# Efficient Media Synchronization Mechanism for SVC Video Transport over IP Networks

Kwang-Deok Seo, Soon-Heung Jung, and Jin-Soo Kim

**The scalable extension of H.264, known as scalable video coding (SVC) has been the main focus of the Joint Video Team's work and was finalized at the end of 2007. Synchronization between media is an important aspect in the design of a scalable video streaming system. This paper proposes an efficient media synchronization mechanism for SVC video transport over IP networks. To support synchronization between video and audio bitstreams transported over IP networks, a real-time transport protocol/RTP control protocol (RTP/RTCP) suite is usually employed. To provide an efficient mechanism for media synchronization between SVC video and audio, we suggest an efficient RTP packetization mode for inter-layer synchronization within SVC video and propose a computationally efficient RTCP packet processing method for inter-media synchronization. By adopting the computationally simple RTCP packet processing, we do not need to process every RTCP sender report packet for inter-media synchronization. We demonstrate the effectiveness of the proposed mechanism by comparing its performance with that of the conventional method.**

**Keywords: Media synchronization, RTP/RTCP, scalable video coding, SVC video transport.**

Manuscript received Sept. 17, 2007; revised Dec. 24, 2008.

This work was supported by the Gangwon-Alberta Research Collaboration Fund and the IT R&D program of MIC/IITA, Rep. of Korea [2005-S-103-02, Development of Ubiquitous Content Access Technology for Convergence of Broadcasting and Communications].

Kwang-Deok Seo (phone: + 82 33 760 2788, email: kdseo@yonsei.ac.kr) is with the Computer and Telecommunications Engineering Division, Yonsei University, Wonju, Rep. of Korea.

Soon-Heung Jung (email: zeroone@etri.re.kr) is with the Broadcasting & Telecommunications Convergence Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Jin-Soo Kim (email: jskim67@hanbat.ac.kr) is with the Department of Multimedia Engineering, Hanbat National University, Daejeon, Rep. of Korea.

## I. 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 demand for digital media services to be provided through various PC- and non-PC devices over the Internet and wireless networks. Such ubiquitous multimedia services pose great challenges to traditional coding techniques, such as the H.264/MPEG-4 AVC coding scheme [1]. Scalable video coding (SVC) is a new scalable coding technique developed to solve the problems of low compression efficiency, unsupportability of combined scalability, and high implementation complexity, which are caused by the conventional layered coding-based scalability attempted in existing H.263, MPEG-2, and MPEG-4 video coding. The SVC technique, also known as the scalable extension of H.264/MPEG-4 AVC, has been standardized by the Joint Video Team (JVT) of the ISO/IEC MPEG and the ITU-T Video Coding Experts Group [2], [3]. SVC is intended to achieve both high compression performance and adaptation to video delivery over heterogeneous networks. SVC is based on H.264/MPEG-4 AVC and provides three scalability modes, including temporal, spatial, and quality scalability. Unlike the conventional scalability modes supported in the MPEG-2, H.263, and MPEG-4, SVC scalability can combine three scalability modes, which are aggregated to a single bitstream.

Figure 1 shows how to construct an SVC combined scalability with two spatial layers and five temporal levels [4]. Each spatial layer consists of a quality base layer and a quality enhancement layer (FGS layer).

The input pictures in spatial layer 0 are created by down-

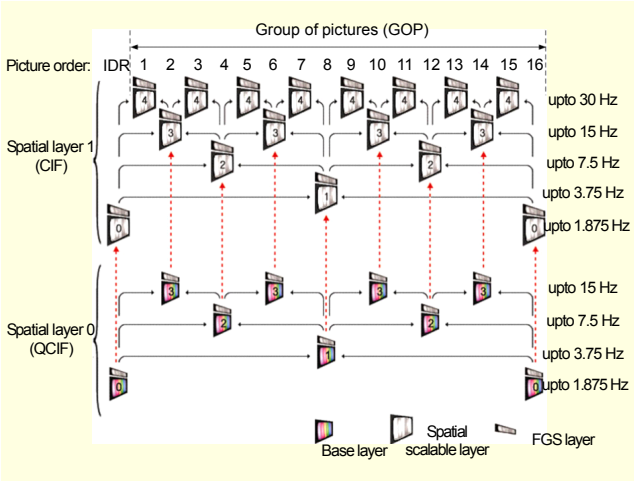


Fig. 1. Example of SVC video coding based on combined scalability.

sampling the input pictures in spatial layer 1 by a factor of two. A group of pictures (GOP) with the size of 16 is coded with hierarchical B-picture techniques to obtain four temporal levels in spatial layer 0 and five temporal levels in spatial layer 1. The lowest spatial layer (layer 0) has QCIF resolution and 4 temporal levels with frame rates of 1.875, 3.75, 7.5, and 15 Hz, respectively. The higher spatial layer (layer 1) has a CIF resolution and 5 temporal levels that give the additional maximum frame rate of 30 Hz. The dotted arrow in Fig. 1 designates inter-layer prediction to remove redundancy between spatial layers. In the temporal dimension, each picture belongs to one temporal layer indicated by the number in the middle of each picture.

In combined scalability, a coded picture of a base or scalable enhancement layer is produced as one or more slices at the video coding layer (VCL). The network abstraction layer (NAL) encapsulates each slice generated by the VCL into a typical NAL unit, which forms the basic structure of an SVC bitstream. To transport the SVC bitstream encapsulated in NAL units over Internet Protocol (IP) in real-time, the real-time transport protocol (RTP) and RTP control protocol (RTCP) are usually employed [5], [6]. RTP carries the payload containing the SVC NAL units with some additional header information, such as sequence number and RTP timestamp, to facilitate the real-time transmission. RTCP controls the quality of the transmitted data. The RTP and RTCP packets run over the same transport layer protocol (namely, UDP [7]); however, they are usually carried on separate channels, that is, separate UDP ports.

Based on the RTP payload format for SVC, which is under standardization in IETF for loading the SVC NAL units onto RTP payload part, we suggest an efficient RTP packetization mode suitable for layer synchronization among scalable layers of SVC video, as shown in Fig. 1. Furthermore, for efficient inter-media synchronization between SVC video and audio, we fully exploit the conventional function of the RTCP sender report (SR) packet in a standard compatible manner. Every client involved in the streaming service generates an RTCP receiver report (RR) packet, with information about all senders they have heard from recently (since the last report). An active server also generates an RTCP SR packet, which is just like an RR packet with 20 bytes of additional information about the server. In particular, an SR packet contains timestamps which allow the recovery of an absolute time reference for synchronization. An RTP timestamp in an SR packet begins at a random number and its rate of increment is proportional to its sampling rate [6]. Thus, it does not directly give information on absolute time reference for synchronization. To synchronize audio and video data, we need to utilize RTCP SR packets to find out the absolute time information corresponding to each RTP timestamp carried by each RTP packet. In this paper, we propose an efficient synchronization method for SVC video and audio. In the proposed method, we do not need to process every RTCP SR packet for synchronization. Moreover, the method does not require any floating-point operations or any divisions at all. Obviously, this is a clear advantage for embedded processors used for video streaming devices. As demonstrated through extensive simulations, the proposed method shows notable advantages compared to previous methods, such as Bertoglio's [8].

## II. Suggested RTP Packetization Mode for Layer Synchronization

In order to transport an SVC bitstream over IP networks, a new payload format for RTP is currently being specified in IETF [8]. The Audio/Video Transport (AVT) Working Group of the IETF started in November 2005 to draft the RTP payload format for SVC and the signaling for layered coding structures. As SVC is a backward compatible extension of H.264, the same should be the case for its RTP packetization. In particular, it is possible to transport the base layer utilizing the same packetization scheme as that of RFC 3984 [9]. Thus, RFC 3984-aware legacy devices are still capable of utilizing an SVC base layer in an RTP transport environment.

An RTP stream carrying only one layer would carry NAL units belonging to that layer only. An RTP stream carrying a complete scalable video bitstream would carry NAL units of a base layer and one or more enhancement layers [10]. In the former case, however, the system administrator of the server should open a separate UDP port for each RTP session to carry a single layer. Thus, the server should open a sufficient number of ports to transport all the layers. System administrators would

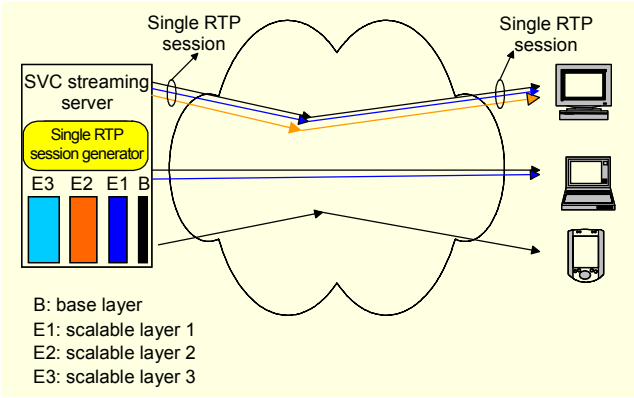


Fig. 2. SVC streaming scenario based on a single RTP session.

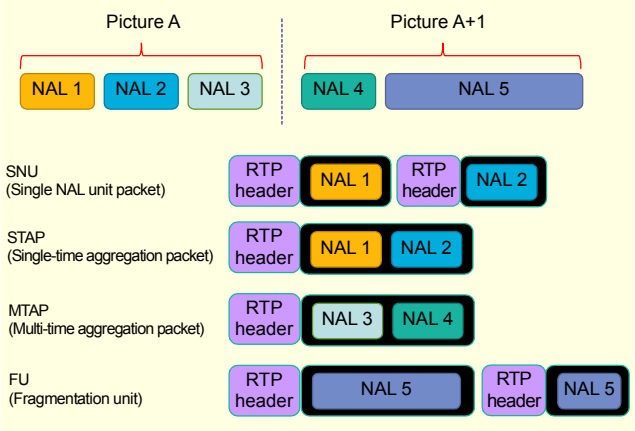


Fig. 3. Basic RTP packet types for SVC.

like to avoid opening too many UDP ports in their firewalls because of the security risk and the administrative effort. Moreover, for mass deployment to end terminals, it is desirable to reduce the number of UDP ports in a firewall to the absolute minimum–ideally to a single one [10]. In this respect, the latter approach is much preferred.

This line of thought leads to the exemplary service scenario depicted in Fig. 2, in which the server opens only a single RTP session to carry one or more layers. For each terminal, the server composes a bitstream tailored to the terminal's needs by aggregating NAL units of appropriate layers. A single-RTP-session generator is used to aggregate the extracted contents from potentially more than one scalable enhancement layer into a single RTP stream carrying one or more layers.

To support the service scenario shown in Fig. 2, it is necessary to support the encapsulation of NAL units from multiple SVC layers into a single RTP packet in the payload format. The IETF specification on RTP payload format for SVC contains four basic mechanisms, including a single NAL unit (SNU), a single-time aggregation packet (STAP), and a multi-time aggregation packet (MTAP) to aggregate more than one NAL unit into a single RTP packet, as well as another mechanism called a fragmentation unit (FU) to split overly large NAL unit into multiple RTP packets [10], [11]. Figure 3 shows the basic principle of forming the four RTP packet types. As shown in Fig. 3, the SNU type can load only one NAL unit (NAL1 or NAL2) in one RTP, and the STAP can simultaneously load multiple NAL units (NAL1 and NAL2) that belong to the same time instant in one RTP packet. There are two types of STAP: the STAP-A type that loads NAL units in an RTP packet in the same order as encoding and the STAP-B type that loads NAL units in an RTP packet without considering the encoding order for interleaving purposes. The MTAP can load multiple NAL units (NAL3 and NAL4) belonging to different time instants in one RTP packet at a time and basically supports interleaving. The MTAP-16 type supports a 16-bit time offset, and an MTAP-24 type supports a 24-bit time offset, depending on the size of the time offset field for displaying the difference in presentation time instant between the NAL units. The FU divides an NAL unit (NAL5) into two or more so that it does not exceed the maximum transmission unit (MTU) size and loads the divided units into respective corresponding RTP packets. This prevents packet fragmentation in a router or gateway, which can occur during transmission if the size of one NAL unit exceeds that of the MTU of a network.

Table 1. Allowed packet types for RTP packetization modes of SVC.

| NAL unit type | Packet type | Single NAL unit mode | Non-interleaved mode | Interleaved mode |
|---|---|---|---|---|
| 0 | Undefined | Ignore | Ignore | Ignore |
| 1-23 | Single NAL unit | Yes | Yes | No |
| 24 | STAP-A | No | Yes | No |
| 25 | STAP-B | No | No | Yes |
| 26 | MTAP16 | No | No | Yes |
| 27 | MTAP24 | No | No | Yes |
| 28 | FU-A | No | Yes | Yes |
| 29 | FU-B | No | No | Yes |
| 30-31 | Undefined | Ignore | Ignore | Ignore |

Three fundamentally different packetization modes of operation are supported in [11]: SNU mode, non-interleaved mode, and interleaved mode. Table 1 summarizes the allowed RTP packet types for each packetization mode [11].

The SNU mode is able to support only the SNU type that can load only one NAL unit having 1 to 23 *NAL_unit_types* in an RTP packet, and its application field is restrictive. Thus, the latest Internet-draft document *draft-ietf-avt-rtp-svc-02.txt*

released in July 2007 designates that the SNU mode shall not be used for RTP packetization for SVC video [11].

In non-interleaved mode, the NAL units should be aggregated in decoding order by adopting STAP-A, whereas in interleaved mode, NAL units belonging to multiple pictures can be aggregated out of decoding order by adopting STAP-B and MTAP. Non-interleaved mode is intended to avoid excessive RTP/UDP/IP header overhead that would result from the encapsulation of small NAL units in each single RTP packets, whereas the interleaved mode provides an error resilience tool against burst errors. STAP-A aggregates NAL units with identical NAL unit times, whereas MTAP aggregates NAL units with differing NAL unit times. Here, NAL unit time is defined as the value that the RTP timestamp would have if that NAL unit was transported in its own RTP packet. As shown in Fig. 1, pictures belonging to different spatial layers but having the same picture number (or display time) must have the same NAL unit time. Thus, by adopting STAP-A, it is far more feasible to provide synchronization between pictures belonging to different spatial layers but with identical NAL unit times. Therefore, non-interleaved mode is the most suitable mode for systems that require very low end-to-end latency and timely synchronization among NAL units from multiple SVC layers aggregated in a RTP packet. Furthermore, as shown in Table 1, except for the SNU mode, only non-interleaved mode supports the single NAL unit packet type that can contain only a single NAL unit in the RTP payload. As a result, non-interleaved mode can be suggested as a mandatory packetization mode for fast and real-time streaming requiring timely synchronization among SVC layers. Interleaved mode can be considered as an optional mode for error resilience as it provides interleaving function against burst packet loss. As for streaming service over highly error-prone channels, interleaved mode of RTP packetization can be effectively used. However, it incurs additional processing delay due to the de-interleaving process at the client. Therefore, we employ the non-interleaved mode as a basic RTP packetization mode for efficient layer synchronization.

## III. Conventional Inter-media Synchronization Method

The next problem to be resolved in relation to the synchronization issue is the need to provide inter-media synchronization between SVC video and audio. For this purpose, the basic idea is to periodically compare audio and video timestamps of RTP packets for the streaming application at well-defined time intervals (synchronization points). However, the intrinsic problem is that audio and video timestamps are coded in different ways so that they cannot be directly compared at all. According to RFC 3550, separate audio and video streams should not be carried in a single RTP session and should be identified based on the payload type or synchronization source (SSRC) fields [6]. However, we cannot directly use RTP timestamps to synchronize data carried by different RTP sessions for two reasons. First, an RTP timestamp should be initialized to random offsets at session startup to minimize the risk of breaking encryption. Second, an 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 timestamps for SVC video and audio sessions are not the same.

To circumvent these problems, RTCP SR packets carrying both the RTP and the network time protocol (NTP) timestamps are generally employed [6]. In the header structure of an RTCP SR packet, the first timestamp is a 64-bit number that indicates, according to NTP [12], an absolute (wall-clock) time since UTC 00.00 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 packets. These values allow lip-synchronization between audio and video streams originating from the same sender since their clock reference will be the same. By inspecting the relation between RTP and NTP timestamps in an RTCP SR packet, we can find out the reference time corresponding to the RTP timestamp specified for RTP packets [6].

As such, the IETF standard RFC 3550 specifies mandatory header information of an RTCP packet to recover the absolute time reference at the receiving terminal. The RFC 3550 standard only suggests using the absolute time reference deducible from the RTCP packets for synchronization. The methods used to efficiently recover the absolute time reference and to apply the recovered absolute time reference to synchronize different media streams are absolutely up to the system designer. Bertolio and others proposed a new method to efficiently recover the absolute time reference using RTCP SR packets [8]. Based on the new recovery method, they also proposed a method to synchronize video and audio streams in two different and separate applications, such as VIC and VAT.

Figure 4 shows RTP and RTCP streams for audio and SVC video sessions. As shown in the figure, each RTP packet of an SVC video session aggregates NAL units that all share the same NAL unit time. The RTP timestamp of each RTP packet must be set to the NAL unit time of all the NAL units to be aggregated. An aggregation packet, such as a STAP-A type

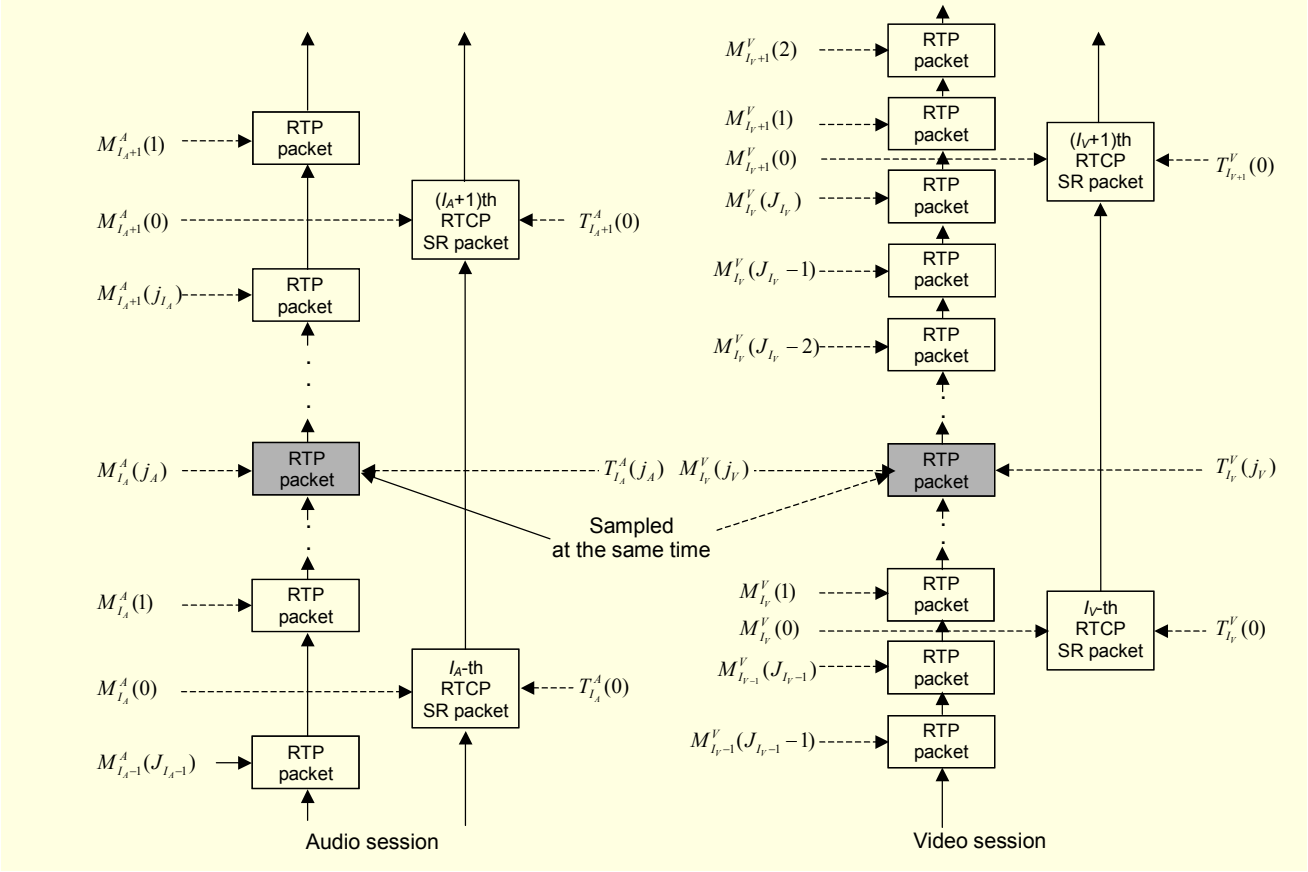


Fig. 4. RTP/RTCP streams for audio and SVC video sessions.

packet can carry as many NAL units as necessary. However, the total amount of data in an aggregation packet obviously must fit into an IP packet, and the size should be chosen so that the resulting IP packet is bound by the MTU size of the transport channel.

The superscripts $A$ and $V$ are used to denote audio and SVC video sessions, respectively. For the derivation of the relationship between the RTP timestamp of a specific RTP packet and the absolute time reference, let us consider the shaded RTP packet in the audio session shown in Fig. 4. For this RTP packet, $T_{I_A}^A(j_A)$ is the absolute time reference for the RTP timestamp of the $j_A$-th RTP packet after the $I_A$-th RTCP packet has been received. NTP tells us how to set the absolute time information. As a special case, when $j_A = 0$, $T_{I_A}^A(0)$ is the NTP timestamp contained in the $I_A$-th RTCP packet. Similarly, $M_{I_A}^A(j_A)$ is the RTP timestamp contained in the $j_A$-th RTP packet after the $I_A$-th RTCP packet, and $M_{I_A}^A(0)$ is the RTP timestamp for the $I_A$-th RTCP packet.

Let us assume that the shaded RTP packet in the SVC video session is sampled at the same time with the shaded RTP packet in the audio session. If the absolute time reference of this SVC RTP packet is represented by $T_{I_V}^V(j_V)$, it is required that $T_{I_A}^A(j_A) = T_{I_V}^V(j_V)$ for perfect synchronization. However, the transmission rates of RTP packets are normally not the same for different sessions. Moreover, RTCP packets for each session may be transmitted at different times. Thus, even if $T_{I_A}^A(j_A) = T_{I_V}^V(j_V)$, $I_A$ and $j_A$ of the audio session may not be equal to $I_V$ and $j_V$ of the SVC video session, respectively. Based on this fact, we can compute $T_{I_A}^A(j_A)$ of an RTP timestamp by using $M_{I_A}^A(j_A)$ carried by this RTP packet. We also use $M_{I_A}^A(0)$ and $T_{I_A}^A(0)$ values in the computation, which can be obtained by the $I_A$-th RTCP packet. In the method proposed in [8], the absolute time reference $T_{I_A}^A(j_A)$ is obtained by

$$T_{I_A}^A(j_A) = T_{I_A}^A(1) + \sum_{k=2}^{j_A} \frac{\Delta M_{I_A}^A(k)}{R^A}, \tag{1}$$

where $R^A$ is the sampling rate of audio data. $T_{I_A}^A(1)$ is obtained by

$$T_{I_A}^A(1) = T_{I_A}^A(0) + \frac{M_{I_A}^A(1) - M_{I_A}^A(0)}{R^A}. \tag{2}$$

In (1), $\Delta M_{I_A}^A(k)$ is the difference between the RTP timestamps of two adjacent RTP packets and is given by

$$\Delta M_{I_A}^{A}(k) = M_{I_A}^{A}(k) - M_{I_A}^{A}(k-1). \quad (3)$$

Computation of (1) and (3) continues until a new RTCP packet is received. After receiving the ($I_A$+1)th RTCP packet, (1) and (2) are computed again using $T_{I_A+1}^{A}(0)$ and $M_{I_A+1}^{A}(0)$ carried by this RTCP packet. When computing $T_{I_A}^{A}(j_A)$ by (1) in Bertoglio's method, the term $\sum_{k=2}^{j_A} \frac{\Delta M_{I_A}^{A}(k)}{R^A}$ is not computed directly. Instead, since the value of $T_{I_A}^{A}(j_A - 1)$ is already known, it is computed by

$$T_{I_A}^{A}(j_A) = T_{I_A}^{A}(j_A - 1) + \frac{\Delta M_{I_A}^{A}(j_A)}{R^A}. \quad (4)$$

The same procedure, computing (1) to (4) can be applied to the SVC video session to obtain $T_{I_V}^{V}(j_V)$.

When processing the $j_A$-th RTP packet for the audio session and the $j_V$-th RTP packet for the SVC video session, Bertoglio's decision rules based on $T_{I_A}^{A}(j_A)$ and $T_{I_V}^{V}(j_V)$ for synchronization are the following:

$$\begin{aligned}
&T_{I_V}^{V}(j_V) - T_{I_A}^{A}(j_A) > \eta_+ : \text{SVC video is ahead of audio,} \\
&\eta_+ \geq T_{I_V}^{V}(j_V) - T_{I_A}^{A}(j_A) \geq -\eta_- : \text{audio and SVC video are in synchronization,} \\
&T_{I_V}^{V}(j_V) - T_{I_A}^{A}(j_A) < -\eta_- : \text{audio is ahead of SVC video,}
\end{aligned} \quad (5)$$

where $\eta_+$ and $\eta_-$ are thresholds used for boundaries of the in-sync region. To apply this decision rule, it is evident that we need to inspect every arriving RTCP SR packet for the computation of (1) through (5). At every synchronization point, the temporal skew, which is the time difference between audio and video is compared to thresholds.

## IV. Proposed Inter-media Synchronization Method

In the conventional synchronization method described in (1) to (5), every calculation step involves truncation or rounding effects caused by division and floating-point operation. If the single-step error is irrelevant, after a number of steps, all the truncation errors lead to an increasing and significant error. Therefore, RTCP SR packets become of great importance, because they are used to periodically re-synchronize the algorithm. When a new RTCP SR packet is received, the NTP timestamp it carries is used to replace the current estimate of the absolute time reference value because it is supposed to be more closely tied to the sender clock. Thus, all approximation errors are removed every time an RTCP SR packet is received. The proposed inter-media synchronization method can improve this kind of repetitive and complex process for media synchronization.

We derive the proposed scheme from the conventional method described in (1) to (5). In this study, we exploit the fact that after a call connection has been setup, the codec type and the sampling rate are usually sustained during the connection.

By canceling out each term in the computation of $\sum_{k=2}^{j_A} \frac{\Delta M_{I_A}^{A}(k)}{R^A}$ in (1) and by using (2), we can simplify (1) as

$$\begin{aligned}
T_{I_A}^{A}(j_A) &= T_{I_A}^{A}(1) + \frac{M_{I_A}^{A}(j_A) - M_{I_A}^{A}(1)}{R^A} \\
&= T_{I_A}^{A}(0) + \frac{M_{I_A}^{A}(j_A) - M_{I_A}^{A}(0)}{R^A}.
\end{aligned} \quad (6)$$

Assuming that $R^A$ is kept as a constant, we can obtain (7) from the NTP and RTP timestamps carried by the 0-th and the $I_A$-th RTCP packet as

$$R^A = \frac{M_{I_A}^{A}(0) - M_0^{A}(0)}{T_{I_A}^{A}(0) - T_0^{A}(0)}. \quad (7)$$

Rearranging (7) for $T_{I_A}^{A}(0)$ and substituting it into (6) yields

$$T_{I_A}^{A}(j_A) = T_0^{A}(0) + \frac{M_{I_A}^{A}(j_A) - M_0^{A}(0)}{R^A}. \quad (8)$$

Similarly, we can apply this procedure to obtain the following relation for the RTP stream of SVC video session:

$$T_{I_V}^{V}(j_V) = T_0^{V}(0) + \frac{M_{I_V}^{V}(j_V) - M_0^{V}(0)}{R^V}. \quad (9)$$

By subtracting (9) from (8) and applying this result to (5), we can derive the following compact decision rule after some arithmetic:

$$\begin{aligned}
&d > \eta_0 + R^A R^V \eta_+ : \text{SVC video is ahead of audio,} \\
&\eta_0 + R^A R^V \eta_+ \geq d \geq \eta_0 - R^A R^V \eta_- : \text{audio and SVC video are in synchronization,} \\
&d > \eta_0 - R^A R^V \eta_- : \text{audio is ahead of SVC video,}
\end{aligned} \quad (10)$$

where the variable $d$ and the threshold constant $\eta_0$ are defined by

$$d = R^A M_{I_V}^{V}(j_V) - R^V M_{I_A}^{A}(j_A), \quad (11)$$

$$\eta_0 = R^A R^V (T_0^{V}(0) - T_0^{A}(0)) + R^V M_0^{A}(0) - R^A M_0^{V}(0). \quad (12)$$

Note that we only need to compute $d$ by (11) when examining the synchronization of each pair of RTP packets by (10). This is because $\eta_0 + R^A R^V \eta_+$ and $\eta_0 - R^A R^V \eta_-$ in (10) need to be

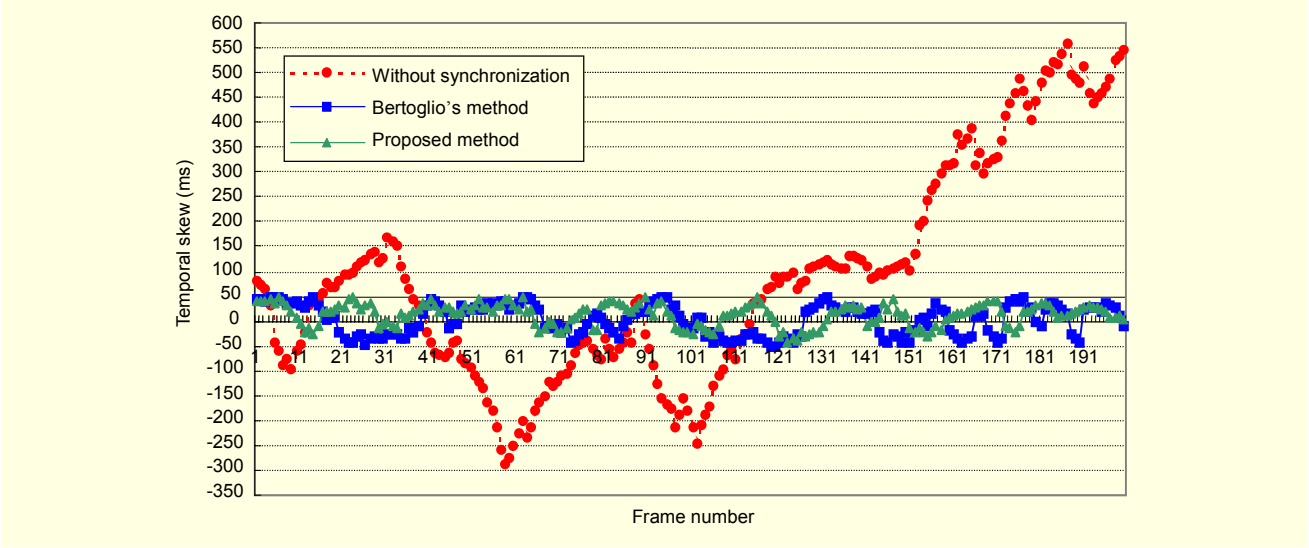


Fig. 5. Comparison of synchronization accuracy by temporal skew.

computed just once after the first RTCP packet is received. Since all of the $R^V$, $R^A$, $M^V_{I_V}(j_V)$, and $M^A_{I_A}(j_A)$ values in (11) are fixed-point numbers themselves, there is no need to utilize floating-point operations at all. Obviously, this is a clear advantage for embedded processors, which usually do not have floating point units. Moreover, (11) does not require any division operations as there are in (1) to (4). For ARM processors, avoiding division is a great advantage because they do not have any hardware divider. In [8], it was reported that truncation round-off errors may accumulate, since the method is computed by repeated divisions and summations. However, it is obvious that there is no possibility of error accumulation in the computation of (11). Note that only two fixed-point multiplications and one subtraction are needed for the computation of $d$ in (11).

## V. Experimental Results

To evaluate the synchronization performance and effectiveness of the proposed mechanism, we developed a prototype streaming system, which we implemented on the Internet using two 3.0 GHz Pentium IV PCs running the Windows XP operating system. 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 set appropriate values for the thresholds $\eta_+$ and $\eta_-$ used in (5) and (10). For this purpose, prior research results on the lip synchronization between audio and video data were 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 [13]. 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 greater 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 opposite. Therefore, it is reasonable to choose the thresholds of $\eta_+ = \eta_- = 50$ ms for high-quality synchronization. Note that the suitable values of $\eta_+$ and $\eta_-$ 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 carried out an experiment in which stored SVC video and AAC audio were transmitted simultaneously from the server to the client over the Internet. The SVC video and AAC audio were transmitted as two distinct transport streams by RTP, which was implemented on top of UDP. In particular, non-interleaved mode RTP packetization was applied to transport the SVC video data. The transmission bit rate used for the AAC audio was fixed, while the bit rate allocated for SVC video varied with time so that dynamic bandwidth adaptation [14], [15] by the extraction process of the SVC video could be activated to the SVC bitstream with the GOP structure shown in Fig. 1.

To evaluate synchronization accuracy between video and audio data, the relative output temporal skew between audio segments and the equivalent video frame outputs was observed

[8]. Figure 5 compares the results of the temporal skew between the video and the audio segment when the proposed synchronization and Bertoglio's synchronization methods are applied with $\eta_+ = \eta_- = 50\,\text{ms}$. To show the importance of the synchronization itself, the results obtained by applying no synchronization are also given in the same plot. In the case without synchronization, 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 ports, and since decoding operations take 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 gradually increases. The low deviation at the initial playback time is due to the initial buffering effect of the SVC video and audio codecs. The proposed method and Bertoglio's method result in similar temporal skew behavior. As Fig. 5 shows, the two synchronization methods can provide quite accurate lip-synchronization. Under the same conditions, the video frame and audio segment were correctly synchronized to within 50 ms. However, the proposed method requires far less computation than Bertoglio's method. Only one subtraction, two multiplications, and two comparisons are needed for the first RTCP SR packet in the proposed method, while three subtractions, two divisions, two additions, and two comparison operations are needed for each RTCP SR packet in Bertoglio's method. The total computational complexity of Bertoglio's method proportionally increases with the number of RTCP SR packets arriving at the client. However, the proposed method greatly reduces the necessary number of synchronization points to one and reduces the number of RTCP SR packets to be transmitted and processed to only one with less computation.

The proposed solution does not take into account the huge differences in delay that may be introduced between the different communication points, potentially resulting in jitter in the reconstruction of the audio-visual streams between different receivers. To ensure higher quality in the reconstructed material, priority is given to audio information. If an audio segment anticipates the corresponding video frame which has not arrived at the buffer within the due arrival time, the receiver simply discards the video frame. Conversely, when a video frame is ahead of its corresponding audio information, the video rendering stage is interrupted until the audio information arrives at the buffer. This approach may cause some visual impairment of the reconstructed video.

We evaluated the performance from a subjective point of view, and had human observers verify the degree of synchronization and the possible loss of quality of the reconstructed video. Clearly, a reliable subjective evaluation procedure is quite difficult to achieve. In our subjective experiment, the selected observers were unaware of the synchronization issues and had limited experience in audio-visual communication systems. Overall, 15 observers participated in the experiment and each observer was asked to evaluate the perceptual quality on the basis of an overall impairment score using a mean opinion score test. A seven-level impairment-score was used as shown in Table 2, instead of the traditional 5 levels. The choice of this modified scale allowed the observers to better express their judgment by increasing the quality scale range. It is important to note that each experiment corresponded to a "blind" test: the observers were always unaware of the synchronization method they were asked to observe to reduce any bias in the simulation results. As the value selected by each observer indicates a global perception of quality, it is difficult to discriminate the main factor influencing his/her decision on the perception of synchronization, audio quality, and video quality. However, this allows us to determine whether the proposed solution for synchronization is comparable to Bertoglio's method in terms of perceptual quality. The experiment was carried out with various synchronization methods employing four different thresholds for $\eta_+$ and $\eta_-$ ($\eta_+ = \eta_- = 30$ ms, 50 ms, 100 ms, and 150 ms). The used threshold value was unknown to the observers. All the evaluation results of this subjective test, based on the impairment score shown in Table 2, were averaged for each threshold condition. All the averaged scores are presented in Table 3. All observers always agreed on the threshold condition exhibiting a better synchronization. The observers showed a preference for the test with lower threshold values of 30 ms and 50 ms. Clearly, by lowering the threshold value to some extent, it can be expected that the synchronization process improves the perceptual quality of the transmission in a proportional way. However, we found that when the temporal skew was less than 50 ms, the observers

Table 2. Impairment score.

| Impairment class | Score |
|---|---|
| Not noticeable | 1 |
| Just noticeable | 2 |
| Definitely noticeable but only slight impairment | 3 |
| Impairment not objectionable | 4 |
| Somewhat objectionable | 5 |
| Definitely objectionable | 6 |
| Extremely objectionable | 7 |

Table 3. Comparison of averaged impairment scores for various synchronization methods.

| | Threshold values for $\eta_+$ and $\eta_-$ | | | |
|---|---|---|---|---|
| | 30 ms | 50 ms | 100 ms | 150 ms |
| Without synchronization | 6.46 | | | |
| Bertoglio's method | 1.71 | 1.86 | 2.74 | 4.19 |
| Proposed method | 1.64 | 1.88 | 2.57 | 4.31 |

could hardly recognize the difference in perceptual quality. Therefore, the most reasonable threshold value for high quality audio-visual transmission service would be around 50 ms. The overall impairment scores for the two methods show similar results for the various threshold values. The subjective evaluation based on impairment score might not show clear superiority to Bertoglio's method, because the basic approaches of the two synchronization methods based on using the RTCP SR packet are very similar. However, the advantage of the proposed method is that it can achieve accurate synchronization with much less computation and processing than Bertoglio's method.

To demonstrate the computational effectiveness of the proposed method, we compared the required clock cycles for the proposed method and Bertoglio's decision rules. For this comparison, a commercially available TI OMAP 1510 multimedia processor was employed to process RTP and RTCP packets in the ARM 925T processor of the OMAP 1510. While media codecs, such as H.264, SVC, and AAC, could be operated on the TMS320C5510 DSP side of the OMAP processor, protocols like RTP and RTCP are usually operated in the ARM side. The commercial real-time operating system Nucleus was ported on the ARM 925T processor of the OMAP 1510. In the implemented system, RTP and RTCP were processed in a single Nucleus task. Table 4 compares the required clock cycles for each decision rule when applied to a pair of audio and video RTP packets in the ARM 925T processor. As shown in this table, the proposed method requires far less computation than Bertoglio's method. From the proposition in section IV, we can easily find that, in the case of the proposed method, we only need to perform one subtraction, two multiplications, and two comparisons. However, we need to perform three subtractions, two divisions, two additions, and two comparison operations in the case of Bertoglio's method. As shown in Table 4, we averaged the number of clock cycles for Bertoglio's method. The number of clock-cycles required for a division operation varies, since it is implemented as a numerical software routine in the case of ARM processors. Table 4 compares the number of clock cycles required for the decision rules shown in (5) and (10). As for the computational complexity to process RTCP SR packets, the proposed method does not require further processing of the RTCP SR packets except the first one. Considering that Bertoglio's method needs to process every RTCP SR packet arriving at the client, the computational savings of the proposed method in RTCP packet processing become greater in proportion to the number of saved RTCP SR packets processed for synchronization.

Table 4. Comparison of clock cycles required to process a pair of RTP packets using different decision rules.

| Decision rule | Bertoglio's | Proposed |
|---|---|---|
| Required clock cycles | 73 cycles | 9 cycles |

## VI. Conclusion

In this paper, we addressed the problem of synchronization for SVC video transport over IP networks. The synchronization issue includes layer synchronization among scalable layers of SVC video and inter-media synchronization between audio and SVC video. We first discussed the suitability of non-interleaved mode RTP packetization to provide layer synchronization among scalable layers of SVC video. Then, we proposed a computationally simple RTCP packet processing method for inter-media synchronization. The proposed method has three main advantages. First, the decision rule is far simpler than that of Bertoglio's conventional method. Second, it does not require RTCP SR packet processing for synchronization except for the first RTCP packet. Finally, the proposed method does not suffer from the accumulation of round-off errors that are inherent in previous methods such as Bertoglio's. An important issue as a future work to further improve the performance of the synchronization process is to take into account buffering strategy to cope with packet inter-arrival jitter.

## References


[1] ISO/IEC JTC1/SC29/WG11, 14496-10 AVC, *Advanced Video Coding for Generic Audiovisual Services*, Mar. 2005.

[2] T. Wiegand, G. Sullivan, J. Reichel, and M. Wien, *Joint Draft 9 of SVC Amendment*, Joint Video Team, JVT-V201, Marrakech, Morocco, Jan. 2007.

[3] J. Reichel, H. Schwarz, and M. Wien, *Joint Scalable Video Model JSVM-9*, Joint Video Team, JVT-V202, Marrakech, Morocco, Jan. 2007.

[4] 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, Sept. 2007, pp. 1103-1120.

[5] D. Wu, Y. Hou, and Y. Zhang, "Transporting Real-Time Video over the Internet: Challenges and Approaches," *Proc. IEEE*, vol. 88, no. 12, Dec. 2000, pp. 1855-1877.

[6] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, IETF RFC 3550, STD 64, July 2003.

[7] J. Postel , "User Datagram Protocol," *IETF STD 0006*, Aug. 1980.

[8] L. Bertoglio and P. Migliorati, "Intermedia Synchronization for Video Conference over IP," *Signal Processing: Image Communication*, vol. 15, no. 1, 1999, pp. 149-164.

[9] S. Wenger, M. Hannuksela, M. Westerlund, and D. Singer, *RTP Payload Format for H.264 Video*, IETF RFC 3984, Feb. 2005.

[10] S. Wenger, Y. Wang, and T. Schierl, "RTP Payload Format for H.264/SVC Scalable Video Coding," *Journal of Zhejiang Univ.*, vol. 7, no. 5, May 2006, pp. 657-667.

[11] 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.

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

[13] R. Steinmetz, "Human Perception of Jitter and Media Synchronization," *IEEE Journ. Selected Areas in Comm.*, vol. 14, no. 1, 1996, pp. 61-72.

[14] T. Thang, Y. Kim, Y. Ro, J. Kang, and J. Kim, "SVC Bitstream Adaptation in MPEG-21 Multimedia Framework," *Journal of Zhejiang Univ.*, vol. 7, no. 5, May 2006, pp. 764-772.

[15] T. Thang, Y. Kim, J. Kang, Y. Ro, and J. Kim, *SVC Video Adaptation with MPEG-21 DIA Adaptation QoS*, ISO/IEC JTC1/SC29/WG11 m12638, Nice, France, Oct. 2005.

**Kwang-Deok Seo** received the BS, MS, and PhD degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1996, 1998, and 2002, respectively. From August 2002 to February 2005, he was with LG Electronics. Since March 2005, he has been a faculty member of the Computer and Telecommunications Engineering Division, Yonsei University, Gangwon, Korea, where he is an associate professor. His current research interests include scalable video coding, protocol design for scalable video transport, and scalable video streaming system design. He is a member of KICS, IEEE, and IEICE.

**Soon-Heung Jung** received the BS degree in Electronic engineering in 2001 from the Pusan National University, Pusan, Korea. He received the MS degree in electronic engineering in 2003 from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. From 2003 to 2005, he was a research engineer in LG Electronics, Korea. Since 2005, he has been a member of engineering staff of the Broadcasting and Telecommunications Media Research Department of ETRI, Korea. His research interests are in the areas of visual communications, video signal processing, video coding, and digital broadcasting.

**Jin-Soo Kim** received the BS degree in electronics engineering from Kyungpook National University, Daegu, Korea, in 1991, and the MS and PhD degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1993 and 1998, respectively. From 1998 to 2000, he was with the Business Division of System LSI in Samsung Electronics, where he was involved in the development of MCU chipsets. Since March 2000, he has been a faculty member in the School of Information Communication and Computer Engineering, Hanbat National University, Korea, where he is an associate professor. His research interests include scalable video coding, networked video-rate shaping and adaptation, media convergence, and multimedia re-multiplexing.