A PRECISE AUDIO/VIDEO SYNCHRONIZATION SCHEME FOR MULTIMEDIA STREAMING

*Won Sup Chi, **Soon-heung Jung, **JeongJu Yoo, and *Kwang-deok Seo

*Computer and Telecommunications Engineering Div., Yonsei Univ., Gangwon, Korea
**Convergence Media Research Team, ETRI, Daejeon, Korea
E-mail: smilechi@naver.com, zeroone@etri.re.kr, jjyoo@etri.re.kr, kdseo@yonsei.ac.kr

ABSTRACT

Synchronization between media is an important aspect in the design of multimedia streaming system. This paper proposes a precise media synchronization mechanism for digital video and audio transport over IP networks. To support synchronization between video and audio bitstreams transported over IP networks, RTP/RTCP protocol suite is usually employed. To provide a precise mechanism for media synchronization between video and audio, we suggest an efficient media synchronization algorithm based on NPT (Normal Play Time) which can be derivable from the timestamp information in the header part of RTP packet generated for the transport of video and audio streams. With the proposed method, we do not need to send and process any RTCP SR (sender report) packet which is required for conventional media synchronization scheme, and accordingly could reduce the number of required UDP ports and the amount of control traffic injected into the network.

Keywords: precise media synchronization, RTP, RTCP, multimedia streaming

1. INTRODUCTION

Digital media is becoming an indispensable part of people's daily life thanks to the rapid development and wide adoption of handy digital media capturing devices, rich digital contents, portable media devices and versatile sharing networks. More and more users show greater demands for enjoying digital media services through various PC and non-PC devices over the Internet and wireless networks [1].

Generally, a Real-time Transport Protocol (RTP) packet is used for transmitting media data in order to transmit video/audio using an Internet Protocol (IP) network, and an RTP Control Protocol (RTCP) packet is used for secondarily cooperating with the RTP packet [1], [2]. In particular, one of various important functions of the RTCP packet is providing media synchronization information. Since the video and the audio are different media, a media sampling rate for acquiring an access unit corresponding to a unit of RTP packetization is different from each other. Accordingly, the video and the audio need to be transmitted using each different and independent RTP session. Information used for synchronization in a header corresponds to a "time stamp" field, and a value is independently generated for each video/audio access unit based on the sampling rate of the video and the audio. Since the video and the audio independently generate a "time stamp" value, synchronization between the video and the audio may not be performed using only "time stamp" information. Accordingly, time information to which a video stream and an audio stream may be commonly referred is required for providing synchronization between the video and the audio.

A method of providing common time information uses an RTCP Sender Report (SR) packet. A "Normal Play Time (NPT) time stamp" field provides the common time information to which the video and the audio are commonly referred, and an "RTP time stamp" field records an RTP time stamp of the video or the audio corresponding to an "NPT time stamp". Accordingly, each RTP time stamp value by which synchronization between the video and the audio is performed by a medium of the "NPT time stamp" may be estimated. Each RTCP session is generated for each of a video session and an audio session, and is transmitted to be within 5 % of the total traffic [2]. Each time the RTCP session is periodically transmitted, the RTP time stamp of each media corresponding to the NPT time stamp is recorded in the RTCP packet and is transmitted, thereby enabling a receiver to acquire the information required for synchronization.

As described above, since a legacy media synchronization method requires the "time stamp" information of the RTP packet and transmission of the RTCP SR packet periodically providing the NPT time stamp value, complexity or a processing process is complex. In particular, when an amount of traffic of a network is excessive, a congestion problem of the network may worsen due to the RTCP SR packet transmission.

In this paper, we propose a method for supporting precise synchronization with respect to video and audio using an NPT induced from time stamp information to be recorded in a header of a RTP packet when performing RTP packetization of the video and the audio information.

2. CONVENTIONAL SYNCHRONIZAZTION METHOD

To transport compressed media information over Internet Protocol (IP) in real-time, RTP and RTCP are usually employed [1], [2], [3]. RTP carries the payload containing the media data and RTP timestamp to facilitate the real-time transmission. RTCP serves for controlling quality of the transmitted data. RTP and RTCP packets both run

over the same transport layer protocol (e.g., UDP), but usually they are carried on separate channels (e.g., a separate UDP port).

To provide inter-media synchronization between video and audio, the basic idea is to compare periodically audio and video timestamps of RTP packets for the streaming application at well-defined time intervals (synchronization points) [3], [4], [5]. However, the intrinsic problem is that audio and video timestamps are coded in different way, so that they cannot be directly compared at all. According to RFC 3550, it is stated that separate audio and video streams should not be carried in a single RTP session and demultiplexed based on the payload type or SSRC fields. However, we cannot directly use RTP timestamp to synchronize data carried by different RTP sessions for the following two reasons. Firstly, RTP timestamp should be initialized to random offsets at session startup to minimize the risk of breaking encryption. Secondly, RTP timestamp increases in proportion to the sampling rate of media. Usually the sampling rates of audio and video data are quite different. Thus, the rates of increase in RTP timestamp for video and audio sessions are not the same. To circumvent these problems, RTCP SR packets carrying both the RTP and the NTP timestamp are generally employed as shown in Fig. 1. In the header structure of RTCP SR packet, the first timestamp is a 64 bit number that indicates, according to NTP (Network Time Protocol [6]), an absolute (wall-clock) time since 00.00 UTC of January 1 1900. The most significant word indicates the number of seconds elapsed since that time, while the least significant word defines the elapsed microseconds converted into a 32 bit number. The second timestamp represents the same value, but it is converted into the format of the RTP timestamp, just like the ones carried in RTP packets. More precisely, it is calculated (from the NTP timestamp) with the same frequency clock, and with the same initial random offset as the timestamp of RTP packet. These values allow lip-synchronization between audio and video streams originating from the same sender, since their clock reference will be the same.

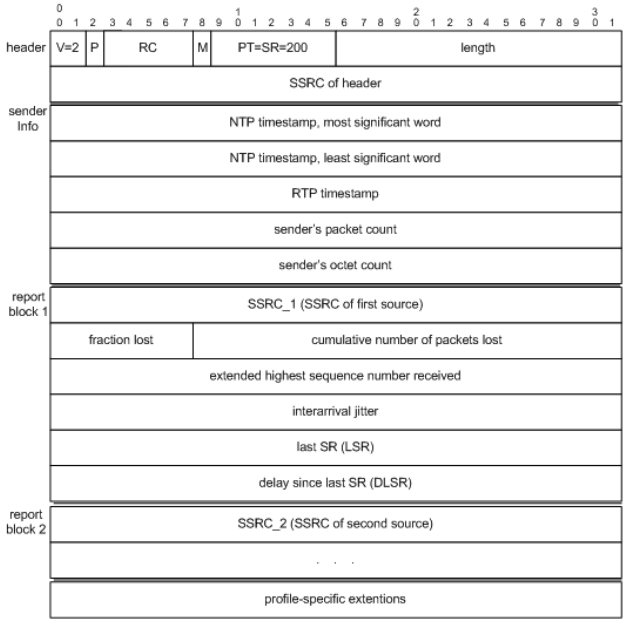


Fig. 1: Structure of RTCP SR packet.

By inspecting the relation between RTP and NTP timestamps in RTCP SR packet, we can find out the reference time corresponding to the RTP timestamp specified in RTP packet [7].

3. PROPOSED SYNCHRONIZATION METHOD

In this section, we propose a method of supporting precise synchronization between video information and audio information using an NPT. The proposed method includes the following subsequent procedures for its general working: *i*) receiving video information using a decoding device, *ii*) receiving audio information using a decoding device, *iii*) calculating the NPT of the video information using an RTP time stamp included in the received video information, *iv*) calculating the NPT of the audio information using the RTP time stamp included in the received audio information, *v*), comparing the NPT of the video information and the NPT of the audio information to calculate a difference value; *vi*) determining whether the calculated difference value is included in a specific synchronization region, *vii*) and outputting the audio information and the video information when the calculated difference value is determined to be included in the specific synchronization region.

The proposed method uses a NPT acquired from a RTP time stamp in order to match synchronization of video information and audio information. Accordingly, the proposed method includes a method of inducing each NPT using only the RTP time stamp included in the video and audio information, and the method is described below.

First, a method of inducing the NPT of the video information is described. In the proposed method, $NPT_{V_o}^k$ corresponding to an NPT of a *k*-th picture in which the video information received from a decoding device is outputted to a display device may be induced using RTP time stamp information by the following Eq. (1):

$$NPT_{V_o}^k = \left(RTPT_{V_o}^k - RTPT_{V_o}^1\right) / SR_v \qquad (1)$$

where $RTPT_{V_o}^1$ denotes an RTP time stamp of a first output picture (generally the first I-picture or IDR picture), $RTPT_{V_o}^k$ denotes an RTP time stamp of a *k*-th output picture, and SR_V denotes a sampling rate with respect to an access unit of video sequence in a transmitter. 90 KHz may be generally applied to SR_V with respect to the video information, however, SR_V is not limited to this value, and a time stamp value with respect to each picture is generated based on SR_V.

Since an outputted unit corresponds to a inconsecutive individual picture in the case of the video information, the NPT may be easily acquired for each output picture as described above. However, since an output unit of the audio information corresponds to a consecutive Pulse Code Modulation (PCM) data block, the output unit may not be classified and the NPT may not be directly acquired. In

order to solve this problem, a method of acquiring the NPT with respect to the audio information using a size of a wave-out buffer where PCM data stay before PCM data is outputted is proposed.

Fig. 2 illustrates a process of playing a single audio frame as PCM data and inputting and outputting the PCM data to a wave-out buffer after the single audio frame is decoded. As illustrated in Fig. 2, the audio information is played as the PCM data, and is inputted/outputted to the wave-out buffer, and audio compression data extracted from an RTP packet for each frame is periodically decoded and is played as the PCM data, and the played PCM data is consecutively stored in the wave-out buffer. A PCM data block stored in the wave-out buffer is transmitted to an output device, and is outputted to a speaker by a device driver. A size of the wave-out buffer is always set as a constant value t_{buff} for audio output of a continuously constant speed.

Then we can estimate $RTPT^{s}_{A_o}$ corresponding to an RTP time stamp with respect to an *s*-th PCM data block to be outputted to the wave-out buffer based on the above-described process of processing audio data using

$$RTPT^{s}_{A_o} = RTPT^{n}_{A_i} - \left(t_{buff} \times SR_A\right), \qquad (2)$$

where $RTPT^{n}_{A_i}$ denotes an RTP time stamp value of n-th PCM data inputted into a wave-out buffer at a time when $RTPT^{s}_{A_o}$ is calculated, and SR_A denotes a sampling rate with respect to a frame corresponding to a basic access unit of audio. A frequency up to a maximum of 48 KHz may be applied to the AAC information, however, the AAC information is not limited to this value.

Accordingly, $NPT^{s}_{A_o}$ corresponding to the NPT of the *s*-th PCM data block to be directly outputted to the speaker may be calculated the following equation:

$$NPT^{s}_{A_o} = \left(RTPT^{s}_{A_o} - RTPT^{1}_{A_o}\right) / SR_A, \qquad (3)$$

where $RTPT^{1}_{A_o}$ denotes a time stamp value of a PCM data block being first outputted.

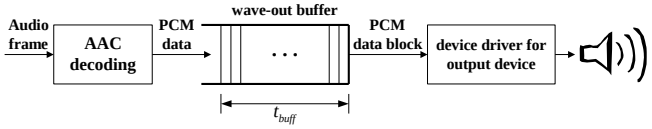


Fig. 2: Input/output process of PCM data in the wave-out buffer.

Fig. 3 is a block diagram illustrating a synchronization algorithm of video information and audio information using an NPT according to the proposed method.

A video information analysis unit calculates the NPT of the video information using an RTP time stamp included in the received video information by using Eq. (1).

In the case of the video information, we first extract $RTPT^{m}_{V_i}$ corresponding to a time stamp for each picture from a received RTP packet, and finds $RTPT^{k}_{V_o}$ for each output picture according to a picture display sequence based on picture reordering considering B-pictures. Thereafter, we calculate $NPT^{k}_{V_o}$ corresponding to an NPT of a *k*-th output picture using the above-described Eq. (1) based on $RTPT^{k}_{V_o}$.

An audio information analysis unit calculates the NPT of the audio information using the RTP time stamp included in the received audio information by using Eq. (2) and (3).

We first perform AAC decoding for each audio frame being loaded in an RTP and arriving to restore PCM data. Then, we can extract $RTPT^{n}_{A_i}$ corresponding to the RTP time stamp of the audio frame sequentially arriving from an RTP packet header. Thereafter, we calculate $RTPT^{s}_{A_o}$ corresponding to a time stamp of a PCM data block to be outputted using Eq. (2) based on $RTPT^{n}_{A_i}$. Now we can calculate $NPT^{s}_{A_o}$ corresponding to the NPT of the PCM

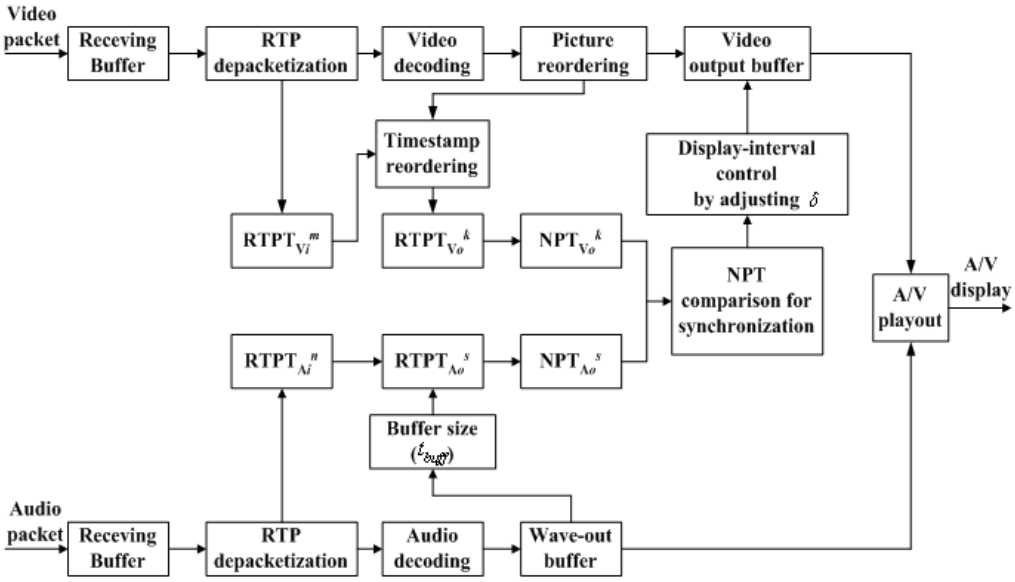


Fig. 3: Block diagram of the proposed synchronization algorithm based NPT.

data block to be outputted using Eq. (3).

When a picture to be outputted is assumed as a *k*-th picture and a PCM data block of audio to be synchronized with the picture is assumed as an *s*-th PCM data block, we compare $NPT_{V_o}^k$ and $NPT_{A_o}^s$, and adjust a display interval of a video picture to match synchronization.

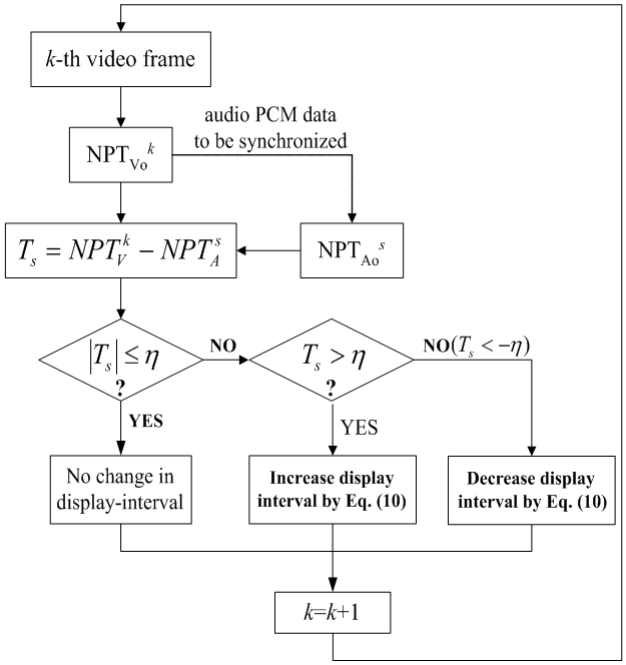


Fig. 4: Block diagram of NPT comparison processing for precise synchronization.

Henceforth, we describe a precise synchronization process of the video information and the audio information in detail. Since the audio information is more important than the video information, we adjust a video display speed in order to enable the audio information to be continuously outputted regardless of the video information and to be synchronized with the outputted video information. Fig. 4 shows a flowchart illustrating NPT comparison processing for precise synchronization of audio information and video information. First, we compare the NPT of the video information and the NPT of the audio information to calculate a difference value by subtracting an NPT value of the audio information from the NPT value of the video information. The difference value T_s between $NPT_{V_o}^k$ and $NPT_{A_o}^s$ to be used for the NPT comparing may be acquired by

$$T_s = NPT_{V_o}^k - NPT_{A_o}^s . \quad (4)$$

We determine whether the calculated difference value is included in a specific synchronization region. When the calculated difference value is determined to be included in the specific synchronization region, we regard that the video picture and audio frame are correctly synchronized to within an in-synch region. Thus, if $|T_s|$ is within η corresponding to the established synchronization region (an in-sync region), synchronization is determined to be matched, and the video picture is displayed at display intervals based on a given video frame-rate established. However, when the calculated difference value is determined to be excluded from the specific synchronization region, we adjust the display interval of the current video data.

When $|T_s|$ is outside η, the display interval adjustment unit may determine whether the video information corresponds to an output state faster or slower than the audio information to adjust the display interval between pictures of the video information.

The display interval of the video information f_I may be calculated by a predetermined frame rate f_R in accordance with Eq. (5):

$$f_I = 1000 / f_R . \quad (5)$$

The picture interval size adjustment parameter may be defined as a value of multiplying the difference value T_s by a scale factor. When $|T_s|$ is outside η, a size of the picture interval size adjustment parameter δ may be determined by a scale factor s_f, and may be represented as Eq. (6):

$$\delta = T_s \cdot s_f \ \text{(ms)}. \quad (6)$$

When synchronization is not performed, s_f may adjust a convergence speed for matching synchronization again and may verify that a value of around 0.05 to 0.1 is appropriate, using an experiment.

The new display interval f_I' adjusted by δ may be calculated in accordance with Equation 7:

$$f_I' = f_I + \delta . \quad (7)$$

According to the proposed method, it is possible to induce an NPT using only an RTP time stamp by eliminating a separate need for transmitting and processing an RTCP SR packet of video information and audio information. Also, it is possible to reduce a number of User Datagram Protocol (UDP) ports required for transmitting an RTCP packet, and to reduce an amount of control traffic coming into a network since RTCP packet transmission is unnecessary.

4. EXPEIRMENTAL RESULTS

In order to evaluate the synchronization performance and effectiveness of the proposed mechanism, we have developed a prototyped streaming system implemented on Internet using Darwin Streaming Server (DSS) developed by Apple [7]. Performance evaluations were executed between a pair of PCs: one as a streaming server and the other as a client station. Before carrying out the experiment, we need to set an appropriate value to the thresholds η used in the proposed method. For this purpose, prior research results on the lip synchronization between audio and video data are considered. Lip synchronization is a crucial human perception issue for video streaming and video telephony systems. Previous research shows the following experimental results on lip synchronization [8]. In most cases, people do not detect the synchronization error if the temporal skew which is the time difference between audio and video is less than 80 *ms*. However, if the temporal skew becomes larger than 160 *ms*, every observer detects this error and feels uncomfortable with the

video service. If the error is larger than 80 *ms* but smaller than 160 *ms*, the detection of the error depends on the communication environment. One interesting result is that people feel more comfortable with the "video ahead of audio" case than the other one. Therefore, it is reasonable to choose the thresholds of η less than 30 ms for precise high-quality synchronization. Note that the suitable values of η for other multimedia communication systems may be slightly different from the current values, since they depend on the hardware performance, the codec types, and/or user requirements.

We have carried out an experiment in which stored SVC video [9] and AAC audio are transmitted simultaneously from the server to the client through Internet. The SVC video and AAC audio are transmitted as two distinct transport streams by RTP which is implemented on top of UDP [10].

In order to evaluate synchronization accuracy between video and audio data, the relative output temporal skew between audio segments and the equivalent video frame outputs is observed. First, to show the importance of the synchronization itself, the results obtained by applying no synchronization are given in Fig. 5. For the case of no synchronization, it is obvious to even a casual observer that the video frame and audio segment are not synchronously output on the client station, even though the video frame and audio segment were synchronously transmitted from the streaming server. Since audio and video are carried by different packets through different UDP port, and since decoding operations are taking place on two separate codecs, the original inter-media synchronization from the server could be lost. As shown in Fig. 5, the deviation of the temporal skew from the origin becomes even larger as time goes by.

Fig. 6 compares the result of the temporal skew between the video and the audio segment when the proposed synchronization synchronization method is applied with $\eta = 25\,ms$. From the figure, we observe that the proposed synchronization method can provide quite accurate lip-synchronization. The video frame and audio segment were correctly synchronized to within 25 *ms*. With different scale factor s_f, we can control the convergence speed to the in-synch region when the temporal skew digresses from the in-synch region. As can be seen in Fig. 6, larger s_f increases the convergence speed.

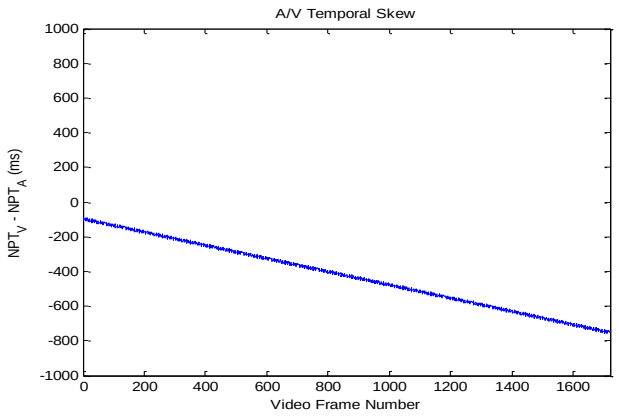
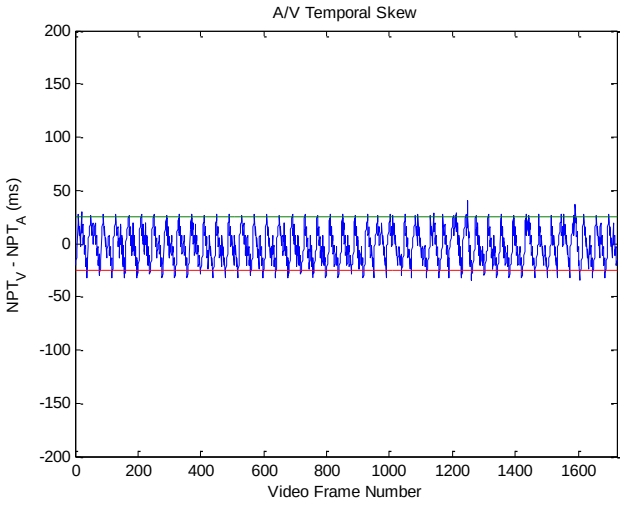
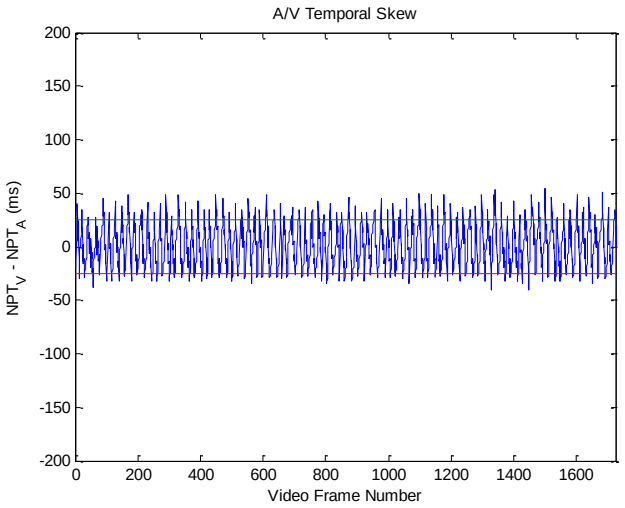


Fig. 5: Temporal skews between video and audio streams without synchronization scheme.



(a) s_f =0.06



(b) s_f =0.1

Fig. 6: Temporal skews between video and audio streams with the proposed synchronization scheme. ($\eta = 25\,ms$).

5. CONCLUSIONS

In this paper, we proposed a precise audio and video synchronization scheme for multimedia streaming over IP networks. The proposed method is based on deriving exact NPT information from the RTP time stamps contained in the header part of RTP packets generated for the transport of video and audio streams over IP networks. Efficient methods for deriving the NPT information for both audio and video streams are described. With the derived NPT information, a precise media synchronization method is proposed. The proposed method does not require to send and process any RTCP SR packet which is required for conventional media synchronization scheme, and accordingly could reduce the number of required UDP ports and the amount of control traffic injected into the network. It is shown by simulations that the proposed method provides high-quality precise synchronization performance between audio and video streams.

6. ACKNOWLEDGEMENT

This work was supported by the IT R&D program of MKE/KCC/IITA. [2008-S-006-01, Development of Open-IPTV Technologies for Wired and Wireless Networks]

7. REFERENCES

[1] D. Wu, Y. Hou, and Y. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1855-1877, Dec. 2000.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Real-time transport protocol," IETF *RFC 3550*, July 2003.

[3] L. Bertoglio, and P. Migliorati, "Intermedia synchronization for video conference over IP," *Signal processing: Image Communication*, vol. 15, no. 1, pp. 149-164, 1999.

[4] A. Boukerche, and H. Owens, "Media synchronization and QoS packet scheduling algorithms for wireless systems," *Mobile Networks and Applications*, vol. 10, no. 1, pp. 233-249, Feb. 2005.

[5] F. Segui, J. Cebollada, and J. Mauri, "Multimedia group synchronization algorithm based on RTP/RTCP," *IEEE Int. Symp. on Multimedia*, pp. 754-757, San Diego, USA, Dec. 2006.

[6] D. Mills, "Network Time Protocol (version 3)," *IETF RFC 1305*, March 1992.

[7] Apple Darwin Streaming Server (DSS) available at http://developer.apple.com/darwin/projects/streaming/

[8] R. Steinmetz, "Human perception of jitter and media synchronization, "*IEEE Journ. Selected Areas in Commun.*, vol. 14, no. 1, pp. 61-72, 1996.

[9] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits and Systems for Video Technol.*, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[10] S. Wenger, Y. Wang, and T. Schierl, "RTP Payload Format for SVC Video," IETF Internet Draft: *draft-ietf-avt-rtp-svc-02.txt*, July 2007.