# A network-adaptive SVC Streaming Architecture

*Peng Chen *Jeongyeon Lim *Bumshik Lee *Munchurl Kim
*Information and Communications University
{chpeng, jylim, bslee, mkim}@icu.ac.kr
**Sangjin Hahm **Byungsun Kim **Keunsik Lee **Keunsoo Park
**Korean Broadcasting System
{casy, bskim2000, kslee22, keunspark}@kbs.co.kr

## Abstract

In Video streaming environment, we must consider terminal and network characteristics, such as display resolution, frame rate, computational resource, network bandwidth, etc. The JVT (Joint Video Team) by ISO/IEC MPEG and ITU - TVCEG is currently standardizing Scalable Video Coding (SVC). This can represent video bitstreams in different scalable layers for flexible adaptation to terminal and network characteristics. This characteristic is very useful in video streaming applications. One fully scalable video can be extracted with specific target spatial resolution, temporal frame rate and quality level to match the requirements of terminals and networks. Besides, the extraction process is fast and consumes little computational resource, so it is possible to extract the partial video bitstream online to accommodate with changing network conditions etc. With all the advantages of SVC, we design and implement a network-adaptive SVC streaming system with an SVC extractor and a streamer to extract appropriate amounts of bitstreams to meet the required target bitrates and spatial resolutions. The proposed SVC extraction is designed to allow for flexible switching from layer to layer in SVC bitstreams online to cope with the change in network bandwidth. The extraction is made in every GOP unit. We present the implementation of our SVC streaming system with experimental results.

## 1. Introduction

In order to deliver video streams over the networks with different bandwidth to a variety of terminals, the conventional streaming systems usually prepare multiple compressed versions of each video to cope with the different characteristics of terminals and networks. Maintaining such multiple versions of videos at the transmission sides usually requires large storage capacity and management resources. To avoid the pitfalls of the conventional methods, SVC is being standardized with the aim at providing a flexible representation of compressed video bitstreams so that the compressed bitstreams can correspond to the changes in delivery and consumption environments [2]. In SVC, the bitstream is represented in salable ways with spatial layer, temporal layer and quality layer. A full SVC bitstreams can be extracted at all layers to form partial bitstreams in fine-granular levels with fast extraction process.

For a fully scalable SVC video stream, we could adaptively extract appropriate amounts of bitstream to meet a specific target bitrate. Here the extraction operation should be fast enough to support the on-line extraction for the change in the network bandwidth. However, the layer dependency causes a problem if we switch a layer to other layers to extract the partial bitstreams at the GOP boundaries in SVC . After inspecting the SVC bitstream structure, we found out that changing the spatial and quality layers would violate the dependency compatibility. Therefore, we can only make changes in the temporal scalability layers in on-line extraction process within the spatial and quality layers.

Based on the inspection above, we design and implement an SVC streaming architecture, which takes as input the network parameters of measuring the available bandwidth at each transmission time and yields the extracted bitstreams in temporal scalability layer in every GOP unit. Then the extracted SVC bitstreams will be transmitted to the target terminals in RTP protocol and the target terminals receive, decode and render the SVC streams.

This paper is organized as follows: in Section 2, we introduce the basic structure of SVC video streams; Section 3 analyzes the switching mechanism of SVC layers for on-line extraction; Section 4 describe the architecture of our proposed SVC streaming system in details; Section 5 present the implementation and experimental results; Finally, we conclude our work in Section 6.

## 2. SVC Bitstream Structure

SVC bitstreams consist of one or more Network Abstract Layer (NAL) units [2]. Each NAL unit consists of its header and payload. The NAL header is an one-byte data containing the dependency flag and the NAL unit type. The NAL payload is the data contained by NAL units and the payload formats vary among

different NAL types. All NAL units are separated by a specific four-byte delimiter "0x00 00 00 01". For an SVC bitstream, there exist usually two kinds of NAL units: configuration NAL unit and compressed video data NAL. The configuration NAL units consist of Sequence Enhancement Information (SEI) NAL, Sequence Parameter Set (SPS) NAL, Picture Parameter Set (PPS) NAL etc. The SEI is a flexible NAL type with many types, each of which has its own defined data and information. We can even define customized SEI unit by using UUID to identify the user defined data. The SPS contains the sequence configuration parameters for each spatial layer. The PPS contains the picture and slice configuration parameters for decoding.

The compressed data NAL units usually consist of base-layer NAL units and enhancement-layer NAL units. In general, one frame in video stream can be encoded into one AVC base-layer NAL unit followed by several SVC scalable extension NAL units. The base layer NAL units contain the encoded data of minimum spatial layer and worst quality layer. On the other hand, the extension NAL units usually contain residual data of different spatial and quality layers. All extension NAL units will depend on their base layer NAL units and other extension NAL units with lower scalable layers. Figure 1 shows an example of the NAL structure. Here are two spatial layers Ls = 0, 1; and two quality layers Lq = 0, 1 for each spatial layer.

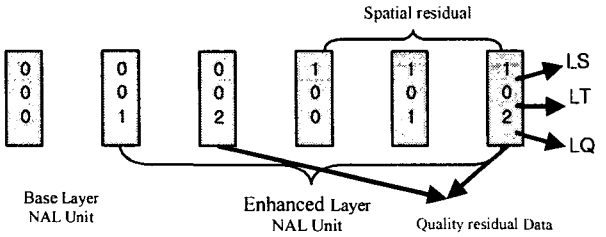


Figure 1 The NAL units structure in one frame

Temporal layer information is not contained in the extension NAL units as the spatial and quality layers; it only relates to frame sequence in each GOP. All the frames are in the same sequence as they are encoded; the sequence is based on the dependency levels. Temporal layer somehow can be seen as the same thing as the dependency level. Figure 2 is an example of temporal layers in SVC stream, the GOP size is 8, so the initial I frame is encoded first, then the P frame of the next GOP is inserted.

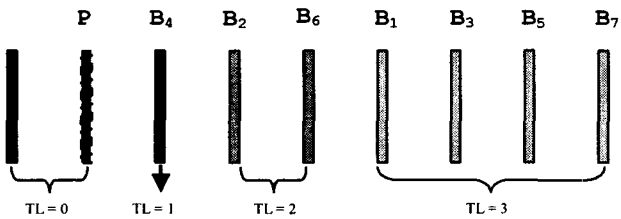


Figure 2 An example of scalability layer change

They have the temporal scalability layer 0. Then other B frames in this GOP are encoded in sequence of dependency level. The Frame B4 depends on I and P frames, so it is encoded first and belong to temporal scalability layer 1. B2 and B6 depend on I, P and B4, which constitute temporal scalability layer 2. B1, B3, B5, B7 are in the last dependency level which constitutes the temporal scalability layer 3.

## 3. Online extraction of SVC bitstreams

As aforementioned, our goal is to extract SVC video stream online. This may require to switch scalable layers for bitstream extraction at GOP boundaries. Switching scalability layers is not always possible due to non-flexible SVC bitstream structure and dependency problem. For spatial, quality and temporal layers, their NAL structures are different. Therefore, we will separately consider the problem of switching spatial and quality scalability layer, and of changing temporal scalability layer.

### 3.1 Spatial and Quality layers

As shown before, the spatial and quality layers are represented in SVC extension NAL units. The extraction process on spatial and quality layers just removes the redundant SVC extension NAL units for all frames. The extraction process is simple but has a problem about the dependency. To ensure that all frames can be decoded correctly, the layer dependency property of the extracted bitstreams should be syntactically correct after extracted. So the extraction must be done at the GOP boundaries. The distance between two consecutive I frames is Intra Period in SVC standard and we call all frames between two consecutive I frames to be I-to-I GOP ($GOP_{I2I}$). If we want to change the spatial scalability layer and quality scalability layer at $GOP_{I2I}$ boundaries, the current SVC specification cannot support such a layer switching as shown in Figure 3.

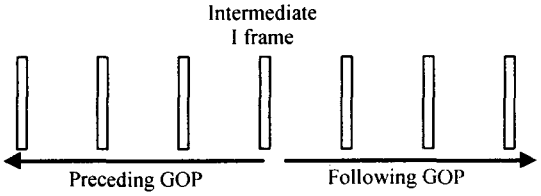


Figure 3 Example of change spatial layer

The I frame at the $GOP_{I2I}$ boundary is referred by the frames in both preceding and following GOP's. So the intermediate I frame must keep residual data of two spatial layers to meet the dependency requirements. However, for SVC decoder implementation, it cannot recognize the extra SVC extension NAL units in the intermediate I frame. Therefore, if we change the spatial or quality layer at $GOP_{I2I}$ boundaries, the SVC decoder won't perform correctly.

Online switching of the spatial and quality scalability layers is

not possible in the current situation unless the video bitstreams are preencoded with the frames in IDR mode at every GOP boundary. Therefore we need to fix this limitation otherwise the SVC decoder will yield errors. One possible solution to the limitation of switching the spatial and quality scalability layer is that the target terminal provides to the SVC extractor the information about its display resolutions, available computational resource, minimum and maximum network bandwidth, required quality level, etc. The streaming server can decide the target spatial and quality layers based on the provided information. Then the streaming server only changes the temporal layer to accommodate with network conditions.

### 3.2 Temporal layer

For changing temporal scalability layer, it is not the same as changing spatial and quality scalability layer. The temporal scalability layer only relates to the frame sequence so we do not worry about intra-frame NAL structure and dependency problem. As we can see in Figure 2, if we want to get temporal layer i, we just remove all frames with the temporal scalability layer greater than I. Then the extraction process is finished. Because the remaining frames do not depend on the frames in the higher temporal scalability layers, there is no dependency problem. Thus, we can do the extraction in every GOP unit. For each GOP, we get the network bandwidth data, calculate the bitrates of all temporal scalability layers with the spatial and quality layers already determined, and then choose one temporal layer that meets the bandwidth constraint.

Network monitor is responsible to detect the network condition and provide the information about the available network bandwidth. The network bandwidth information will be input to the Extraction Decision Engine which calculates the best combination of the scalable layers for the current GOP unit. The SVC Extractor is the main part of the server side and it extracts partial SVC bitstreams according to the best combination of scalable layers determined by the Extraction Decision Engine. For each extracted GOP unit, RTP packetizer can packetize all the extracted NAL units into RTP packets and send them out to over the network.

The streaming client part consists of the RTP unpacketizer [1], the SVC decoder [2] and the YUV player [1]. the RTP unpacketizer will unpack the received RTP packets to NAL units. All NAL units are put into the SVC decoder for decoding. The decoder's output is video frames in YUV format. An YUV player will play the decoded video stream.

If a client wants to play an SVC video stored at server side. First, the clients will send a packet containing client configuration metadata with the necessary information such as player's compatible resolution, frame rate, available quality level, network connection characteristic etc. With the information, the server can decide the spatial layer and quality layer during the transmission session. Then the server can decide the temporal layer by monitoring the currently available network bandwidth. After all three parameters for scalable layer are fixed, the server can extract one GOP unit to form required video clip. Over the network, the extracted SVC bitstreams will be transmitted to the client for playing.

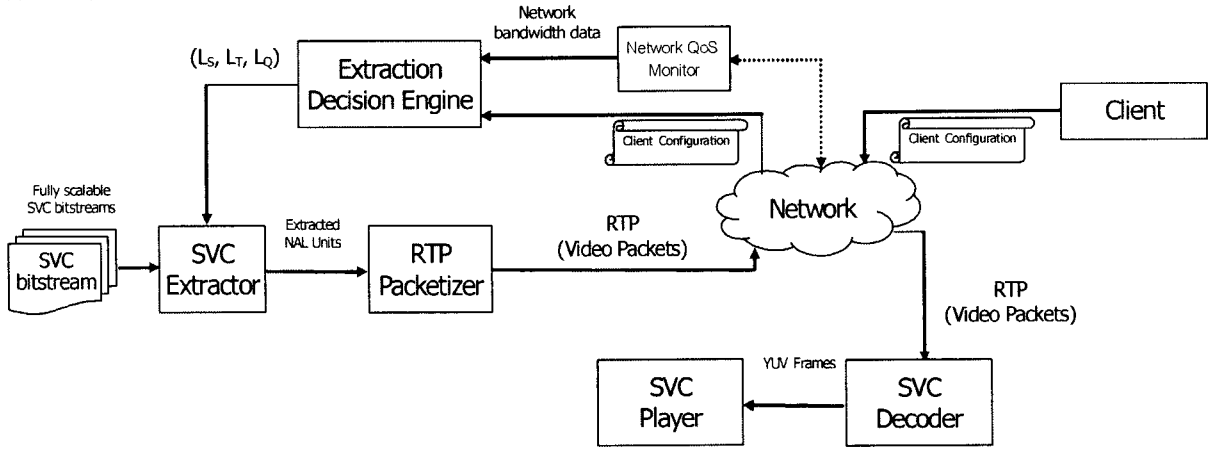


Figure-4: Architecture of the proposed SVC streaming testbed

## 4. Architecture of streaming

With all the above considerations for changing in spatial, temporal and quality layers, we design an architecture for SVC streaming bitstreams as shown in Figure 4. The streaming server consists of an SVC extractor [2], an RTSP packetizer [1], an Extraction Decision Engine [3] and a Network monitor. The

## 5. Implementation and Experiment

The implementation has two main parts: network transmission part and SVC part. The network transmission part mainly refers the MPEG-21 testbed [1] project from NCTU. It implements a testbed to transmit video streams by using RTSP protocol. It originally supports ASP and FGS video types. We add the SVC

modules [2] to the testbed so it can support SVC streaming. The SVC part contains the SVC decoder and extractor; it mainly comes from the SVC referenced software JSVM 5.0 [2]. We integrate the SVC extractor from JSVM 5.0 to the MPEG-21 testbed server part and integrate the SVC decoder to the client part.

For RTSP protocol, there are four sessions: SETUP, DESCRIBE, PLAY and TEARDOWN. In DESCRIBE part, the client will send the configuration metadata to server. Server will configure the extractor decision engine to find best spatial and quality layer. Here the decision engine must analyze the video file to get bitrate information for all scalable layers to make the decision on spatial and quality layers.

Then for each GOP, the extraction decision engine will analyze the bitrates of all temporal layers in the current GOP unit. Suppose the bitrate for temporal layer k is BTk. The network monitor will get the current network bandwidth, and calculate the bandwidth by

$$Bn = Bn\text{-}1 * p + Bm*(1-p)$$

Here Bn is the calculated bandwidth for the current GOP, Bn-1 is calculated bandwidth of previous GOP, Bm is the current monitored network bandwidth and p is the coefficient to control the change rate of bandwidth. With the calculated Bn, we compare it with all BTk, to find the largest k matching BTk < Bn. Then the k value is the chosen temporal layer for the current GOP. So the GOP unit becomes ready for extraction.

At client side, the output of the SVC decoder does not contain any information about the temporal layer or frame rate. Therefore, the SVC player cannot play the decoded video as the extracted frame rate. We need to implement a mechanism to inform the player about frame rate information. Here we can use a special SEI type: unregistered user define data SEI. In this SEI, we can define our own data format identified by a GUID. At extractor part, we can append these SEI NAL units to every GOP units; at decoder part, the SVC decoder will ignore these SEI NAL units. We can add a special parser before the decoder picks up these SEI NAL units and parse the frame rate information, and then pass the information to player. With the frame rate information, the SVC player can play at the right frame rate.

The experiment is done in the LAN network. We use a emulate file to configure the network bandwidth changes. The following figures are snapshots of the testbed running: Figure 5 is the server finish extracting; Figure 6 is the Client GUI; and Figure 7 is the client running.

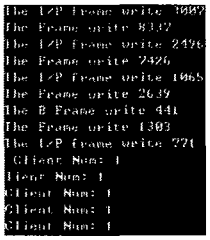


Figure-5: Server running

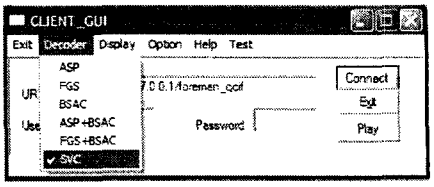


Figure-6: Client GUI

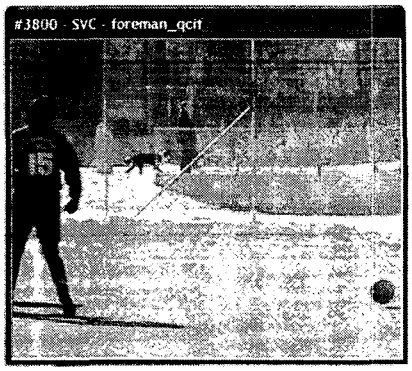


Figure-7: Client Running

## 6. Conclusion

The architecture presented in this paper works well on the SVC video bitstreams. When the emulated network bandwidth are input to the server side, the required amount of SVC bitstream can be extracted appropriately to correspond the changing bandwidth. In the client part, the streaming playback also reflects the temporal layer changes well.

The future work would be to support for changing the spatial and quality layers online. A new NAL type or SEI type should be defined in the standard to carry the changed spatial and quality layers information, and decoder should be modified to be able to parse the information contained in new type. Then the changed spatial and quality layers can be recognized and decoded correctly.

## References:

[1] Chung-Neng Wang et. al,, "Scalable Multimedia Streaming Test Bed for Media Coding and Testing in Streaming Environments," ISO/IEC JTC1/SC29/WG11 MPEG2004/M11117, Redmond, July 2004

[2] ISO/IEC JTC 1/SC 29/WG 11, Joint Scalable Video Model JSVM-5, N7796, Bangkok, Thailand, Jan. 2006.

[3] ISO/IEC JTC 1/SC 29/WG 11, JSVM 5 software, N7797, Bangkok, Thailand, Jan. 2006.

[4] Jeongyeon Lim et. al., "An optimization-theoretic approach to optimal extraction of SVC bitstreams," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 JVT-U081 Hangzhou, China, Oct. 2006