Special Paper
방송공학회논문지 제24권 제7호, 2019년 12월 (JBE Vol. 24, No. 7, December 2019)
https://doi.org/10.5909/JBE.2019.24.7.1259
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

Advanced Delivery Timing Model Design for MPEG MMT Protocol

A-young Kim[a)], Eun-bin An[a)], and Kwang-deok Seo[a)‡]

Abstract

Maintaining timing relationships among packets in a single media stream or between packets from different media streams is an essential criterion in MMT system. It is the function of the synchronization and de-jittering algorithms to re-adjust timing relationship between the MMT packets to assure synchronized playback. Thus, delivery of time constrained MPEG media on time, according to their temporal requirements, is an important goal of MMT. For this purpose MMT needs to specify syntax and semantics of a timing model to be used by the delivery functions. In this paper, we propose a proper timestamp-related header format for MMT delivery timing model to support media synchronization in various delivery scenarios including hybrid delivery.

Keywords : delivery timing model, media delivery, MPEG media transport (MMT) protocol

I. Introduction

With the explosive growth of the Internet, IP (Internet Protocol)-based point-to-point communications via broadband access have become prevalent in many industrial countries and are expected to become another important mode of communications for multimedia. On the other hand, television broadcasting via terrestrial and satellite transmission channels has been the main means for multimedia transport for quite some time and is still an important mode of multimedia delivery. Thus it is believed that a hybrid delivery environment combining various modes of communications would enable advanced and sophisticated multimedia services [1].

The MPEG has recently developed a new standard, MPEG media transport (MMT), for the next generation hybrid media delivery service over IP networks considering the emerging convergence of digital broadcast and broadband services [2]. As an example of a hybrid content delivery environment, multimedia contents will be offered to both broadcasting channels and point-to-point communication channels based on IP networks. Some components, such as coded audio signals and coded video signals, are transported from different sources to consumer devices via different channels in the hybrid environment [3], [4]. As an example, consumer devices need to present these com-

a) Division of Computer and Telecommunications Engineering, Yonsei University
‡ Corresponding Author : Kwang-deok Seo
E-mail: kdseo@yonsei.ac.kr
Tel: +82-33-760-2788
ORCID: http://orcid.org/0000-0001-5823-2857
※ This research was supported by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A01058873).
・Manuscript received October 14, 2019; Revised December 12, 2019; Accepted December 12, 2019.

ponents in a synchronized manner.

In this paper, we propose a proper timestamp-related header format for MMT delivery timing model to support media synchronization in various delivery scenarios including hybrid delivery. By exploiting the delivery timing model, it is possible to maintain timing relationships among packets in a single media stream or between packets from different media streams [5], [6].

Ⅱ. Proposed Delivery Timing Model Design

1. Identified delivery timing information

One of the important roles of the delivery layer (D-layer) of MMT is to provide timing information generated in creating MMT packets to the receiver side for media synchronization [7]. The procedure for the creation of the MMT packets includes delivery layer packetization process. Firstly, we identify major timing parameters generated in the delivery layer packetization process. Figure 1 shows the proposed overall timing model that can be considered for the MMT system. There are five important timing instants in the proposed timing model, such as Sampling_Time, Delivery_Time, Arrival_Time, Decoding_Time, and Rendering_ Time. Sampling_Time reflects the sampling instant of the input frame to the media encoder. The sampling instant is derived from a clock that increments monotonically and linearly in time to allow synchronization. Considering that the RTP timestamp and DTS/PTS of MPEG-2 system commonly employ 90 kHz clock resolution for their representation, a 33-bit 90 kHz timestamp can be used to represent the Sampling_Time. Delivery_Time corresponds to the delivery instant of the packetized MMT packet after sender processing delay which is the elapsed time to prepare the deliverable MMT packet from Sampling_Time. Arrival_Time represents the arrival instant of the transmitted MMT packet at the receiver side after network transmission delay. Decoding_Time corresponds to the decoding instant of the recovered compressed bitstream data for timely decoding after experiencing receiver processing delay induced by MMT delivery layer depacketization, encapsulation layer (E-layer) decapsulation, and decoder buffering delay and so on. Rendering_Time designates the presentation or composition time of the reconstructed media data occurring after rendering time offset which corresponds to the time spent in a rendering buffer for frame reordering of I-, P-, B- pictures before presentation to the output devices. Rendering_Time_Offset is the difference between the Decoding_Time and the actual presentation.

Among the above five timing instants and timing parameters, Sampling_Time and Rendering_Time_Offset are determined during the media encoding and encapsulation stage performed before actual packet delivery processing including packetization, and thus can be within the scope of the encapsulation layer timing model. On the contrary,

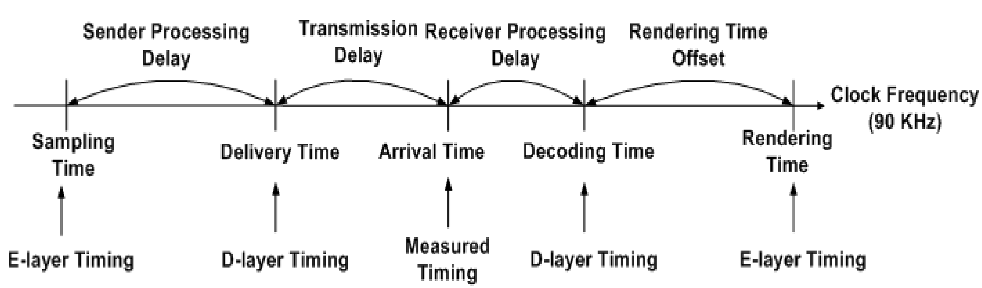


Fig. 1. Flow for hybrid delivery scenario

Delivery_Time, Arrival_Time, and Decoding_Time are affected by the Sender_Processing_Delay, Transmission_Delay, and Receiver_Processing_Delay parameters, and thus out of the scope of encapsulation layer layer timing model, but rather related to the delivery timing model. Therefore, we identify the Delivery_Time, Arrival_Time, and Decoding_Time as the proper timing information related to the delivery timing model.

Based on the overall timing model as shown in Figure 1, Figure 2 and Figure 3 show the timing model diagram at the MMT sender and receiver sides, respectively. Data flows at a constant (or specified) rate to the encoder and the output of the encoder is at a variable rate to the encoder buffer. The constant rate output of the transmission buffer transmitted via IP networks after MMT packet packing procedure including encapsulation and delivery layer packetization processes, and into the MMT packet de-packing process after passing through the receiver buffer and then into the decoder buffer. The variable rate output of the decoder buffer is fed into the decoder to pass a constant (or specified) rate output to the decoder, then to the rendering buffer if necessary. By comparing Figure 1 with Figure 2 and Figure3, we can notice the specific time positions of the five timing instants as well as the specific time durations of the timing parameters.

2. Required syntax and semantics

The syntax format and the corresponding semantics for the delivery timing information of MMT delivery layer are listed in Table 1.

Ⅲ. Usage of the Proposed Timing Instants and Parameters

Sampling_Time (TSam) reflects the sampling instant of the input frame to the media encoder and it is carried in the encapsulation layer header as encapsulation layer timing information. Starting form this time instance, the timing

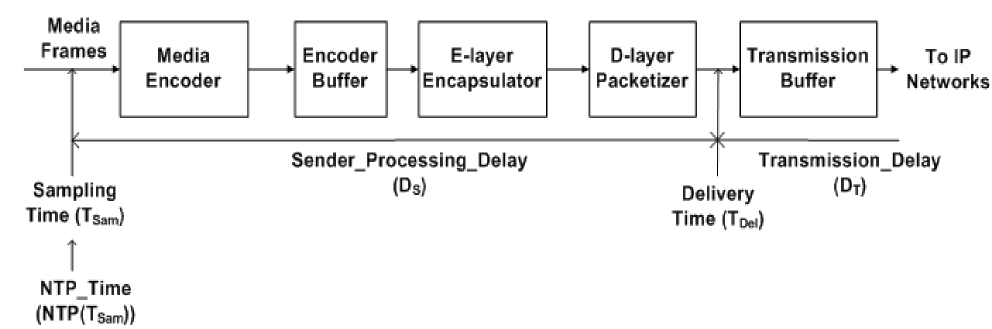


Fig. 2. Timing model diagram at the MMT sender side

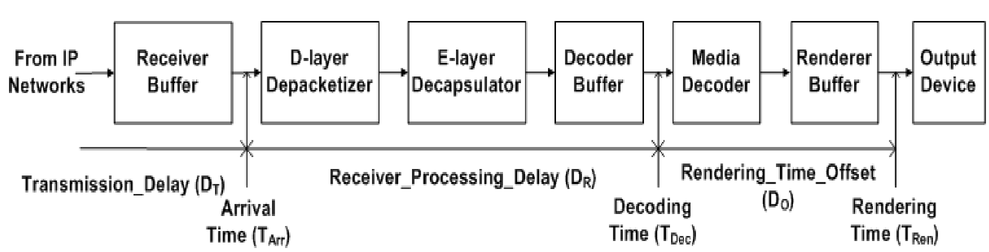


Fig. 3. Timing model diagram at the MMT receiver side

Table 1. Syntax format and semantics for the delivery timing information

Syntax	Bits	Semantics
NTP_Time (NTP(T_{Sam}))	32, 48 or 64	Indicates a UTC time in NTP format corresponding to Sampling_Time of E-layer Timing model. The full resolution NTP timestamp is a 64-bit unsigned fixed-point number with the integer part of 32 bits and fraction part of 32 bits. This NTP timestamp can be truncated down to 48 bits, 16 bits for seconds and 32 bits for fractions, or can also be truncated down to 32 bits, 16 bits for seconds and 16 bits for fractions, according to required timestamp precision for accurate enough media synchronization. *Note that this NTP_Time field can also be represented in the E-layer timing model. In that case, we don't need to include this field in the D-layer timing model as described above.*
Sender_Processing_Delay (D_S)	20	The time elapsed since media frame data enters into the media encoder till an MMT packet ready for delivery is generated. This value is represented by 90kHz clock resolution.

instants and parameters shown in Figure 2 and Figure 3 have the following relationships:

$$\begin{aligned} T_{Del} &= T_{Sam} + D_S \\ T_{Arr} &= T_{Sam} + D_S + D_T \\ T_{Dec} &= T_{Sam} + D_S + D_T + D_R \cdot \end{aligned} \quad (1)$$

In addition to the above time instants, Rendering_Time (TRen) designates the presentation or composition time of the reconstructed media data occurring after rendering time offset which corresponds to the time spent in a rendering buffer as shown in Figure 3. This Rendering_Time is determined by the Rendering_Time_Offset parameter which is carried in the encapsulation layer header. Thus, among the five time instants, the three time instants shown in Eq. (1) are relevant to the scope of the delivery timing model. In order to provide continuous playback in a synchronous manner between different media streams, the total end-to-end delay occurring in the MMT system should be maintained as a constant value for all the media streams. Thus the following relationship should be observed.

$$D_S + D_T + D_R = D_{Tot} \quad (2)$$

The DTot consisting of Sender_Processing_Delay, Transmission_Delay and Receiver_Processing_Delay should be kept constant for continuous decoding and playback at the receiver. Thus, a constant quantity equal to a proper total end-to-end delay is added, creating decoding time information as shown in Eq. (1). The DTot can be determined at the receiver considering general end-to-end transmission delay and initial latency to be experienced by the client. It could also be exchanged a priori between the receiver and the sender considering a suggested initial playback delay, for example by C.1 layer signaling, before actual starting of the MMT data transmission. It could even be adaptively adjusted based on the actual packet reception delay experienced by the receiving party. This is the reason, unlike DTS of MPEG-2 TS, why RTP which is usually used for packet transmission over IP networks does not specify any definite decoding time information its header field. The sender transmits NTP(TSam) and DS as delivery timing information by attaching the two values in the delivery layer header of MMT packet. With the provided

NTP(TSam) and DS values, UTC time in NTP format corresponding to the Delivery_Time (TDel) can be recovered at the receiver as follows:

$$NTP(T_{Del}) = NTP(T_{Sam}) + D_S / 90{,}000 \quad (3)$$

The Arrival Time (TArr) can be measured after the MMT packet has arrived at the receiver terminal. If the UTC time of TArr expressed in NTP format is denoted as NTP(TArr), the transmission delay DT can be obtained as follows:

$$D_T = \left(NTP(T_{Arr}) - NTP(T_{Del})\right) \times 90{,}000 \quad (4)$$

From the DS and DT values, the appropriate Receiver_ Processing_Delay (DR) is given by

$$D_R = D_{Tot} - \left(D_S + D_T\right) \quad (5)$$

Start_Clock_Offset can be derived by multiplying the CF (clock frequency) that was used to obtain the timestamp of the MMT data to the Start_Time_Offset as shown in the following Equation.

$$Start\_Clock\_Offset = Start\_Time\_Offset \times CF \quad (6)$$

With the Start_Clock_Offset count, it is possible to derive the exact timestamp (TS) corresponding to the ABSsyn which is marked in the MMT header. The MMT packet with timestamp (TSk) is Start_Clock_Offset counts away (corresponding to the Start_Time_Offset seconds) from the first MMT packet which has timestamp of TS0. The equation for obtaining TSk is shown in Equation 7.

$$TS_k = TS_0 + Start\_Clock\_Offset \quad (7)$$

With this DR value, the receiver can identify the exact time duration that the compressed media data (i.e. Access Unit) should spend in the decoder buffer before it is decoded. By observing this process, the delivered MMT packet can be decoded at exact Decoding_Time instant, thus providing perfect media synchronization between several media sources even delivered by different servers via different networks or channels. Note that RTP does not give any information on the amount of buffering that may be needed at the receiver, or the decoding time of the packets.

In (9), The timing information DS value can also be used to more accurately measure the amount of packet arrival jitter. In RTP, the interarrival jitter is defined to be the mean deviation of the difference in the packet spacing at the receiver compared to the sender for a pair of packets. Because the jitter calculation is based on the RTP timestamp which represents the instant when the first data in the packet was sampled, any variation in the delay between that sampling instant and the time the packet is transmitted will affect the resulting jitter that is calculated. Such a variation in delay would occur for audio packets of varying duration. It will also occur for video encodings because the timestamp is the same for all the packets of one frame but those packets are not all transmitted at the same time. The variation in delay until transmission does reduce the accuracy of the jitter calculation as a measure of the behavior of the network by itself. Unlike the conventional case of jitter calculation, in the proposed timing model, Delivery_ Time corresponding to the exact delivery instant is used for interarrival jitter calculation. This may result in more accurate estimation of the required de-jitter buffering time which corresponds to the sufficient size of the receiver buffer in Figure 3. Note that the proposed delivery timing mechanism can be applied to both GOP structure of IPPPP and IBBPBBP. The purpose of such kind of de-jitter buffering is to recover the temporal relationship between MMT packets arriving with different timing at the buffer output. Under this approach, we can adequately remove jitter induced by IP networks and forward the de-jittered MMT packets to the MMT receiver processing side. The detailed procedure of how the receiver client does the jitter calculation would be out of the scope of the MPEG standard.

Ⅳ. Conclusions

In this paper, we identified relevant delivery timing parameters for MMT and proposed a proper timestamp-related header format for MMT delivery timing model to support media synchronization in various delivery scenarios including hybrid delivery.

References

[1] S. Aoki, Y. Kawamura, K. Otsuki, N. Nakamura, and T. Kimura, "Development of MMT-based broadcasting system for hybrid delivery," IEEE Int. Con. Multimedia and Expo Workshops, 2015, pp. 1-4.

[2] ISO/IEC JTC1/SC29/WG 11 23008-1: High efficiency coding and media delivery in heterogeneous environments – Part 1: MPEG Media Transport (MMT), Sep. 2013, Vienna, Austria.

[3] Y. Kawamura, K. Otsuki, A. Hashimoto, and Y. Endo, "Functional evaluation of hybrid content delivery using MPEG media transport," IEEE Int. Conf. Consumer Electronics, Jan. 2016, pp. 239-240.

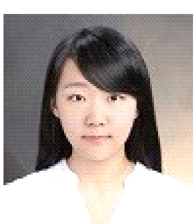
[4] M. Deventer, H. Stokking, M. Hammond, J. Feuvre, P. Cesar, "Standards for multi-stream and multi-device media synchronization," IEEE Commun. Magazine, vol. 54, no. 3, pp. 16-21, Mar. 2016.

[5] ISO/IEC JTC1/SC29/WG 11 N11542: "Use Cases for MPEG Media Transport (MMT)," July 2010, Geneva, Switzerland.

[6] ISO/IEC JTC1/SC29/WG 11 N11540: "Requirements on MPEG Media Transport (MMT)," July 2010, Geneva, Switzerland.

[7] K. Seo, T. Jung, J. Yoo, C. Kim, and J. Hong, "A new timing model design for MPEG media transport (MMT), IEEE Int. Symp. Broadband Multimedia Systems and Broadcasting, 2012, pp. 1-5.

저 자 소 개



A-young Kim

- Feb. 2016 : B.S. degree, Division of Computer and Telecommunications, Yonsei Univeristy
- Mar. 2016 ~ currently : Ph.D. Candidate, Division of Computer and Telecommunications, Yonsei Univeristy
- ORCID : https://orcid.org/0000-0002-3793-1365
- Research interests : Visual communication, Real-time streaming, 360/VR video

Eun-bin An

- Aug. 2016 : B.S. degree, Division of Computer and Telecommunications, Yonsei Univeristy
- Mar. 2017 ~ currently : Ph.D. Candidate, Division of Computer and Telecommunications, Yonsei Univeristy
- ORCID : https://orcid.org/0000-0001-7681-4682
- Research interests : Visual communication, Real-time streaming, 360/VR video

저 자 소 개

Kwang-deok Seo

- Feb. 1996 : B.S., Department of Electrical Engineering, KAIST
- Feb. 1998 : M.S., Department of Electrical Engineering, KAIST
- Aug. 2002 : Ph.D., Department of Electrical Engineering, KAIST
- Aug. 2002 ~ Feb. 2005 : Senior research engineer, LG Electronics
- Sep. 2012 ~ Aug. 2013 : Courtesy Professor, Univ. of Florida, USA
- Mar. 2005 ~ currently : Professor, Yonsei University
- ORCID : http://orcid.org/0000-0001-5823-2857
- Research interests : Video coding, Visual communication, digital broadcasting, multimedia communication system