

A Synchronization Scheme for Hierarchical Video Streams over Heterogeneous Networks

Yejin Sohn¹, Minju Cho², Minjae Seo², and Jongho Paik²

¹Department of Contents Design, Seoul Women's University, Seoul, Korea
[e-mail: yjsohn@swu.ac.kr]

²Department of Multimedia, Seoul Women's University, Seoul, Korea
[e-mail: cmj1228, seominjae, paikjh@swu.ac.kr]

*Corresponding author: Jongho Paik

*Received March 10, 2015; revised July 5, 2015; accepted July 23, 2015;
published August 31, 2015*

Abstract

Owing to the increase in consumption of multimedia content and the improvement of device capacity, user demand for high-quality content has increased. However, it is difficult to transport such large amounts of content over the existing broadcasting network with limited bandwidth. To provide high-definition broadcasting, some studies suggest methods of transporting multimedia over heterogeneous networks after encoding content hierarchically. MPEG Media Transport (MMT), standardized by Moving Picture Experts Group (MPEG), is a solution that enables large-volume media transport over heterogeneous networks such as digital broadcasting networks and packet-switched networks. In the case of delivering a scalable encoded video over different networks, synchronization of each stream is an important issue. MMT defines a synchronization scheme, but does not contain sufficient functions to implement it. In this paper, we propose a synchronization scheme for media streams that are encoded hierarchically, divided into layers, and transported over heterogeneous networks. We implement our scheme using MMT and HTTP, and experimented using three encapsulated video streams with different durations. As a result, we show that the proposed scheme can reduce the waiting time to display high-quality video, relative to Dynamic Adaptive Streaming over HTTP-Scalable Video Coding (DASH-SVC) by requesting segments of enhanced layers after calculating the transmission time. Additionally, we find out that the selection of durations have a relation to the characteristics of the video.

keywords: Hierarchical video streaming, MMT, DASH-SVC, ISOBMFF

This research was supported by the ICT R&D program of MSIP/IITP. [R0101-14-0189, Development of the next generation convergence broadcasting and monitoring systems combined with the networks]

This work was supported by a special research grant from Seoul Women's University(2015).

A preliminary version of this paper was presented at ICONI 2014, and was selected as an outstanding paper.

1. Introduction

Recently, digital broadcasting services have become increasingly common as display performance and video compression efficiency have been improved. Additionally, users have convenient access to a variety of content because a wide variety of terminals and improved network performance are available. Following these changes, users are requesting higher quality content. A media segmented at a fixed size can be transported using the MPEG-TS (Transport Stream) which is a traditional transmission technique. However, they are not suitable for transporting high-quality content because of their low transmission efficiency. High-quality content provides about 4 to 16 times better definition than that provided by high definition (HD), and therefore, it requires additional bandwidth. To meet these requirements, digital broadcasting is steadily being developed [1].

MPEG Media Transport (MMT) was standardized so as to replace MPEG-2 TS by MPEG in 2014. MMT defines three functional layers that are independent from video codecs and transport networks- media processing unit (MPU) format, signaling messages, and delivery protocol [2][3]. The MPU layer includes ISO base Media File Format (ISO BMFF), which it encapsulates in variable-length boxes. This layer can further improve the transmission efficiency relative to TS. However, the problem of providing content adaptively to diverse terminals over the existing broadcasting network, which has limited bandwidth, remains. To solve this problem, MMT allows for several hybrid service models to provide content over heterogeneous networks such as broadcasting networks and broadband networks at the same time [4]. The first model is one that transports different components of the content for live broadcasting. The second model is a hybrid transport scenario that transports stored components and live components. The third model divides the components of the content and transports them over different heterogeneous networks.

As mentioned above, high-quality video is too large to be transported over the existing broadcasting network; therefore, the third service model is a useful method for transporting it. This service model is feasible with Scalable Video Coding (SVC) and Scalable High Efficiency Video Coding (SHVC). SVC arranges video layers in a hierarchy for temporal, spatial, and quality scalability. SVC encodes a video into a base layer and more than one enhancement layer; therefore, it is also called 'hierarchical video coding'. This method arranges video layers in a hierarchy; thereafter, they can be split or merged.

Recent studies have proposed many methods for transporting media based on MMT using SVC. [5] encoded hierarchical video using SVC for UHD transmitted content and proposed a broadcasting transmitter/receiver system for hierarchical layers over heterogeneous networks. [6] designed a NIT table, which is an extension of signaling table in MMT, to allow receipt of hierarchical video streams over heterogeneous networks. [7] designed and implemented a player that can store and access each MPU file of the content, following the method suggested by MMT. Likewise, there are many studies that transport video streams divided hierarchically over heterogeneous networks; however, studies for synchronization schemes are insufficiently able to allow the receiver to stably play the divided video streams. Each divided hierarchical stream is transported over a heterogeneous network that has different characteristics in terms of bandwidth, delay, transmission time, and so on. Furthermore, there is very little information regarding the use-case of the suggested service model in MMT. In this paper, we propose a synchronization scheme for video streams transported over broadcasting networks and networks based on HTTP, which is the third hybrid delivery service model suggested by MMT.

Additionally, we analyze major influencing factors such as file size and transport unit length for the stable services through experiments.

The remainder of this paper is organized as follows. Section 2 addresses two related techniques. Section 3 presents the background of the proposed scheme and explains the process of media-stream synchronization. To evaluate this scheme, we explain the environment of the test in section 4. Experimental results are provided in section 5. Finally, we present conclusions and discuss the remaining issues for future research in section 6.

2. Related Work

2.1 MPEG Media Transport (MMT)

MMT, specified as ISO/IEC 23008-1 (MPEG-H Part 1), is a media container standard developed by MPEG. MMT is a technology for transporting coded media data over heterogeneous packet-switched networks including IP networks and digital broadcasting networks; it includes three functional layers: delivery protocol, signaling messages, and the MPU format. The delivery functional layer defines an application layer transport protocol and a payload format. The signaling message functional layer manages delivery and consumption of media data and defines table formats such as MMT presentation information table (MPIT). The MPU format based on ISO/BMFF defines the logical structure of the media content, the package, and the format of the data units for encapsulating the encoded data. In addition to those three functions, there is a composition information (CI) layer. This layer, specified using HTML5 and XML documents, controls the temporal and spatial layout of media. CI file (written by XML) provides information on temporal relationships among MPUs within a single package to complement the associated HTML5 document. A HTML5 file is referred to as a presentation information (PI) which provides initial information on spatial relationships among media elements[2].

Fig. 1 below shows the relationship among the three layers of MMT. Because the signaling messages include a PI document, the receiver is able to obtain location information of the media data in advance. Actual media data can be obtained by the encapsulation layer. A media data consists of several boxes in MPU format. A ‘moof’ box contains information regarding media data and an ‘mdat’ box contains binary data regarding the actual media. The unit, which combines a ‘moof’ box and an ‘mdat’ box, can be split up into durations of several seconds[8]. Media delivered through HTTP and broadcasting networks can be presented along with information described in the PI.

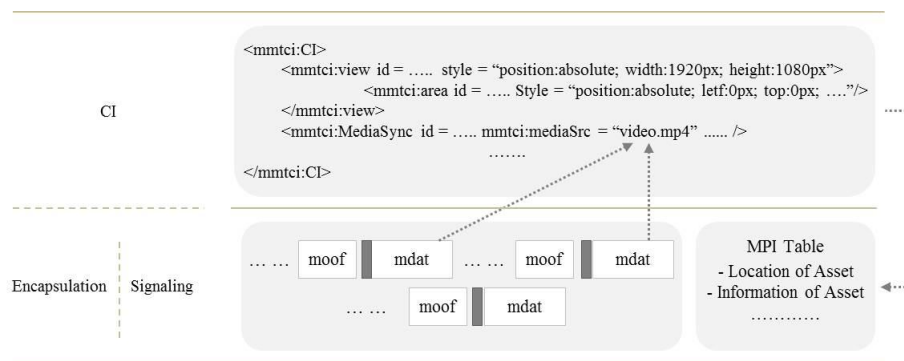


Fig. 1. The relationship among three layers of MMT

2.2 Hierarchical Video Streaming over HTTP (DASH-SVC)

DASH is an MPEG standard that defines a format that can be delivered over HTTP. This method divides encoded media files into segments of a certain period and stores them in a server to allow the network environment to quickly adapt. DASH consists of media segments and MPD, which contains the lists of media segments. MPD is a kind of metadata used to generate a URL that enables a client to locate content. MPD has elements and attributes to manage information regarding media content. Each period contains MPD divided by the specified time interval. *AdaptationSet* contains information regarding media content in one period. *Representation* has more than one media stream included in *AdaptationSet*. Each piece of content is classified according to quality and managed with an attribute value called *id*. [9] proposed a streaming service to provide hierarchical video based on DASH using SVC. The enhancement layer contains *dependencyId*. If there is no media that matches the referenced *id* attribute, the desired media cannot be played. If an enhancement layer refers to media that has a *dependencyId*, it must include the *id* of the referenced media and also the *dependencyId* attribute value in its own *dependencyId*.

Fig. 2 shows the difference between DASH and DASH-SVC. In the case of representation, the using a *id* set to tag1_2, it is an enhancement of tag1; it refers to tag1 using a *dependencyId*. Tag1_2 cannot be played if tag1 does not exist. In case of the tag1_3, the method refers to tag1_2, which obtains tag 1 as the *dependencyId*; tag1_3 must include tag1_2 and tag1 in its *dependencyId*. [10] considers the multiple connection environment of [9]. This method can provide the proper size of media based on the user network environment. If SVC service is provided as in this method, it can provide media for many users in different environments and both minimize traffic and maximize efficiency.

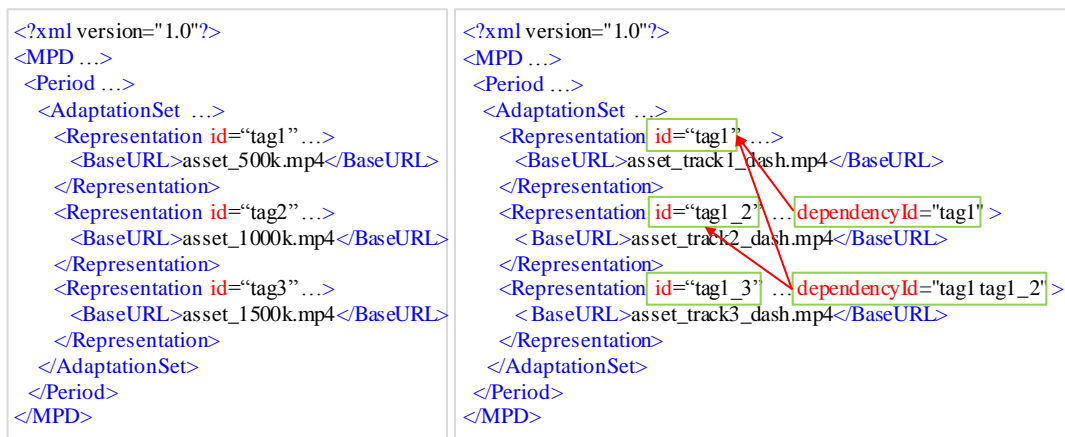


Fig. 2. The difference between DASH and DASH-SVC

3. Proposed Scheme

3.1 Background

Fig. 3 presents the process of high-quality service over heterogeneous networks. Video content is arranged in a hierarchy by a content generator. The base layer can be transported over broadcasting networks. The enhancement layer can be transported using a web server. If users can receive data over heterogeneous networks, high-quality media services can be made

available, by merging the base and enhancement layers over the heterogeneous networks.

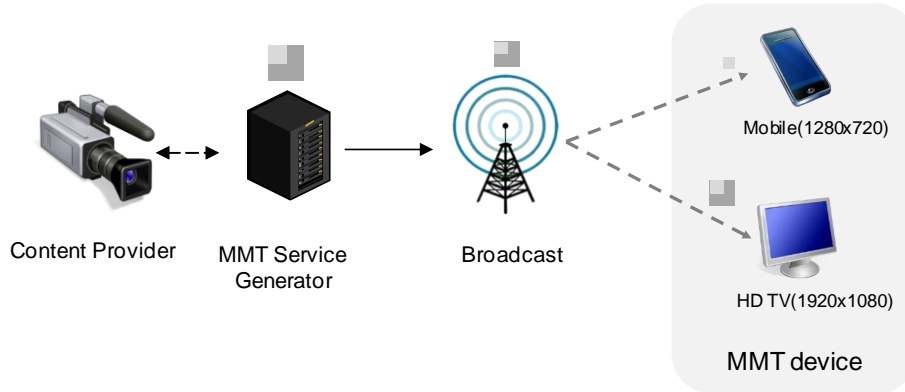


Fig. 3. Architecture of high-quality services over heterogeneous networks

3.2 Generating media segments

Each encoded layer of video should be encapsulated in a segment for transmission. The following paragraph describes the process of segmenting a video into two layers. When the number of layers increases, the number of sets also increases.

The hierarchically- encoded video consists of a series of pictures; it is divided into a base layer and an enhancement layer. A single encoded video can be represented as a set $P, P = \{P_0, P_1, P_2, \dots, P_n\}$, where the sum of the elements of set P is the one encoded content. Likewise, each picture set of the base layer and enhanced layer is represented as $B = \{B_0, B_1, B_2, \dots, B_n\}$ and $E = \{E_0, E_1, E_2, \dots, E_n\}$, respectively. n signifies the total number of pictures in the content, which is equivalent to the total number of elements of all sets. B and E are included in P , each set is exclusive as below (1).

$$\begin{aligned} B &\subset P, & E &\subset P \\ (B \cup E) &= P, & B \cap E &= \emptyset \end{aligned} \quad (1)$$

Pictures in the base layer and the enhanced layer are separate data from pictures that are elements in P . Therefore, the sum of all elements of B and E is same as the sum of all elements in the encoded video, as follows in (2).

$$P = \sum_{i=0}^n (B_i \cup E_i) \quad (2)$$

Segments are divided based on time (in seconds). The duration is represented by d ; the duration value can vary according to the user's definition. In addition, the total play time of the content is defined as T seconds; the number of frames per second is defined as f . The total number of frames (N) is the product of T and f . Additionally, we can calculate the number of segments (m) that can be generated for one piece of content using (3).

$$m = \lceil N/d \cdot f \rceil \quad (3)$$

Sets B and E consist of sets of several segments; they can be represented by (4).

$$\begin{aligned} B = \text{Segment}_B &= S_{(B,1)} \cup S_{(B,2)} \cup S_{(B,3)} \cup \dots \cup S_{(B,m)} \\ E = \text{Segment}_E &= S_{(E,1)} \cup S_{(E,2)} \cup S_{(E,3)} \cup \dots \cup S_{(E,m)} \end{aligned} \quad (4)$$

The elements in sets B and E can be represented by $S_{(x,y)}$, another set. $S_{(x,y)}$ is the set of pictures in the n^{th} segment; its range is determined by d and f , as shown in equation (5).

$$\begin{aligned} S_{(x,y)} &= \{x_{(y-1)d \cdot f}, \dots, x_{y \cdot d \cdot (f-1)}\} \\ x &\in \{B, E, P\}, y \in \{y | 1 \leq y \leq m\} \end{aligned} \quad (5)$$

Additionally, as shown in (6), the sum of the segment sets of each layer is the same as that of the encoded video.

$$P = \sum_{i=1}^m (S_{(B,i)} \cup S_{(E,i)}) \quad (6)$$

3.3 Documents for synchronization

To transport segments, a particular document, with storage location and quality information regarding the segments, should be sent in advance. DASH-SVC transmits a list of media segments with documents called MPD, and transports the hierarchical video information using the *dependencyId*. This method can provide different-quality video services without requiring the configuration of a variety of representations. PI, which is the configuration information used in MMT, consists of information regarding media composition and media content locations on screen for use in the playback of multimedia content. Content information and the document are transmitted to the MMT signaling to provide the service to the user. In the case of hierarchical video content, the document defines which media information should be transported using *dependencyId* for extension information. In this paper, we define MMT eXtension Document (MXD), which is similar to MPD, to synchronize content over heterogeneous networks, and organize content synchronization information by inserting MXD information into the PI. The MXD and CI configurations used in this paper are shown in Fig. 4.

```
<mmtci:CI>
  <mmtci:view id="View1" style="position:absolute; width:1920px; height:1080px"
    mmtci:begin="0s" mmtci:end="indefinite" >

    <mmtci:divLocation id="divL1" style="position:absolute; left:0px; top:0px;
      width:1920px; height:1080px"
      mmtci:begin="0s" mmtci:end="indefinite"
      mmtci:refDiv="Area1" />

  </mmtci:view>
  <mmtci:MediaSync id="MediaSync1" mmtci:isReferenced="TRUE"
    mmtci:refId="mmt://asset1.mp4" mmtci:depId="http://Lab/ABC/abc.mxd" />
</mmtci:CI>
```

Fig. 4. The examples of MMT CI

Fig. 4 presents examples of CI configuration information within MMT. CI delivers the information whether or not the hierarchical video is used with the value of *isReferenced* in the *MediaSync* element. MMT essentially includes base-layer content of the hierarchical video. If there is an enhancement layer, it delivers asset information regarding the base layer content based on *refId* and MXD information regarding enhancement layer extended based on *depId*. The *refId* and *depId* attributes include a physical address transmitted by the uniform resource identifier (URI) format and the name of the content. **Fig. 5** shows the configuration information of the MXD, which is composed of only enhancement content. MXD uses the same reproduction time and segments as the content of the base layer of the MMT. *BaseURL* in MXD shows the location and name of the base layer content, and *dependencyId* indicates the asset id of the content in the base layer. The segment list contains information obtained by dividing the enhancement layer into the same times as the base layer content. *mediaRange* indicates the start address of the segment or the m4s list that is divided into a segment file such as MPD in DASH.

```

<?xml version="1.0"?>
<!-- MXD file Generated with ABC-->
<MXD minBufferTime="PT1.500000S" type="static" mediaPresentationDuration="PT0H2M52.80S" >
  <Period duration="PT0H2M52.80S">
    <AdaptationSet segmentAlignment="true" bitstreamSwitching="true" maxFrameRate="25" par="16:9"
      maxWidth="1920" maxHeight="1080">
      <Representation id="1_2" mimeType="video/mp4" codecs="svc1.53c033"
        width="1920" height="1080" frameRate="24" bandwidth="5477857" dependencyId="asset1">
        <BaseURL>mmt://asset1.mp4 </BaseURL>
        <SegmentList timescale="25000" duration="49655">
          <SegmentURL mediaRange="805-34778" indexRange="805-848"/>
          <SegmentURL mediaRange="34779-61693" indexRange="34779-34822"/>
          ⋮
          <SegmentURL mediaRange="69294434-69358729" indexRange="69294434-69294477"/>
        </SegmentList>
      </Representation>
    </AdaptationSet>
  </Period>
</MXD>

```

Fig. 5. An example of the proposed CI (MXD)

3.4 An algorithm for selecting segments at the receiver

Section 3.3 presented steps to prepare the segment for transmission. This section introduces an algorithm for selecting the appropriate segment, utilizing each XMD of the base and the enhancement layers at the receiver. It is important to select an appropriate segment to stably provide the high quality of service desired by users[11][12][13]. The separated video streams are combined into the original video in the environment as quickly and accurately as possible to allow separation of content and transmission over networks with different characteristics.

Fig. 6 shows pseudo-code of the process from the time that the user requests high-quality service to the time that segments with the same sequence number of $Segment_B$ and $Segment_E$ are input into the decoder.


```

#define d SECONDS // SECONDS: User define value,  $d > 0$  unsigned integer
if(extended_indicator==TRUE)
{
    //get_enhanced_content_list; MXD file
    request MXD file:
        SegmentE = {S(E,1), S(E,2), ..., S(E,m)};
    //calculate current network bandwidth at the first time
    BW = MXDSize/t
    τ=get_currentNumber(SegmentB);
    //the size of S(E,τ)
    s=request_Enhanced_Segment(τ);

    //calculate the expected download time of s
     $\hat{T} = \frac{s}{BW}$ ;

    //request and store enhanced segment;
    if( $\hat{T} \leq d$ )
        request_Enhanced_Segment and store( $\hat{t} = \tau + 1$ );
    else
        request_Enhanced_Segment and store( $\hat{t} = \tau + \lceil \hat{T}/d \rceil$ );
}
while(τ<MAX_SEG_NUM)
{
    if(τ== $\hat{t}$ )
        decode(τ,  $\hat{t}$ );
    else
        decode(τ);
}

```

Fig. 6. The pseudo code of the proposed selection algorithm

If the user requests the enhanced layer, the receiver requests the MXD of the enhanced layer after the location and size of the MXD are obtained from the CI. The bandwidth of the current network (BW) can be calculated based on the size of the MXD and download time (t). To calculate \hat{T} , which is the expected download time of $S_{(E,\tau)}$ for BW , the size of $S_{(E,\tau)}$ should be obtained from MXD. \hat{T} has an effect on which segment is selected in the enhanced layer. \hat{t} is decided by comparing \hat{T} to d . If \hat{T} is less than d , this method immediately requests the next segment; if it is larger than d , \hat{t} is selected based on (7).

$$S_{(E,\hat{\tau})} = \begin{cases} \hat{t} = \tau + 1, \hat{T} < d \\ \hat{t} = \tau + \lceil \hat{T}/d \rceil, \hat{T} \geq d \end{cases} \quad (7)$$

4. Experimental Environment

4.1 Test scenario

Fig. 7 shows a service scenario that transmits high-quality video over a broadcast network and a broadband network, enabling stable service via a synchronization scheme. For the test of the proposed synchronization scheme, we designed a service model using software. Our test

environment over heterogeneous networks is replaced by two LANs with different protocols: MMT and HTTP. These are assumed to be error-free networks.

A hierarchically encoded video is separated into multi-layered-streams by the content generator. Then each layered stream is transmitted respectively over different networks shown in Fig.7. The PI and the stream for the base layer are delivered via the broadcast. PI contains essential information such as the content resolution, location of content, and MXD.

When users request high-quality media, devices should first obtain the MXD. The receiver selects the proper segment of the enhanced layer using the proposed synchronization scheme.

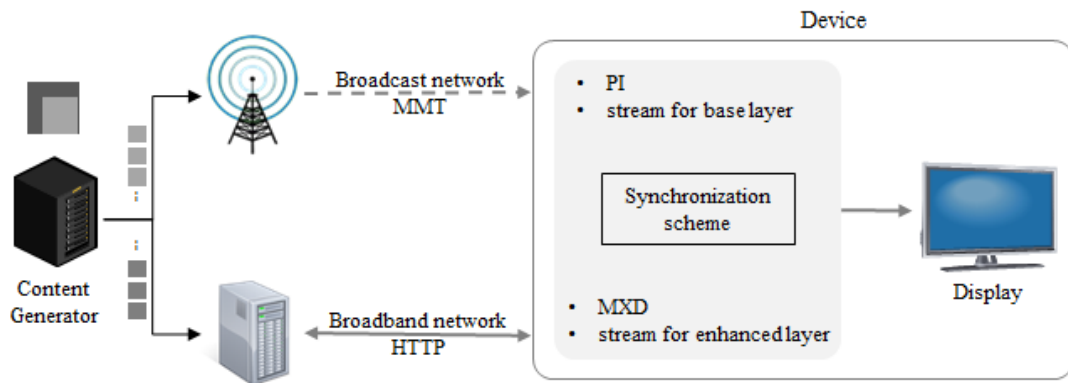


Fig. 7. Service scenario that transmits high-quality video over heterogeneous networks

4.2 Configuration

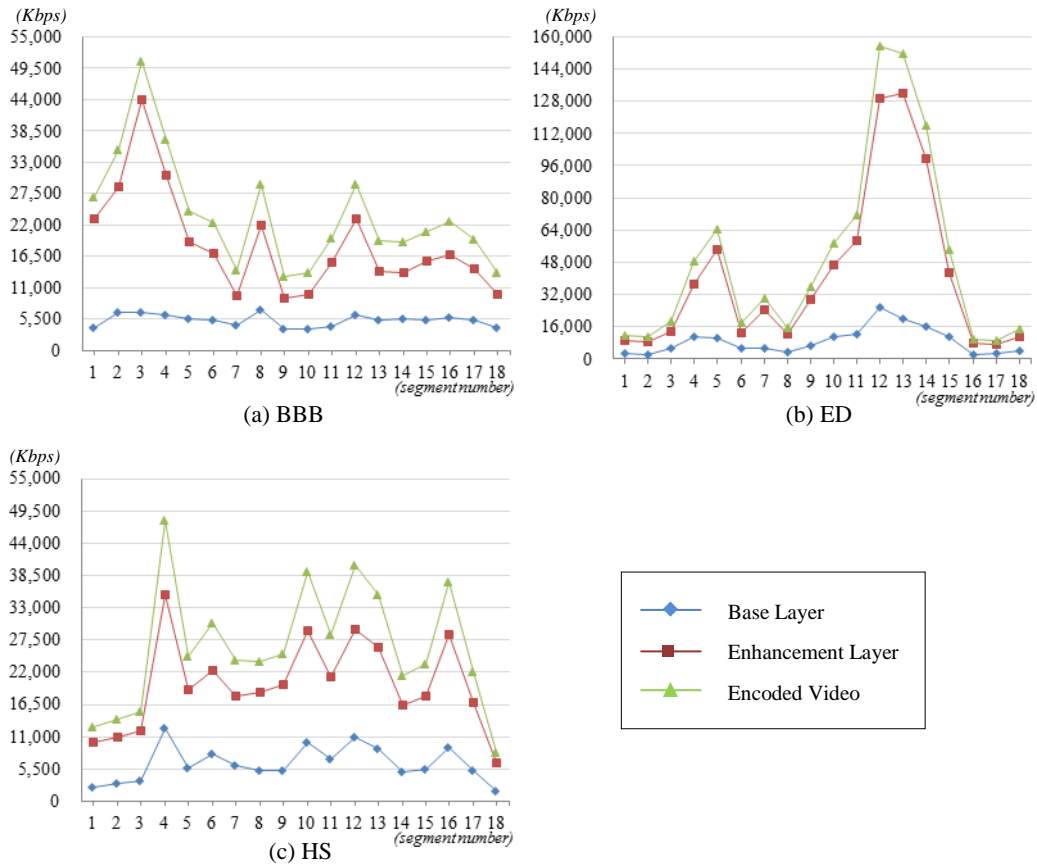
Before evaluating the proposed scheme, we implemented the content generator and the receiver software using Visual Studio 2010. Videos called Big Buck Bunny(BBB), Elephants Dream(ED) and Heli-Ski(HS) are encoded using JSVM(SVC reference software Joint Scalable Video Moder) for hierarchical video encoding in this test[14]. Each video encodes a base layer (BL) and one enhanced layer (EL), according to information given in Table 1. The playtime of each video is one hundred eighty seconds.

We make segments of duration(d), setting the d variable values to 1, 2, 4, and 10 seconds. Fig. 8 shows the required bitrates for delivering each segment of encoded video described in Table 1. The x-axis presents the segment number, as calculated using equation (3). The green line (labelled Encoded Video) in Fig. 8 indicates the sum of bitrates per segment, based on equation (6).

Table 1. Encoding information

Encoding option	Big Buck Bunny	Elephants Dream	Heli-Ski
BL-resolution	720x480	720x480	720x480
EL-resolution	1920x1080	1920x1080	1920x1080
fps	24	24	24
GOP/IDR/Intra	16/16/16	16/16/16	16/16/16
Presentation time(s)	180	180	180
duration(s)	1/2/4/10	1/2/4/10	1/2/4/10

Fig. 8. The bitrates of sample videos



5. Experimental Results

This section presents the experimental results regarding the proposed synchronization scheme. We compare DASH-SVC to the proposed scheme to evaluate the performance of this scheme. We choose three videos with different characteristics, namely, BBB, ED, and HS, which were introduced in section 4. BBB and ED are animations; however, ED is more dynamic than BBB. HS is a live-action video.

We assume that segments of BL and EL are stored in each input buffer as soon as they arrive at a receiver. If the streams of BL and EL are synchronized, the encoded video should be pushed to the decoding buffer. If not, only BL should be used. Fig. 9 shows the bitrates that were measured at the decoding buffer. In this test, d is set to two seconds and BW is calculated to be 7,500Kbps. As a result of streaming using DASH-SVC and the proposed scheme, only BL arrives at the decoding buffer in the case of DASH-SVC[15][16][17]. DASH-SVC cannot synchronize the BL and EL segments because it selects the incorrect segment of EL. In contrast, our scheme correctly selects the segment based on the equation (7).

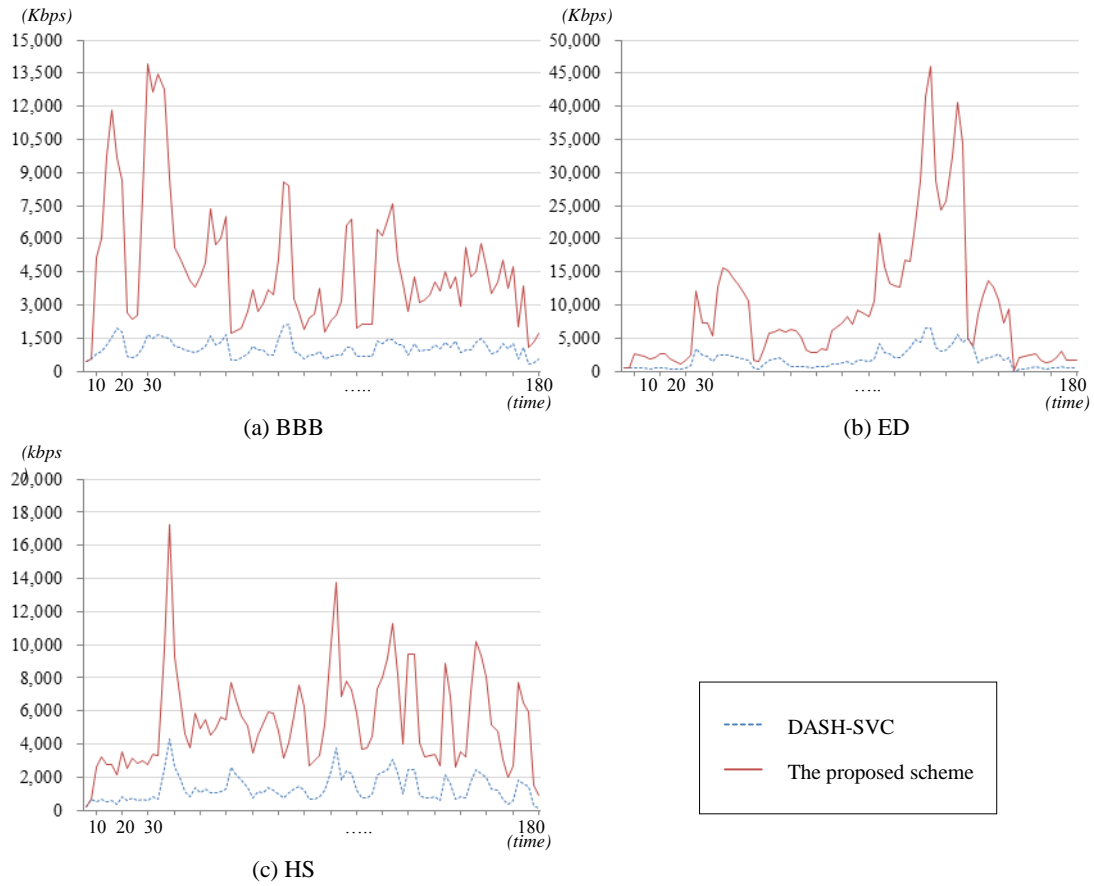


Fig. 9. A performance comparison between DASH-SVC and the proposed scheme

Table 2. The synchronization process of Fig. 9 (a) according to running time

	running time	BL buffer	EL buffer	request	receiving time	decoding buffer
	-	-	-	-	-	$S_{(B,3)}$
DASH-SVC	00:08	$S_{(B,4)}$	-	$S_{(E,4)}$	0.2s	$S_{(B,4)}$
Propose scheme		$S_{(B,4)}$	-	$S_{(E,5)}$	0.4s	$S_{(B,4)}$
DASH-SVC	00:10	$S_{(B,5)}$	$S_{(E,4)}$	$S_{(E,5)}$	0.4s	$S_{(B,5)}$
Propose scheme		$S_{(B,5)}$	$S_{(E,5)}$	$S_{(E,6)}$	0.2s	$S_{(P,5)}$
DASH-SVC	00:12	$S_{(B,6)}$	$S_{(E,5)}$	$S_{(E,6)}$	0.2s	$S_{(B,6)}$
Propose scheme		$S_{(B,6)}$	$S_{(E,6)}$	$S_{(E,7)}$	0.8s	$S_{(P,6)}$

Table 2 shows the status of BL, EL, and the decoding buffers according to the running time for the BBB video. The running time gradually increases by two seconds, according to d . The segment to be consumed is pushed to the decoding buffer, this segment can be $S_{(B,x)}$ or $S_{(P,x)}$ (x indicates the x^{th} segment). For example, DASH-SVC requests $S_{(E,4)}$, while $S_{(B,4)}$ stays in the BL buffer at eight seconds. However, the proposed scheme requests $S_{(E,5)}$ after the calculation. $S_{(P,5)}$, including EL segments, is displayed in the case of the proposed scheme, whereas DASH-SVC can display only $S_{(B,5)}$ at ten seconds. Therefore, selection of the segment number is an important function for supporting high quality content delivered over broadcast and broadband networks.

Table 3. The receiving results according to d values

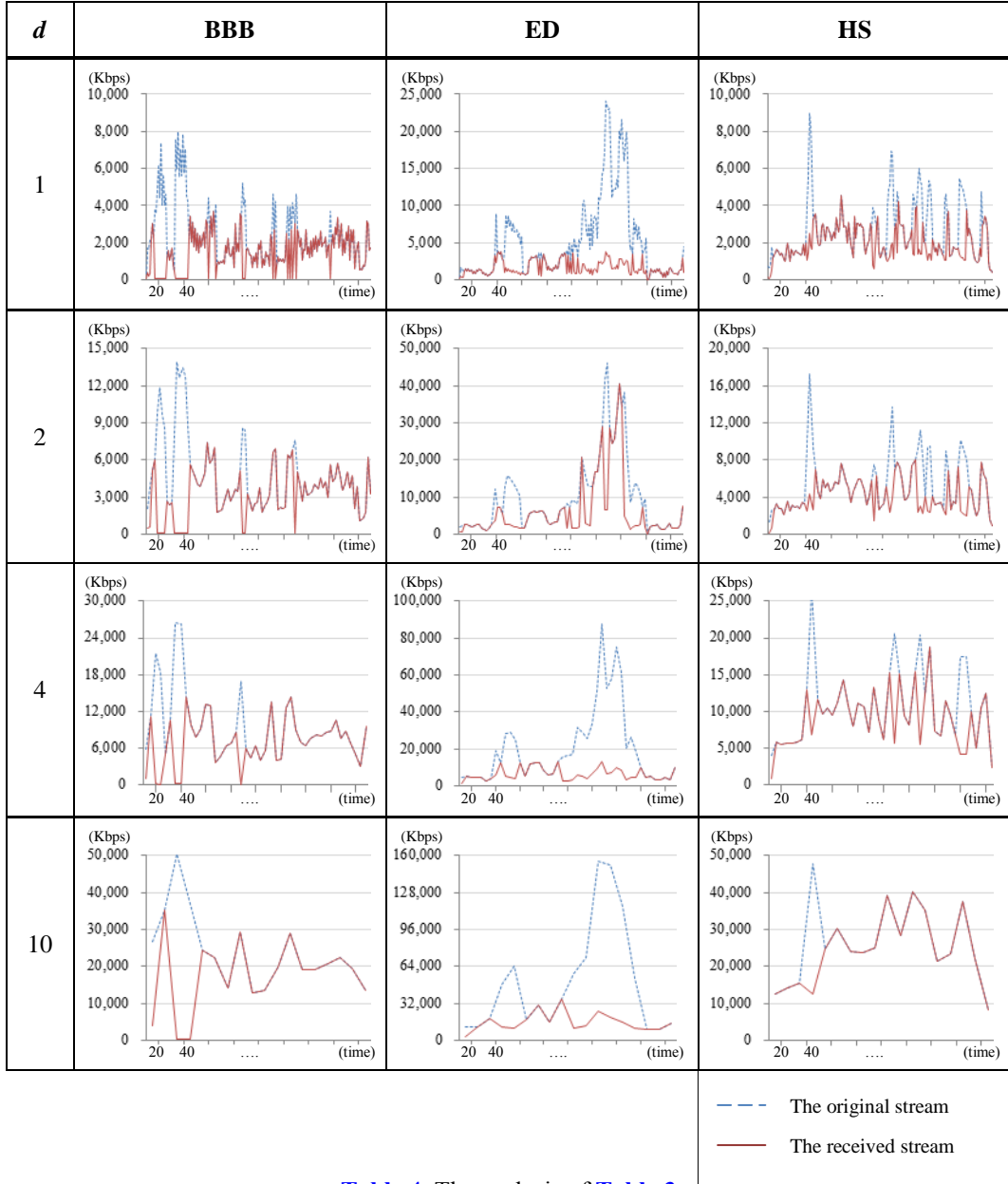


Table 4. The analysis of Table 3

d	BBB		ED		HS	
	Play time only BL	Switching count	Play time only BL	Switching count	Play time only BL	Switching count
1s	32	24	77	24	33	20
2s	26	8	52	16	30	14
4s	20	6	60	10	20	8
10s	20	2	50	6	10	2

Table 3 presents the results of the experiments, which applied various d values to each encoded video. Time is indicated along the horizontal axis of the graph; the bitrate for each segment is indicated along the vertical axis. The blue line signifies the original encoded stream; the red line indicates the stream received by the proposed algorithm. When the blue and red lines overlapped each other, both BL and EL layers are received and decoded. During this period, the high-quality video can be provided to users. According to **Table 3**, we found out that the period of the overlapped line has a relation to the characteristics of the video, not the d values. **Table 4** presents a statistic analysis of the data in **Table 3**. Play time indicates the time that only BL is decoded, without EL. The switching count shows the number of changes in the resolution. We found out that the less d value, the more switching count and play time. Because an amount of received data becomes small when the d value decreases, the receiver adaptively works much more; however, only the time of displaying BL increases.

6. Conclusion

MMT was suggested for use in three service scenarios to deliver content over heterogeneous networks such as existing broadcasting networks and broadband networks. However, it is difficult to transport high-quality content using MMT. To solve this problem, a synchronization scheme is required in addition to video coding and transport techniques. In this paper, we present a service scenario for delivering high-quality contents using MMT and HTTP. We encoded a video hierarchically and extracted BL and EL. Then BL and EL were segmented in time and transported over heterogeneous networks to achieve stable streaming service. In addition, we presented a synchronization scheme for the receiver to generate contents from streams transported over heterogeneous networks. This scheme can provide more stable high-quality content than existing methods by requesting appropriate segments and calculating the segment transmission time. To evaluate our scheme, we experimented with streaming for various durations and found out that duration and video characteristics have an influence on the streaming. In the future, we will study a method to allow the system to adapt to network conditions or video characteristics so as to reduce the switching count.

References

- [1] S.Aoki, K. Otsuki, and H. Hamada, "Effective Usage of MMT in Broadcasting Systems," *Broadband Multimedia Systems and Broadcasting (BMSB)*, In *Proc. of 2013 IEEE International Symposium on*, pp. 1-6, 2013. [Article \(CrossRef Link\)](#).
- [2] ISO/IEC 23008-1, "Information technology — High efficiency coding and media delivery in heterogeneous environments — Part1: MPEG media transport (MMT)", 2014. [Article \(CrossRef Link\)](#).
- [3] ISO/IEC 14496-12:2008, "Information technology - Coding of audiovisual objects - Part 12: ISO base media file format", 2008. [Article \(CrossRef Link\)](#).
- [4] ISO/IEC 23008-13, "Information technology — High efficiency coding and media delivery in heterogeneous environments — Part13: MMT Implementation guidelines", 2014. [Article \(CrossRef Link\)](#).
- [5] Y. Sohn, M. Cho, and J. Paik, "Design of MMT-based Broadcasting System for UHD Video Streaming over Heterogeneous Networks," *Journal of broadcast engineering*, vol. 20, no. 1, pp. 16-25, 2015. [Article \(CrossRef Link\)](#).
- [6] K. Yu, M. Seo, and J. Paik, "Design of MMT Signaling for Streaming Service based on SVC in Mobile Environments," in *Proc. of The 6th International Conference on Internet (ICONI 2014)*, pp. 195-196, 2014.

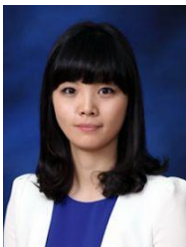
- [7] S. Park, and K. Kim, "Design and Implementation of MPEG-MMT contents player," in *Proc. of Conference of broadcast engineering*, vol. 2013, no. 11, pp. 200-203, 2013. [Article \(CrossRef Link\)](#).
- [8] I. Kofler, R. Kuschnig, and H. Hellwagner, "Implications of the ISO base Media File Format on Adaptive HTTP Streaming of H.264/SVC," in *Proc. of Consumer Communications and Networking Conference*, pp. 549-553, 2012. [Article \(CrossRef Link\)](#).
- [9] Y. Sánchez, C. Hellge, T. Schierl, W. V. Leekwijck, and Y. Louédec, "Scalable Video Coding based DASH for efficient usage of network resources," in *Proc. of The Third W3C Web and TV workshop*, 2011. [Article \(CrossRef Link\)](#).
- [10] H. Kalva, V. Adzic, and B. Furht, "Comparing MPEG AVC and SVC for Adaptive HTTP Streaming," in *Proc. of 2012 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 158-159, 2012. [Article \(CrossRef Link\)](#).
- [11] S. Wei, and V. Swaminathan, "Low Latency Live Video Streaming over HTTP 2.0," in *Proc. of Network and Operating System Support on Digital Audio and Video Workshop*, pp. 37-42, 2014. [Article \(CrossRef Link\)](#).
- [12] Y. Sanchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. D. Vleeschauer, W. Van Leekwijck, and Y. L. Louedec, "Efficient HTTP-based streaming using Scalable Video Coding," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 329-342, 2012. [Article \(CrossRef Link\)](#).
- [13] S. Ibrahim, A. H. Zahran, and M. H. Ismail, "SVC-DASH-M: Scalable Video Coding Dynamic Adaptive Streaming Over HTTP Using Multiple Connections," in *Proc. of Telecommunications(ICT) 2014 21st International Conference*, pp. 400-404, 2014. [Article \(CrossRef Link\)](#).
- [14] M. Grafl, C. Timmerer, H. Hellwagner, W. Cherif, and A. Ksentini, "Evaluation of Hybrid Scalable Video Coding for HTTP-based Adaptive Media Streaming with High-Definition Content," in *Proc. of World of Wireless*, in *Proc. of Mobile and Multimedia Networks (WoWMoM), 2013* in *Proc. of IEEE 14th International Symposium and Workshop on*, pp. 1-7, 2013. [Article \(CrossRef Link\)](#).
- [15] Y. Li, C. Chen, T. Lin, C. Hsu, Y. Wang, and X. Liu, "AN END-TO-END TESTBED FOR SCALABLE VIDEO STREAMING TO MOBILE DEVICES OVER HTTP," *Multimedia and Expo (ICME)*, in *Proc. of 2013 IEEE International Conference*, pp. 1-6, 2013. [Article \(CrossRef Link\)](#).
- [16] C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and User-centric Comparison of a Novel Adaptation Logic for DASH with SVC," in *Proc. of Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium*, pp. 1318-1323, 2013. [Article \(CrossRef Link\)](#).
- [17] C. Müller, D. Renzi, S. Lederer, S. Battista, and C. Timmerer, "USING SCALABLE VIDEO CODING FOR DYNAMIC ADAPTIVE STREAMING OVER HTTP IN MOBILE ENVIRONMENTS," in *Proc. of Signal Processing Conference (EUSIPCO), 2012* in *Proc. of the 20th European*, pp. 2208-2212, 2012. [Article \(CrossRef Link\)](#).



Yejin Sohn received the B.S, M.S. and Ph.D. degrees in Computer Science from Seoul Women's University, Korea, in 2005, 2007, and 2013 respectively. She is currently working as a researcher at the same university. Her research interests include the packet-level transport protocol and FEC mechanisms for the video streaming in broadband and broadcasting networks.



Minju Cho received the B.S, M.S. and Ph.D. degrees in Computer Science from Seoul Women's University, Korea, in 2005, 2008, and 2012 respectively. From 2008 to 2011, she has been an associate researcher engineer with iSET Corp.. Currently, she is a researcher with Seoul Women's University. Her research interests are mobile DTV, convergence computing.



Minjae Seo received the B.S in Computer Science from Seoul Women's University, Korea in 2013. Now she is on Master's course since 2014. Her research interests are mobile DTV and digital communications system.



Jongho Paik received the B.S., M.S., and Ph.D. degrees in the school of Electrical and Electronic Engineering from Chung-Ang University, Seoul, Korea, in 1994, 1997, and 2007, respectively. He was a Director with Advanced Mobile Research Center at Korea Electronics Technology Institute (KETI) by 2011. Since 2011, he is currently an assistant professor in the department of multimedia, Seoul Women's University, Seoul. His research interests are in the areas of web-based communication, software testing, wireless/wired communications system design, video communications system design and system architecture for realizing advanced digital communications system and for advanced mobile broadcasting networks as well.