Continuity and Synchronization in MPEG

P. Venkat Rangan, Srihari SampathKumar, and Sreerang Rajan

Multimedia Laboratory
Department of Computer Science and Engineering
University of California at San Diego
La Jolla, CA 92093-0114

E-mail: venkat@chinmaya.ucsd.edu, Telephone: (619) 534-5419, Fax: (619) 534-7029

accepted for publication in

IEEE Journal on Selected Areas in Communications

Special Issue on Multimedia Synchronization

June, 1995

CONTINUITY AND SYNCHRONIZATION IN MPEG

P. Venkat Rangan, Srihari SampathKumar, and Sreerang Rajan

Multimedia Laboratory
Department of Computer Science and Engineering
University of California at San Diego
La Jolla, CA 92093-0114

E-mail: venkat@chinmaya.ucsd.edu, Telephone: (619) 534-5419, Fax: (619) 534-7029

June 8, 1995

Abstract

The requirement of continuous retrieval, and the presence of multiple media streams whose display must proceed in a mutually synchronized manner are the distinguishing features that are unique to digital multimedia. In the emerging international multimedia encoding standard MPEG, continuity and synchronization are handled at different layers of the multimedia stream. We discuss how they are specified and propose techniques for their implementation within a distributed multimedia environment.

1 Introduction

Multimedia services, in which stored media objects are retrieved on demand by end users, are rapidly emerging to be offered on digital communication networks. There are two distinguishing requirements of media playback: intra-media continuity and inter-media synchronization [6]. In the emerging international multimedia encoding standard MPEG (*Motion Picture Experts Group*), continuity and synchronization are handled at different layers of the multimedia stream. In this paper, we discuss how they are specified and propose techniques for their implementation within a distributed multimedia environment.

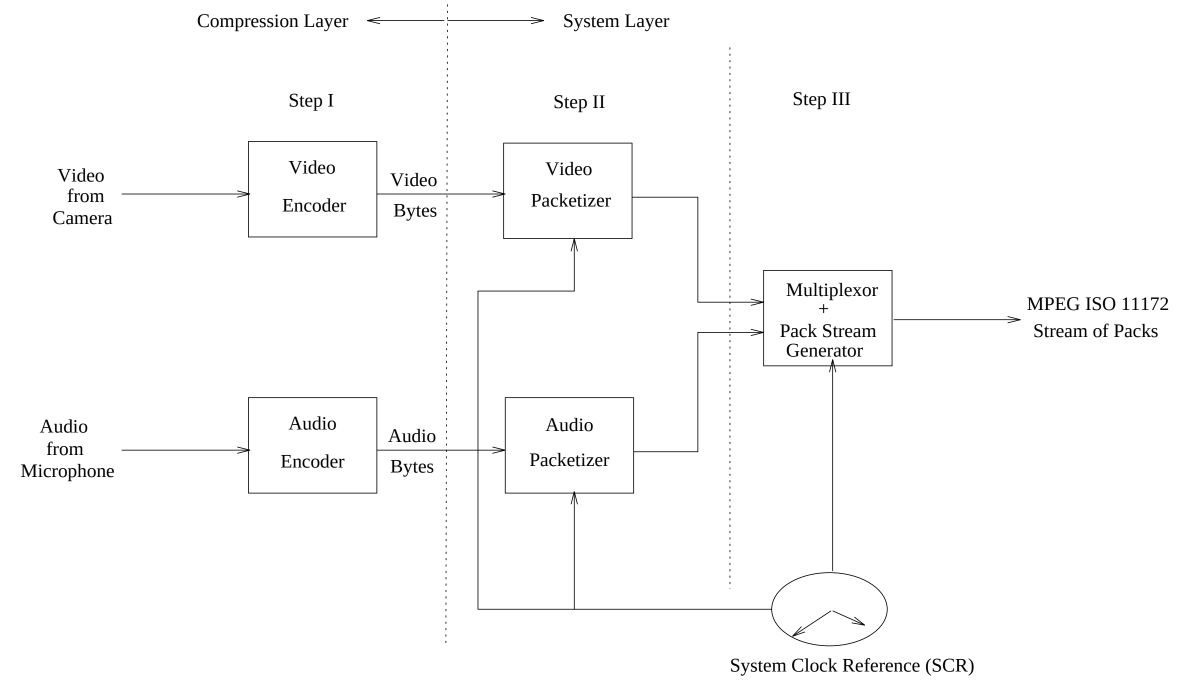

Figure 1: *MPEG Encoder System Configuration*

In order to understand MPEG synchronization, it is necessary first to familiarize with the encoder's system architecture, without of course going into details of the encoding algorithms themselves. At the time of encoding in MPEG [7], in the first stage, video frames and audio samples are separately encoded, and an independent stream of bytes is output for each media channel. This is shown in step I of Figure 1. Each media stream is then packetized independently (step II). Packets from the different streams are interspersed to form a single multiplexed stream of packets, and the multiplexed stream is then organized into packs, with each pack comprising of an integral number of packets (step III). The multiplexed stream is called an ISO 11172 stream.

At the time of decoding (Figure 2), the packs are broken into packets, and packets of different media streams are demultiplexed into their respective decoders (step I). The video decoder then outputs its presentation units, which are frames, and the audio decoder outputs PCM (*Pulse Code Modulation*) samples (step II). The demultiplexor and the media decoders are together referred to as the *System Target Decoder (STD)*. Table 1 summarizes this terminology.

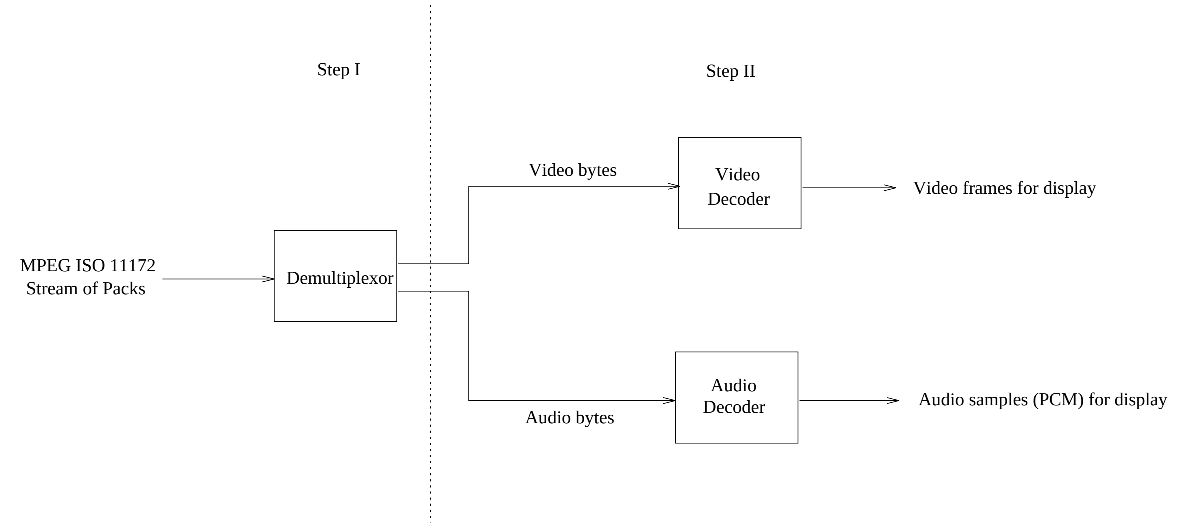

Figure 2: *MPEG Decoder System Configuration, referred to as STD (System Target Decoder)*

In a MPEG stream, whereas continuity is handled at the pack layer, synchronization is handled at the packet layer. We first present the pack and packet architectures, and then address continuity and synchronization. We will carry out all of the discussion in the context of MPEG-I media streams, mainly because, our experimental system is implemented using real-time MPEG-I hardware, with which we have direct practical experience; the more recent MPEG-II standard does not yet have commonly available hardware platforms. However, all of the main principles of media synchronization that we develop apply equally well to MPEG-II streams, whose main distinction from MPEG-I is in the compression layer; the only changes may be in the header formats of packs and packets in the media streams.

2 MPEG Pack Architecture

In MPEG streams, packs constitute a common system layer wrapped around a media-specific compression layer, and house the following functions at the encoder:

- Interleaving of multiple compressed streams into a single stream
- Time stamping: insertion of the System Clock Reference (SCR)

Term	*Definition*
STD	System Targed Decoder
SCR	System Clock Reference
PTS	Presentation Time Stamp
DTS	Decoding Time Stamp
DSM	Digital Storage Medium

Table 1: Terminology used in the paper

Pack header	*Bits*
start code	32
...	...
SCR	33
...	...
mux rate	22
...	...

Table 2: Pack header format

SCR's are sampled values (in Hz) of the encoder's clock (see Figure 1), which is constrained to meet the following drift bounds on its frequency [1]:

$$Nominal\ frequency,\ \mathcal{F} = 90,000Hz \tag{1}$$

$$Fractional\ frequency\ drift,\ \rho \leq 0.000005 \tag{2}$$

$$Frequency\ drift,\ \nu \leq 4.5Hz \tag{3}$$

$$Rate\ of\ change\ of\ drift,\ \delta\nu/\delta t \leq .00025Hz/s \tag{4}$$

Note that the bound on the absolute drift has to be satisfied in addition to that on the rate of change of drift; otherwise, at the rate of .00025 Hz/s, the frequency can go beyond the permitted 4.5 Hz drift bound over a period of about 5 hours. Hence, continuous operation of the encoder for longer than 5 hours may need resynchronization with an external master clock.

A SCR is inserted into each pack header (see Table 2). The SCR value in a pack is set to equal the reading of the encoder's clock at the instant the last byte of the SCR is constructed at the encoder. Successive SCRs cannot differ by more than 0.7 seconds to assure accurate clock updating at the STD, and hence, the time interval between successive packs cannot exceed 0.7 seconds. A pack header also contains the rate of the ISO 11172 multiplexed stream as manifested in the pack, and is termed the mux rate. The mux rate does not in itself equal the bit or byte rate, but is a scaled value, and is computed as follows by the encoder for a pack p leaving the encoder: Suppose that, following the SCR element in the

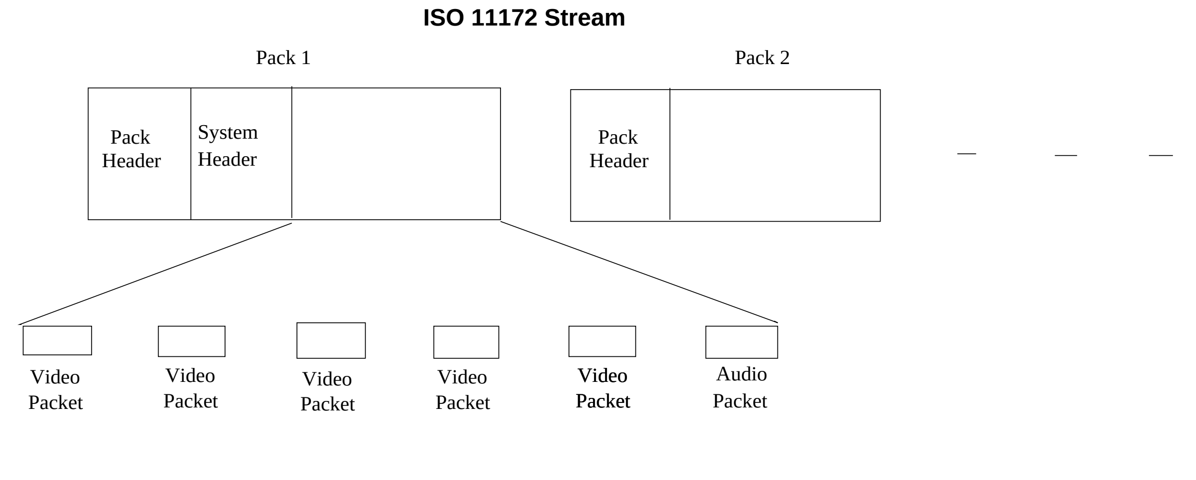

Figure 3: *Pack Architecture*

pack p are l_p bytes till the end of the pack, and the last byte of the pack is constructed by the encoder at time t_l. The mux rate is computed as follows:

$$\text{mux rate} = \frac{l_p}{50 \times (t_l - SCR)} \tag{5}$$

The mux rate may vary from pack to pack.

The first pack header usually contains additional system headers (see Figure 3). In these headers (see Table 3), the rate bound is the maximum value of the mux rate field coded in any pack of the stream. The video and audio bound are integers indicating the maximum number of the respective media streams simultaneously multiplexed (upto 16 for video, and upto 32 for audio). The video and audio lock flags indicate if there is a specified constant harmonic relationship between their respective media sampling rates and the SCR frequency.

The fixed flag indicates whether the stream is of fixed bitrate, in which case, SCR's of each pack in the multiplexed stream bear the following linear relationship with the size offset of the pack within the stream:

$$SCR(i) = \lfloor (a * i + b) \rfloor \ modulo \ 2^{33} \tag{6}$$

where a and b are real-valued constants, i is the index of the final byte of the SCR field counted from the start of the multiplexed ISO 11172 stream. The fixed flag may be set if the data rate averaged over the time intervals between successive SCRs is constant throughout the stream and equal to the value in the mux rate field. If, on the other hand, the mux rate fields indicate a rate higher than the average rate between successive SCRs, it implies that the stream is delivered in bursts at the peak rates indicated by the mux rate fields (the mux rate field can never be lower than that implied by the SCR fields). In order to transform a bursty stream into a constant bit rate stream, MPEG permits an encoder to insert

System header	*Bits (not in exact order)*
start code	32
...	...
rate bound	22
video bound	8
audio bound	6
video lock flag	1
audio lock flag	1
fixed flag	1
list of:	
stream id	8
buffer size bound	13

Table 3: Major contents of the system header, generally present in the first pack header

a padding stream. Such a padding stream would consist of a sequence of packets interleaved with audio and video packets within each pack to achieve a constant bit rate over the entire pack. At the STD, upon demultiplexing, the padding stream packets are simply discarded.

The system header also specifies video and audio buffer sizes that are needed at the STD. The sizes are specified in terms of a size and the scaling factor, the product of which yields the bound on the respective media's input buffering requirement.

3 Packet Architecture

A packet consists of a packet header followed by packet data (see Figure 4). The packet header begins with a 32-bit start code and an identifier indicating the media stream to which the packet belongs (see Table 4). The packet header may contain a decoding time stamp (DTS) and a presentation time stamp (PTS) corresponding to the first presentation media unit (e.g., video frame) that commences within the packet. Whereas the absolute time of presentation of the first media unit contained in a packet is indicated by the PTS field, the absolute time at which the decoding of the media unit is to be begun is indicated by the DTS field. In an idealized decoder, the decoding process is assumed to be instantaneous, and hence, the PTS and DTS may not differ at all. However, in real decoders, the difference between PTS and DTS represents the decoding computational overhead. These time stamps reside in the packet layer, since they are specific to individual media streams. The timestamps are represented as ticks counted on the 90 kHz system clock (modulo 2^{33}, of course), and therefore can time video playbacks of greater than 24 hours in length. In fact, the longest duration of a playback can be:

$$\frac{2^{33}}{90 \times 10^3 \times 3600} = 26\ hours,\ 30 minutes \tag{7}$$

If required, the PTSs can be transcoded into other presentation formats such as SMPTE for conventional video playback. There is no requirement that the PTS values are initialized to any particular value at the start of the stream. A PTS need not be provided for each media presentation unit, but the separation between successive PTSs should not exceed 0.7 seconds (mainly to limit the maximum accummulated asynchrony due to clock drifts between successive PTS arrivals at the decoder).

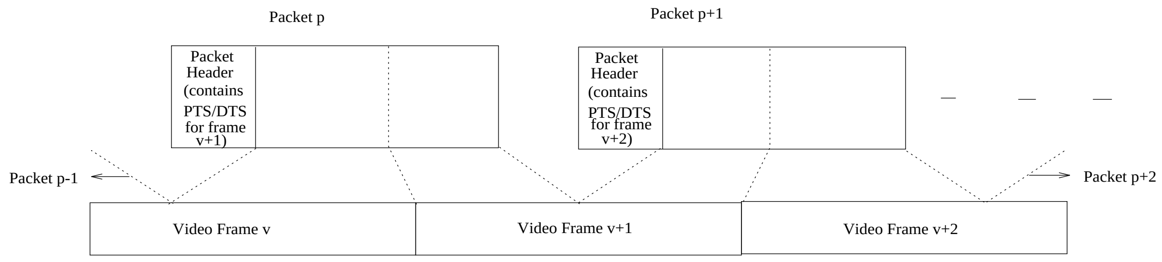

Figure 4: *Packet Architecture*

Packet header	*Bits*
Start code	24
Stream ID	8
Packet length	16
STD Buffer Scale	1
STD Buffer Size	13
PTS	33
DTS	33
...	...

Table 4: Packet header format

Packet headers also contain STD buffer size and scale fields, the product of which defines the maximum input buffer size needed at the STD for the packet. (The maximum of all the packet buffer sizes over the entire stream is placed in the system header - see Section 2). The input buffering delay, which is the difference in time between a byte entering the input buffer and when it is decoded, cannot exceed one second:

$$\forall bytes\ i \in presentation\ unit\ p:\ t_d(p) - t_a(i) \leq 1\ second \tag{8}$$

where, $t_d(p)$ is the decoding time of presentation unit p and $t_a(i)$ is the arrival time of byte i contained in presentation unit p. An encoder must construct its ISO 11172 multiplexed stream so that the input buffers at STD neither overflow nor underflow. Otherwise, there is wide latitude in how the multiplex is constructed (the size of packets and the relative placement of packets from different streams). In general, short packets require less STD buffering but more coding overhead than large packets. Other considerations, such as the need to align packets with the sectors on specific storage media, may influence the choice of packet length. Stuffing bytes can be inserted in packet headers for alignment.

An ISO 11172 multiplexed stream is a Constrained System Parameters Stream (CSPS) if both of the following conditions are met:

- The stream's buffer requirements at STD do not exceed 46×1024 bytes for video (assuming standard sized frames) and 4096 bytes for audio, and
- the stream conforms to the following linear bounded relationship between the packet rate and the $\mathsf{mux\ rate}$ (which is indicative of the bit rate):

$$Packet\ rate \leq \begin{cases} 300\ packets/s & \text{if } \mathsf{mux\ rate} \leq 10^5 \\ 300 \times \frac{\mathsf{mux\ rate}}{10^5}\ packets/s & \text{otherwise} \end{cases}$$

The data part of a packet contains a variable number of contiguous bytes from an individual media stream. Packets need not start at media presentation unit boundaries: hence, video frame start codes and audio sync words are not necessarily located at the start of packets, and there may be more than one media presentation unit commencing in a packet.

4 MPEG Stream Example

Consider a MPEG multiplexed stream with a 1.2 Mb/s video stream (yielded by encoding of standard sized video frames) and a 192 kb/s audio stream (representing encoded CD quality sterophonic sound). We can choose the length of both audio and video packets to be 188 bytes, a size recommended in MPEG-2, which, nevertheless, is also convenient for transmission on our local area network environment (including the Ethernet). One audio packet is interleaved with every 6 or 7 video packets to match the 6.25 ratio of video bit rate to audio bit rate. In particular, the interleave consists of one audio packet for every 6 video packets - repeated three times after which there are 7 video packets before the next audio packet, and this whole sequence repeats (see Figure 5).

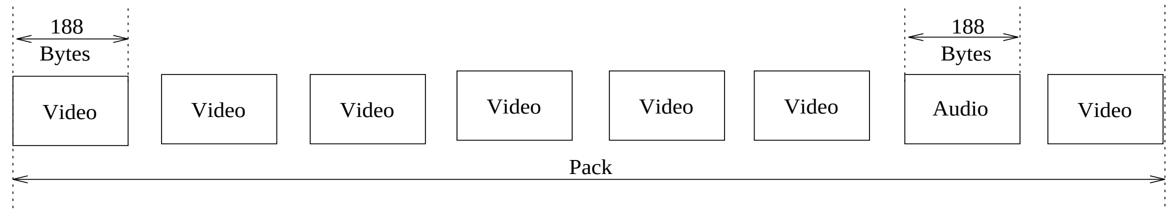

Figure 5: *MPEG ISO 11172 Stream Example*

In order to choose the pack size, observe that a smaller pack size enables more frequent transmission of system headers, consequently enabling a display site that comes up sometime after the start of a MPEG stream transmission to decode the MPEG stream. However, too small a pack size would impose unnecessary additional "packing" overheads. The upper bound on the pack size is the 0.7 second maximum separation between successive packs, since the SCRs extracted from the pack headers are constrained to be within 0.7 seconds of each other for periodic clock updating at the decoder. We chose the pack size to be eight packets, which is close to a comfortable 10 ms separation between successive packs, but there is indeed quite a bit of flexibility in this choice. At this pack size, pack headers account for a low 0.8% of the total bitrate.

The video presentation unit is a frame, and the audio presentation unit is chosen to be 1584 bytes (66 ms long, which is close to the upper limit on the acceptable audio latency).

In our example, the first pack has a SCR equal to $4,008$ (ticks, measured on a 90,000 Hz clock). The first video frame has a DTS equal to 22,012. The difference between the two is the video decoder set up time at startup:

$$Startup\ delay = \frac{22,012 - 4,008}{90,000} seconds = 200ms \tag{9}$$

Thus, the first I picture is decoded 200 ms after arrival. The PTS of the first video frame is 24,000, which is an extra 20 ms after decoding but before displaying the first I frame: this represents the time needed to initialize the video display buffer. The example can be extended to the audio stream in a similar fashion.

5 Playback Continuity

MPEG is mainly targeted towards stored media applications (as opposed to live video tele-conferencing). In such applications, the video (and its associated audio) is captured, encoded, and stored on a *Digital Storage Medium (DSM)* at the time of content creation, and later retrieved on demand for playback, at which time, the DSM transmits the MPEG ISO 11172 pack stream to display sites. At a display site, the system target decoder (STD) receives the packs and enqueues them in an input buffer for demultiplexing, decoding and display. In an idealized STD, all the packs containing the media presentation unit that has been in the input buffer longest are demultiplexed and decoded instantaneously.

After being decoded, video (but not audio) presentation units (i.e., frames) may be delayed in reorder buffers before being presented to the viewer at the output of the STD. This is because, during encoding, MPEG transforms video frames into a sequence of *Intracoded (I)*, *Predictive-coded (P)*, and *Bidirectionally-coded (B)* frames, producing a sequence such as follows:

$$\mathbf{I_1}\ \mathsf{B_2}\ \mathsf{B_3}\ \mathsf{B_4}\ P_5\ \mathsf{B_6}\ \mathsf{B_7}\ \mathsf{B_8}\ P_9\ \mathsf{B_{10}}\ \mathsf{B_{11}}\ \mathsf{B_{12}}\ \mathbf{I_{13}} \tag{10}$$

The above is just an example; the number of B frames between I and P, and the number of P frames between two I frames, are all application dependent. The point to observe is that, a B frame is bidirectionally encoded from both its preceding I/P and its succeeding I/P frame; hence, at the time of decoding, the B frame would need, not only its preceding I/P, but also its succeeding I/P. Thus, the MPEG encoder places the succeeding I/P prior to the B in the ISO 11172 stream, as a consequence of which, the above Sequence (10) would appear in the encoded ISO 11172 stream as follows:

$$\mathbf{I_1}\ P_5\ \mathsf{B_2}\ \mathsf{B_3}\ \mathsf{B_4}\ P_9\ \mathsf{B_6}\ \mathsf{B_7}\ \mathsf{B_8}\ \mathbf{I_{13}}\ \mathsf{B_{10}}\ \mathsf{B_{11}}\ \mathsf{B_{12}} \tag{11}$$

During decoding, the P_5 is decoded before B_2, B_3, and B_4. P_9 is decoded before B_6, B_7, and B_8. $\mathbf{I_{13}}$ is decoded before $\mathsf{B_{10}}$, $\mathsf{B_{11}}$, and $\mathsf{B_{12}}$. These would have to be reordered back into the original Sequence (10). This resequencing is done in the display reorder buffers immediately after decoding. In particular, an I-picture or a P-picture decoded before one or more B-pictures must be delayed in the reorder buffer. It should be delayed until the next I-picture or P-picture is decoded. Thus, the decoding time and the presentation times differ by integral number of picture periods for these reordered frames. For all other frames, the two are assumed to be identical, i.e., the decoding is assumed to be instantaneous, even though, real decoders generally introduce post-processing and presentation delays. However, MPEG specifies that these decoding delays in real decoders should not be compensated by real encoders. Real encoders must generate bitstreams that play correctly on idealized STD. The delays that occur in any specific real decoder must be compensated in that decoder, not the encoder.

For continuous playback of media streams in a real decoder, it is essential that media units be available at the decoder prior to their respective decoding times, and that the rate of consumption of packs at the STD match the rate of transmission by the DSM. When the DSM transmits the first pack of a MPEG media stream, the DSM sets its clock to equal the SCR_1 contained in the first pack. When the first pack reaches the STD, STD also sets its clock to SCR_1. However, owing to non-zero delays in real transmission channels, the clocks of DSM and STD may differ (see Figure 6):

$$T_{DSM} - T_{STD} = \Delta \in [\Delta_{max}, \Delta_{min}] \tag{12}$$

where, Δ_{max} and Δ_{min} are bounds on the transmission delay (possibly introduced by the network) between the DSM and the STD [5].

If at any time, the STD buffer suffers an underflow, discontinuity in playback may occur. The worst case scenario for buffer underflow is when an earlier pack arrives after minimum delay and a later pack arrives after maximum delay. Suppose that the first pack arrives after a minimum delay. Using Equation (12):

$$T_{DSM} - T_{STD} = \Delta_{min} \tag{13}$$

Further suppose that a later pack p suffers maximum delay. This pack would have been transmitted by the DSM when the DSM's clock shows $SCR(p)$. At the same time, the STD clock would show $SCR(p) - \Delta min$ (from Equation (13)). Adding a transmission delay of Δ_{max}, the arrival time of pack p is given by $SCR(p) + \Delta_{max} - \Delta_{min}$. However, the decoding and presentation time schedule: DTS and

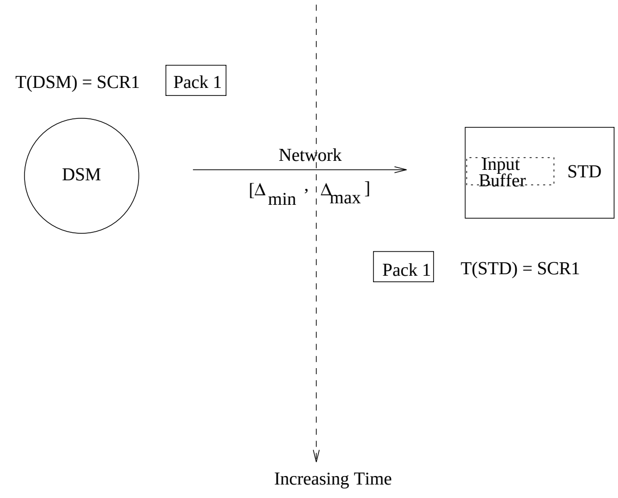

Figure 6: *Lag between DSM and STD clocks*

PTS encoded in the pack headers assume an on-time arrival of pack p: that is, each pack must arrive at the STD on the precise schedule specified in the pack's SCR, since, only then are the STD buffers assured not to underflow during decoding. Hence, in order to prevent underflow, the STD delays setting of its clock to the first pack's SCR by $\Delta_{max} - \Delta_{min}$ after the arrival of the first pack (see Figure 7). Since this additional jitter factor is specific to the STD set up and possibly unknown to the encoder (that may be operating in a different system environment), it cannot be automatically factored into the computation of SCR by the encoder.

With this modification, the problem of underflow gets solved. However, it compounds another problem: that of overflow. In order to prevent overflow, how much additional buffer might be required? A casual reader may be inclined to conclude that, an additional buffer capable of accommodating the maximum number of media units whose decoding times can differ by at most $\Delta_{max} - \Delta min$ should be sufficient. Surprisingly, this is not true. In fact, the problem is much more. To see why, observe that buffer overflow occurs if the first pack suffers maximum delay, and a later pack p suffers minimum delay, in their respective journeys from the DSM to the STD (this is the reverse of the worst-case scenario for buffer underflow). When the first pack arrives at the STD, the STD clock's effective time is:

$$Arrival\ time\ of\ first\ pack\ = SCR(1) - (\Delta_{max} - \Delta_{min}) \tag{14}$$

since, with the modification of the last paragraph, the STD clock is set to $SCR(1)$ after $\Delta_{max} - \Delta_{min}$ time units past the arrival of the first pack. The later pack p suffers only Δ_{min} in comparison with the maximum delay that the first pack suffers, hence, its arrival time is:

$$\begin{aligned} Arrival\ time\ of\ pack\ p &= SCR(p) - (\Delta_{max} - \Delta_{min}) - (\Delta_{max} - \Delta_{min}) && (15) \\ &= SCR(p) - 2 \times (\Delta_{max} - \Delta_{min}) && (16) \end{aligned}$$

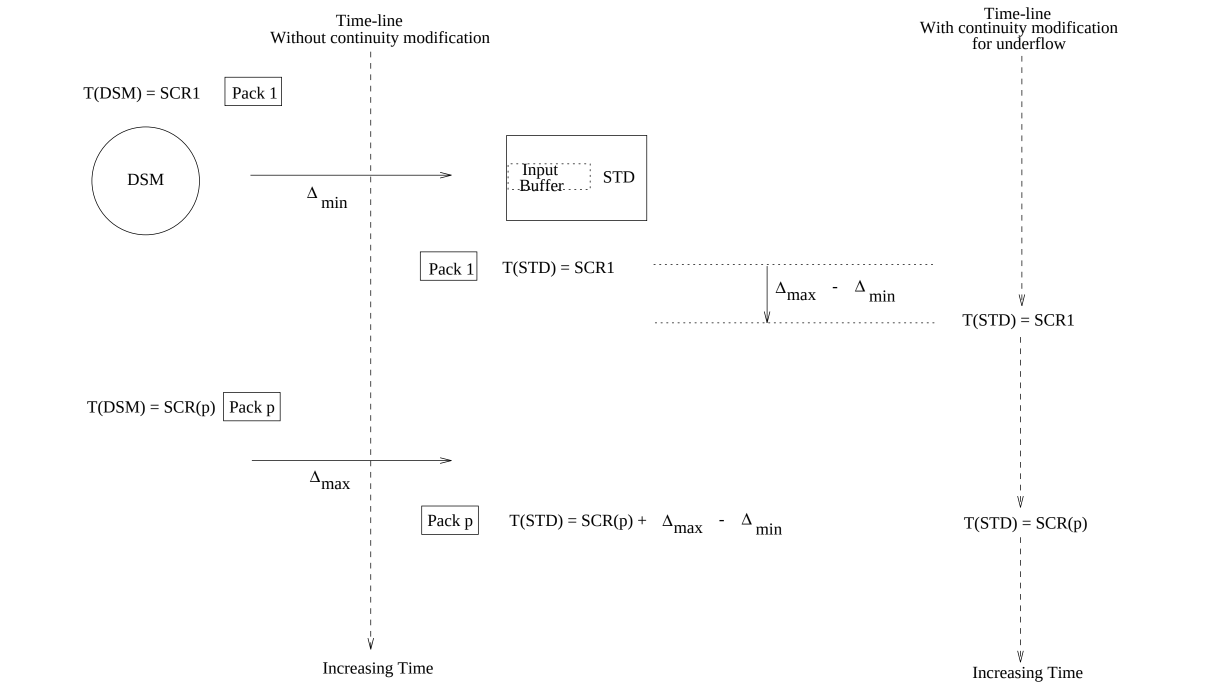

Figure 7: *Delayed setting of STD clock for preventing playback discontinuity due to buffer underflow*

However, the encoder specified buffer sizes are sufficient to account for on-time arrival of each pack. Early arrival of a pack, as given by 16, requires additional buffering,[3] equal to the maximum number of additional bytes that might arrive in an interval of $2 \times (\Delta_{max} - \Delta_{min})$; see Figure 8.

6 Synchronization

Synchronization in MPEG is handled at the packet layer, with the SCR, PTS, and DTS fields serving as instruments. After the incoming packs are demultiplexed into individual media packets in the input queue of the STD, the media packets are assembled into media presentation units, which are then forwarded to their respective media decoders. The decoders parse the presentation units to extract the PTS and DTS fields. It must be remembered that PTS and DTS fields are not necessarily encoded for each video frame or audio presentation unit, but are only required to occur with intervals not exceeding 0.7 seconds for periodic updating of the decoders' clocks. Whereas the DTSs specify the time at which all the bytes of a media presentation unit are removed from the buffers for decoding, the PTSs specify the actual time at which the presentation units are displayed to the user.

The STD model assumes instantaneous decoding of media presentation units. For audio units and B frames of video, the decoding time is the same as the presentation time, and so only their PTSs are listed

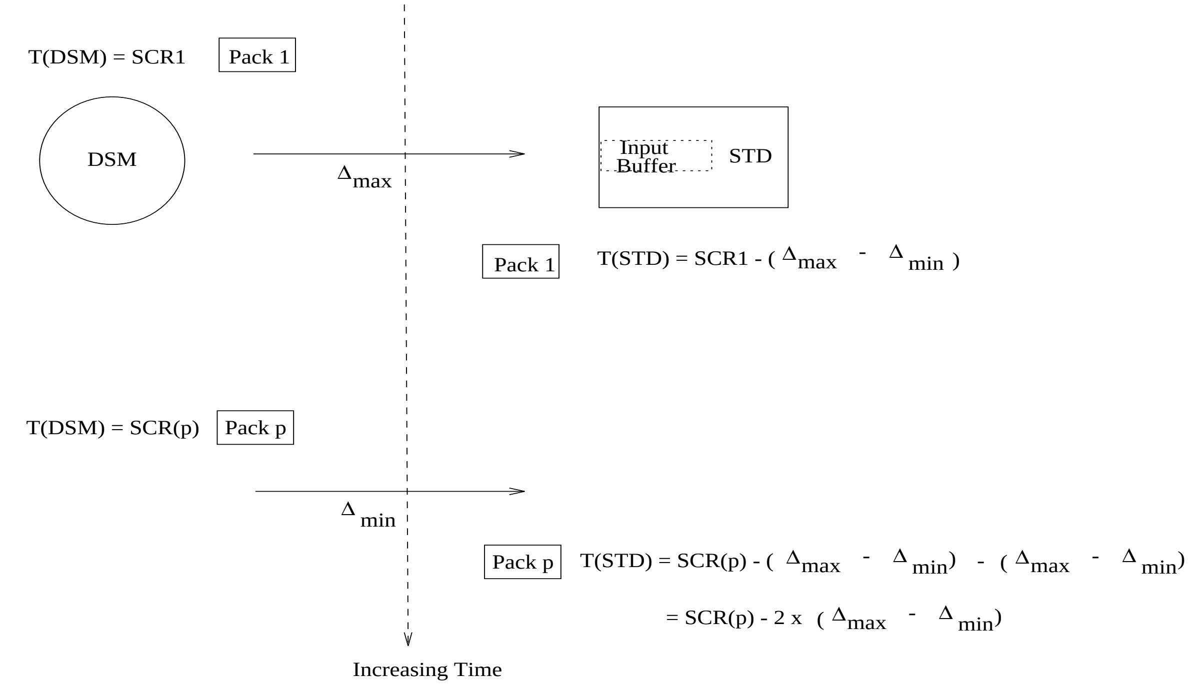

Figure 8: *Buffer overflow scenario during playback*

in their respective packet headers. On the other hand, I and P frames of video have a reordering delay (since their transmission and decoding would have preceded earlier B frames - see Section 5) between their decoding and presentation, and hence, their PTS and DTS values differ by some integral number of frame periods (equal to the display time of the earlier B frames).

After the PTS and DTS are extracted from a media presentation unit, the data bytes are routed for decoding and display. The display of different media units must proceed in a mutually synchronized manner. The synchronization can be driven by either one of the media streams serving as the master, or by the DSM. Both of these approaches are elaborated next.

6.1 Synchronization using A Master Stream

All of the media streams being decoded and displayed must have exactly one independent master [4]. Each of the individual media display units must slave the timing of their operation to the master stream. The master stream may be chosen depending on the application. Whichever media stream is the master, all the media streams but the master must slave the timing of their respective displays to the PTSs extracted from the master media stream.

As an illustration, suppose that the audio stream is chosen to be the master; the audio playback will drive the progression of playback of all the streams. The audio stream will be played back continuously,

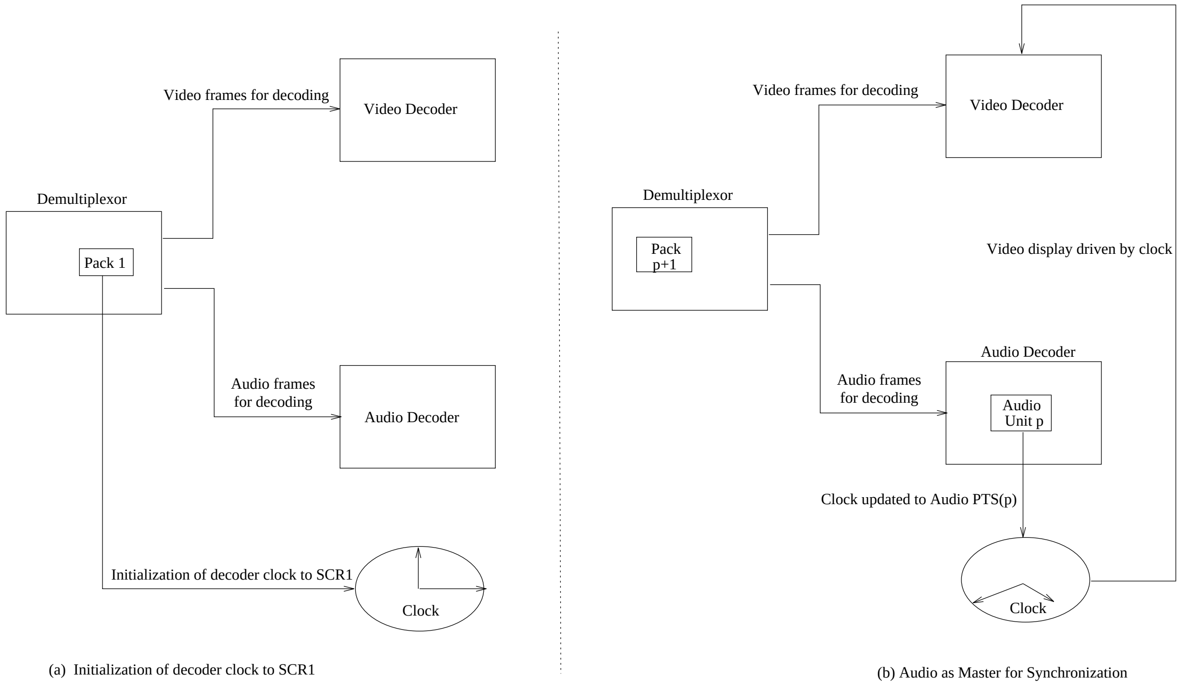

Figure 9: *Synchronization using audio as the master*

with the clock being continually updated to equal the PTS value of the audio unit being presented for display. In particular, the STD clock is typically initialized to be equal to the value encoded in the first SCR field when that field enters the decoder's buffer. Thereafter the audio decoder controls the STD clock. As the audio decoder decodes audio presentation units and displays them, it finds PTS fields associated with those audio presentation units. At the beginning of display of each presentation unit, the associated PTS field contains the correct value of the decoder's clock in an idealized decoder following the STD model. The audio decoder uses this value to update the clock immediately (see Figure 9).

The other decoders simply use the audio-controlled clock to determine the correct time to present their decoded data, at the times when their PTS fields are equal to the current value of the clock. Thus, video units are presented when the STD clock reaches their respective PTS values, but the clock is never derived from a video PTS value. Thus, if the video decoder lags for any reason, it may be forced to skip presentation of some video frames; on the other hand, if the video decoder leads, it may be forced to pause; but the audio is never skipped or paused - it will always proceed at its natural rate since the audio has been chosen to be the master.

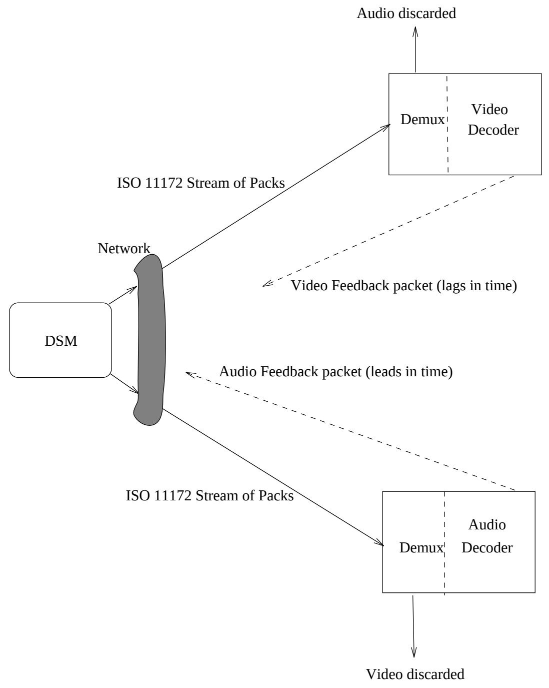

Figure 10: *Synchronization using feedback in distributed MPEG playback*

6.2 Synchronization in Distributed Playback

If the audio and video decoders are physically at different locations on the network, which is not unlikely in future integrated networks that provide a common interconnection between servers, video displays and audio speakers, neither the audio nor the video stream can be used as the master. Instead, the DSM serves as the time master, with the audio and video decoders implemented as separate decoder subsystems, each receiving the complete multiplexed stream. Each decoder parses the received multiplexed stream and extracts the system layer information and the coded data needed by that decoder. The decoder then determines the correct time to start decoding by comparing the DTS field extracted from the stream with the current value of its STD clock.

In the idealized STD, audio and video decoding is assumed to be instantaneous. Real decoders, however, may experience non-zero decoding delays; furthermore, the audio and video decoding delays may be different, causing them to go out of synchrony. Over a period of one second (i.e., 90,000 cycles), errors of 50 parts per million can lead to PTS values differing from the nominal values by 4 or 5, which accumulates over time. In order to maintain proper synchronization, the DSM must track the real-time progression of playback at the decoders, for which purpose, the decoders transmit feedback messages to the DSM (see Figure 10). A feedback message is a light-weight packet that is transmitted concurrently with the display of a media unit, and contains the PTS of that media unit. When a feedback message from a decoder arrives at the DSM, the DSM extracts the PTS contained in the feedback. PTSs extracted from feedbacks of different decoders, when compared, reveal the asynchrony if any. The DSM can then take corrective action - for instance, by muting the leading stream until the lagging stream catches up [2].

We have developed and implemented this feedback technique for media synchronization in our UCSD Multimedia Laboratory. The laboratory hardware environment consists of four 486-based personal computers connected by both Ethernet and FDDI to a multi-gigabyte storage server. The personal computers are all equipped with Sigma Design's Reel Magic MPEG decoder cards, which can decompress and display MPEG streams in real time at 30 frames/s. The storage server is equipped with Optibase MPEG encoder cards, which can capture and compress NTSC quality video into MPEG (also in real time, at 30 frames/s). We have omitted a detailed description of the feedback techniques themselves and their peformance evaluation in this paper; for an elaborate treatment of these topics, the reader is kindly referred to a companion paper [3].

7 Concluding Remarks

MPEG is the emerging standard for multimedia retrieval and playback. It consists of an encoding layer around which is wrapped a system layer. Whereas the encoding layer handles compression and decompression, the system layer handles streaming, continuity, and synchronization. The system layer packetizes compressed media units, interleaves the packets of different media and organizes them into a stream of packs. The pack is the basic unit for maintaining continuity of playback, via a system clock reference (SCR) time stamp that is inserted in the pack header. The packet is the basic unit for maintaining synchronization between media playback, via decoding and presentation time stamps (DTS and PTS) inserted in the packet headers. Whereas MPEG provides the framework for insertion, transmission and extraction of these time stamps, additional feedback protocols are used to make use of these time stamps in the enforcement of synchronization, particularly in a distributed multimedia environment.

References

[1] WG 11. MPEG I: Coded Representation of Picture, Audio and Multimedia/Hypermedia Information. *ISO/IEC JTC 1/SC 29*, November 1991.

[2] Srinivas Ramanathan and P. Venkat Rangan. Adaptive Feedback Techniques for Synchronized Multimedia Retrieval over Integrated Networks. *IEEE/ACM Transactions on Networking*, 1(2):246–260, April 1993.

[3] P. Venkat Rangan, Srinivas Ramanathan, and Srihari SampathKumar. Feedback Techniques for Continuity and Synchronization in Multimedia Information Retrieval. *ACM Transactions on Information Systems*, April 1995.

[4] P. Venkat Rangan, Srinivas Ramanathan, Harrick M. Vin, and Thomas Kaeppner. Techniques for Media Synchronization in Network File Systems. *Computer Communications*, 16(3):168–176, March 1993.

[5] P. Venkat Rangan, Harrick M. Vin, and Srinivas Ramanathan. Designing an On-Demand Multimedia Service. *IEEE Communications Magazine*, 30(7):56–65, July 1992.

[6] R. Steinmetz. Synchronization Properties in Multimedia Systems. *IEEE Journal on Selected Areas in Communications*, 8(3):401–412, April 1990.

[7] Ralf Steinmetz. Multimedia Encoding Standards. *ACM/Springer Multimedia Systems*, 1(5), 1994.