard of frames cannot be perceived. Only in the case of low frame rates or high frame losses, tolerable crackle noises can be heard. Image artifacts can occur when losses or errors take place before intercoded frames. The round trip delay, which mainly depends on the nominal buffer level value, can be kept as low as 450ms with a frame rate of 30 frames/s. This still enables a comfortable communication.

We experienced with network loads on ethernet up to 60%, whereby only the jitter increased but the connection could be kept stable with approximately 1 frame loss per second. For testing the application over longer distances, a data mirror was installed at the remote end of a MAN, which just returned received frames to the sender. This is an even worse situation compared to having another video phone station at the remote end, because every media unit suffers twice the single transmission delay. Over a single distance of approximately 20 km inside the urban network, communication was possible without any noticeable disturbances. Only absolute delay and jitter increased, the latter could be compensated by choosing a slightly higher buffer level. Reasonable communication between two different MANs over about 250 km (Munich and Stuttgart) was only possible in non-busy hours. It should be noticed, that MAN interconnection is realized over a 2 Mbit/s link and that it is part of the german universities network, that extensively carries data traffic.

6 Conclusion

We described a control loop based model of an intrastream synchronization mechanism, which is not based on a specific system environment and may be used on multimedia playout devices working with frame oriented data. The synchronization mechanism is independent from the sender and based on regulating the level of the receiver frame buffer. Correction is done on frame level by inserting and discarding frames. Additional to the normal receiver synchronization, the control mechanism keeps the buffer level on a constant average level. This generates a small delay variance of the video and audio frames and enables a small amount of buffered frames, an important constraint within a bidirectional communication environment. In order to predict system behavior, realistic models of the control loop components were given. After choosing an appropriate control algorithm, stability and dynamic properties of the system have been derived. The results give good correspondance to practical measurements. Finally, the implementation of the synchronization method using a specific multimedia system has been described and experiences were reported.

References

[1] N. Almeida, J. Cabral, A.Alves: End-to-End Synchronization in Packet Switched Networks. In: Proc. of the 2nd Int. Workshop on Network and Operating System Support for Digital Audio and Video, 1991. Lecture Notes in Computer Science, Springer Verlag, Vol.614, 1992, pp.84-93.

[2] D.P. Anderson, G. Homsy: A Continuous Media I/O Server and Its Synchronization Mechanism. IEEE Computer Magazine, Vol.24, No.10, 1991, pp.51-57.

[3] A.S. Andreatos E.N. Protonotarios: Receiver synchronization of a packet video communication system. Computer Communications, Vol.17, No.6, June 1994, pp.387-395.

[4] S. Chakrabarti, R. Wang: Adaptive Control for Packet Video. Proc. of Int. Conference on Mulitmedia Computing, May 1994, pp.56-62.

[5] J. Escobar, D. Deutsch, C. Partridge: A Multi-Service Flow Synchronization Protocol. BBN Systems and Technologies Division, 1991, pp.1-16.

[6] J. Escobar, D. Deutsch, C. Partridge: Flow Synchronization Protocol. Proc. of the IEEE Globecom, Vol.3, 1992, pp.1381-1387.

[7] D. Köhler. Optimierung und Weiterentwicklung einer Bildtelefonanwendung unter OS/2. Diplomarbeit am Lehrstuhl für Kommunikationsnetze, Technische Universität München, Munich, Germany, 1994.

[8] T.D.C. Little, F. Kao. An Intermedia Skew Control System for Multimedia Data Presentation. In: Proc. of the 3rd Int. Workshop on Network and Operating System Support for Digital Audio and Video, 1992. Lecture Notes in Computer Science, Springer Verlag, Vol.712, 1993, pp.130-141.

[9] T.D.C. Little: A framework for synchronous delivery of time-dependent multimedia data. Multimedia Systems, Vol.1, No.2, 1993, pp.87-94.

[10] O. Mildenberger: Entwurf Analoger und Digitaler Filter. Vieweg Verlag, 1992.

[11] H. Müller: Bewegtbildkommunikation über LAN und MAN. In: Fokus Praxis, Information und Kommunikation, Bd. 5, Verteilte Multimedia-Systeme. Hrsg.: W.Effelsberg u.a. München, Saur Verlag, 1993, pp.126-141.

[12] S. Ramanathan, P.V. Rangan: Continuous Media Synchronization in Distributed Multimedia Systems. In: Proc. of the 3rd Int. Workshop on Network and Operating System Support for Digital Audio and Video, 1992. Lecture Notes in Computer Science, Springer Verlag, Vol.712, 1993, pp.328-335.

[13] P.V. Rangan, S. Ramanathan, et al.: Techniques for Multimedia Synchronization in Network File Systems. Computer Communications, Vol.16, No.3, March 1993.