# A Framework for QoE Measurements of Real-time Scalable Video Coding Streaming Using Conventional Servers and Clients[1]

João Almeida, Rui Neto Marinheiro, João Silva and José Moura,

[2] Instituto Universitário de Lisboa (ISCTE-IUL), Instituto de Telecomunicações, Portugal
{jmdaa, rui.marinheiro, joao.silva, jose.moura}@iscte.pt

**Abstract.** SVC streaming studies are mostly constrained to simulations, but prototypes also exist. However, for the latter, modification to streaming servers are required to implement extractors, with obvious modifications in the client side to implement aggregators.
This paper contributes with a new proposal for a practical and simple SVC streaming solution that can be used with of-the-shelf AVC streaming servers and SVC clients, without resorting to aggregators and extractors. The focus is on modifications to the hint tool, creating a standard MP4 container for SVC.
A complete evaluation framework is also proposed, with hardware on the loop, for performing objective and subjective QoE tests, with the support of SVC.
With this framework, several scenarios have been tested, by emulating network conditions. User subjective MOS results have been obtained and then compared with objective PSNR results converted to MOS. These tests demonstrate the framework validity, opening perspectives for future work.

**Keywords:** QoE, SVC, MP4, hinting

## 1 Introduction

At present, the use of video in the Internet is increasing significantly and it is already one of the most popular applications, supplanting peer-to-peer traffic that used to be the most demanding for ISPs (Internet Service Providers).

Streaming is a common approach for video delivery, and in this context video contents should be adapted to meet various constraints of heterogeneous environments; variety of access networks; variety of devices with different capabilities ranging from cell phones with small screens to high-end high-definition displays; etc [1]. A limitation of the prior single layer video coding is that it is not adaptable enough for this video streaming context. Once a source video stream has been encoded with H.264/AVC [2], it will remain the same throughout the

[1] This work was supported by the FCT projects:
**PTDC/EEA-TEL/120666/2010** and **PEst-OE/EEI/LA0008/2011**

communication process. Also the server needs to store a copy of the same video for different resolutions, qualities and timings, to meet the numerous user equipments.

A more eligible solution is to use SVC. Standardized by the Joint Video Team of the ITU-T VCEG and the ISO/IEC MPEG, as a scalable extension of H.264/AVC with the aim of enabling the creation of a bit stream that can be rapidly and easily adapted to fit with the bit-rates of various transmission channels and with the display capabilities and computational resource constraints of various receivers [3].

A number of researchers have studied the issues related to SVC video streaming. In particular [4], Wanger et al. presented an overview on the key aspects of SVC and related transport issues. SVC files themselves are not streamable by a streaming server. The SVC must be stored in a proper file format to facilitate packet-based parsing. For this purpose, the MP4 file (ISO/IEC 14496-14:2003) [5] works as a multimedia container, and it is standardize as part of MPEG-4 Part 14.

Through the hinting process a separate hint track is built to include streaming information in the MP4 file. In [6], authors provided an extensive introduction to the structure of tracks inside the MP4 file. Not many works have been presented concerning the hinting issues for SVC streaming with MP4 containers [4][7][14]. Most works employ complex methods (dependent on extractor and/or aggregators) that make the hinting process cumbersome and expensive, both for human and computing resources [7]. Furthermore, these schemes in order to function require several changes on servers and clients in order to deal with extraction and aggregation operations of the Network Abstraction Layer Units (NALU) [3].

To overcome these limitations, we propose an innovative and working approach that does not need to resort to aggregators and extractors to make the SVC streaming. Our method successfully creates standard-compliant MP4 files containing SVC media from given YUV video files, by implementing modifications in the hint tool. Chen et al [14] follows a similar approach, however they failed to demonstrate it in a real test bed. Conversely, we have implemented a complete end-to-end solution that demonstrates that our method is practical, easy and intuitive for SVC streaming, and it can use any legacy AVC streaming servers and SVC clients.

In order to demonstrate our solution, and support future work, we also propose an evaluation framework for Quality of Experience (QoE) measurements. For the evaluation of video transmitted over a real network, the EvalVid framework is widely used [8] which allows evaluation of H.264 video using objective metrics (such as PSNR). However QoE is a subjective measurement, where Mean Opinion Score (MOS) is mostly used, which scales the human quality impression on the video from bad (0) to excellent (5). Our framework allows the evaluation of video according to these two perspectives. To do this, we have modified and expand the EvalVid framework. We have implemented PSRN evaluations of H.264/SVC streaming, with PSNR obtained with true decoded YUV without filtering NALs dependent on error frames. For this the openSVC decoder [9] was used and modified in our work. We also consider QoE measurements of real-time SVC streaming with a legacy Streaming Server and SVC Client.

Using this framework, we have conducted several tests with real-time streaming scenarios, using off-the-shelf hardware and software, and emulating network conditions in a wireless mobile networks environment. QoE subjective metric using MOS have been obtained, with real test subjects and conditions. Objective PSNR have also been gathered. In order to compare both results, the objective PSNR has been converted to MOS, using Evalvid conversion function.

Section II introduces concepts and clarifies the background. Sections III and IV present the overall scheme architecture and summarize the methods of creating and hinting the resulting MP4 file into a streamable version. Section V illustrates the implementation platform used for the tests. In section VI, subjective MOS and objective PSRN tests are respectively described. Section VII provides experimental results before final conclusions.

## 2 Concepts and Background

Below, we describe some concepts that affect the quality of the SVC streams. We also depict the state of the art in the real time SVC streaming and video evaluation.

### 2.1 Video Quality Evaluation

We can distinguish between Quality of Service (QoS) and Quality of Experience (QoE). Traditional QoS focuses on network performance and data transmission. QoE meant to describe quality from the perspective of the user or consumer, with a focus on perceived quality of the content (user experience), ITU-T recommendation J.247.

A popular objective video quality metric is peak signal-to-noise ratio (PSNR). Like other objective metrics it does not perfectly correlate to perceived visual quality.

For subjective metrics, the mean opinion score (MOS) provides a numerical indication of the perceived quality from the users' perspective of received media after compression and/or transmission. MOS tests audiovisual quality in multimedia services are specified by ITU-T recommendation P.910 [10] and ITU-R BT.500 [11].

The MOS is generated by averaging the results of a set of standard, subjective tests where a number of viewers rate the video quality of test sentences over the communications medium being tested. When there is a reference sequence included in the tests, the output is known as Degradation Mean Opinion Score DMOS. [12]

We identify the main parameters that have an impact on the perceived quality of the video streams. Other parameters, such as frame rate, are assumed constant; parameters such as delay and jitter, leading to delayed packets, are converted into losses. These parameters are independent on the scalability type. Even though we considered spatial scalability in our implementation, these parameters are still relevant for other types of scalability solutions (temporal, quality, or a combination of both).

Network parameters as well as scenarios are both described in recommendation ITU-T G.1050. This recommendation also defines the network architecture and the means of simulating network parameters.

IDR Frequency: The frequency of IDR pictures is an essential factor for the final quality. The increase in the number of IDR frames that in turn decreases the number of P-frames (Predicted frame, contain only the data that have changed from the preceding I-frame, short for intraframe is a single frame of digital content that the compressor examines independently of the frames that precede and follow it), are beneficial because an error will propagate only until the next I-frame arrives. Unless the target playback device requires a different value, every I-frame is an IDR frame. Nevertheless, an encoded I-frame is larger in size as compared to a P-frame. Thus, the final size of the video will be increased.

SVC GOP structure: In the experiments conducted, we used the G16B15 structure with inter-layer prediction. There are 15 B frames between 2 I/P frames. This GOP structure has 4 temporal levels. The I and P frames are at temporal level 0 and the B frames are at temporal levels 1, 2, 3 and 4. With inter-layer prediction that can only be used inside an access unit, and thus between base and enhancement layer pictures at the same time instant.

NALU loss rate for each layer: The NALU is the transport unit of video packet. A NALU can only transport information of one layer. The loss of a NALU affects only a single layer. However, it is important to mention that losing a NALU belonging to the base layer has much more impact on the video quality, than the loss of NALU belonging to other enhancing layers. Because it impacts the other layers as all the other layers in SVC use the base layer as reference and any error in this layer propagates to other layers.

### 2.2 Additional Information Required by the Volume Editor

MP4 file format is a multimedia container that is standardized as MPEG-4 Part 14, or formally as ISO/IEC 14496-14:2003. The format is essential for streaming servers to interpret the media content of the file correctly.
The MP4 file is a container where all data are organized in boxes. The important boxes of the MP4 file are the ftyp, which is compulsory, the moov, and the mdat. The ftyp box is placed at the start of the MP4 file, and describes the file type (e.g. AVC or SVC). The mdat box data is where all the bit streams are dumped into [2].

The moov boxe is logically divided into two track boxes. Media track boxes which contains all meta data, including track information and the reference pointer to the data. Hint trak boxes, which enable the packetization of data for streaming (contains instructions for a streaming server how to form transport packets), from its corresponding media track in a MP4 file, Fig 1.

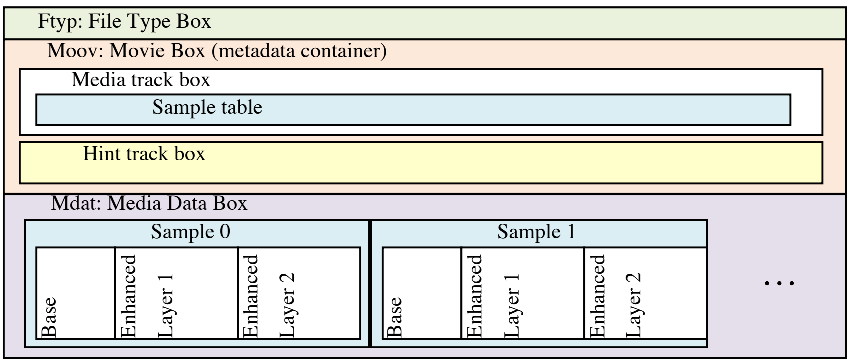


Figure 1. MP4 Container.

The sample table, a structure in a media track box, provides the detailed information about each sample; defines both the physical location of each sample and its timing. I-frames samples (frames) can be decoded without knowing any other samples. Such samples provide useful random access or synchronization points, there indices are stored in the table Sync Sample Box.

### 2.3 Related Work

Concerning the evaluating problem of SVC transmissions, EvalSVC [13] has been developed with large modifications on the Evalvid platform [8] in the coding part, in order to accept SVC bitstreams. Using the same Evalvid platform, with minor modifications, we have managed to obtain SVC evaluations. Thanks to our hinted SVC MP4 container, Evalvid component is able to use the hinted SVC bitstream as if it was a common AVC bitstream.

As mentioned, the hinting issue for SVC in MP4 containers has only been superficially addressed. Also these hinting schemes [4] imply the use of extractors and aggregators, with the already identified disadvantages. Two projects, SVC4QoE [14] and SCALNET [7], managed to get real-time SVC streaming, using these techniques with the above limitations. To address these issues, [15] introduced an MP4 file creator for SVC streaming; however their work was supported only on simulations with trace files, and they did not achieve any real-time streaming. They used the mpeg4ip project [16] with no support for SVC payload, and for this reason they may have encountered some integration problems. The work reported here used the GPAC multimedia framework [17] that already has some support for SVC.

## 3 Scheme Architecture

Our goal is to achieve a simple SVC streaming solution that can be used in several scenarios. The proposed architecture is presented in Fig. 2. This includes a video evaluation framework, partially based in Evalvid, that implements objective and subjective evaluation tests. A description of this scheme follows:

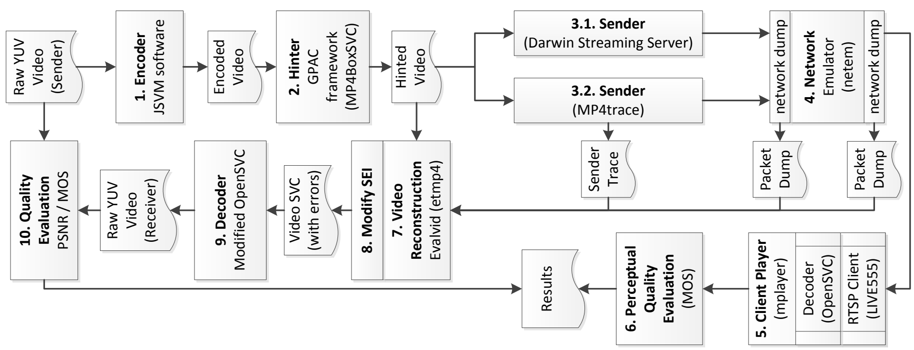


Figure 2. Scheme Architecture

– First (1), a valid SVC video sequence, using the JSVM software encoder [18], is produced and stored in a .264-file, using raw YUV videos.

– Then (2), the modified MP4box (see section IV for more details) insert this SVC video in a MP4 container. At this stage, the MP4 file is hinted by introducing a hint track in order for it to be streamable with legacy streaming servers.

– To obtain subjective measurements, the video is sent using the Darwin streaming server (DSS) (3.1) [19].

– During transmission (4), distortions in the stream are created with real-time network emulations. These are implemented using the network tool NetEM from Linux [20] that operates a variety of functions for different distortion models. Here it is used in a hotspot for emulating network conditions in our scenario. In practice, a network may drop, corrupt, delay, duplicate and reorder packets, which are transmitted through it, according with certain patterns.

– The Client (5) receives the video, using the OpenRTSP tool [21] and decodes it with the openSVC decoder [9]. This video is presented to real subjects to conduct subjective tests (6), and obtain Perceptual quality results (MOS).

– To achieve objective measures, following Evalvid methodologies for evaluation of h264 AVC video, MP4trace component (3.2) will send the hinted file. While transmitting, a trace file to locate frames and macro-blocks in the packetized stream is generated. This contains the logs of the sequence numbers, types, and sizes of the video frames; the number of UDP packets used to transmit each frame (since the frame size may exceed the UDP/RTP maximum payload sizes) and their timestamps.

– Client side and server side packet dumps during transmission (4) are also saved.

– The obtained packet trace, the client and server side packet dumps and the hinted encoded video is processed by EvalVid etmp4 (7) to construct a video sequence that considers packet losses. This video is modified in its SEI information to be compliant with SVC (a change of the SEI address) (8).

– Afterwards, the video is decoded to raw YUV (9) and the objective quality degradation in terms of PSNR is measured (10), and MOS can be estimated using a mapping function. The decoder included in JSVM software suite considers that the NAL unit is in error even if only one byte of data in the NAL unit is corrupted. Also based on the GOP structure, the P and B frames are predicted from either I, P or B frames. So, if a NAL unit corresponding to a frame is in error, JSVM has to eliminate the NAL units corresponding to the frames that are dependent on the frames in error. This process is called filtered bit stream. To avoid these limitations, the Open SVC Decoder developed by IETR was used in this work. The Open SVC Decoder only count frames as lost when the first packet is missing. These characteristics will avoid the filtering. A macro within the library code of the openSVC decoder build was enabled to create YUV raw videos from the decoding process.

The details of the scheme will be discussed in Sections IV to VI. Theoretically, it can handle all kinds of SVC media, regardless of the used codec. The proposed scheme has been implemented by extending and improving existing projects such as Evalvid framework for evaluation, GPAC MP4box for multimedia file construction and openSVC for raw video decoding.

# 4 Creating and Hinting MP4 Files

During the hinting process, our goal is to ensure that no extractor or aggregator is required, in order to simplify the process. We aim that MP4 file can be streamed smoothly by legacy servers that recognizes MP4 format while the delivered data packets are acceptable to any standard SVC client video player, without any modifications, like openSVC or Mainconcept Showcase.

To store video, it is necessary to create a MP4 file container. The MP4Box from GPAC project [17] is available to achieve this. Recent updates state that it can create SVC containers; however there are still some compatibility issues to be solved. To overcome this limitation, a modified MP4Box developed in the openSVC project [22] has been used, where the parser was changed to be fully compliant with the SVC standard. We also added a few changes, like the management of the number of NLAUs, avoiding the truncation of the extra scalability NALUs.

The hinting is then necessary, to creating indication, known as hint tracks from media tracks without affecting the original media content. This provides servers with the needed media information for streaming.

To handle the hinting issue for SVC media tracks, we base our solution in the modified MP4box from the openSVC project, where most hinter functions for H264 have already been declared. The hinter function for SVC was worked out by extending similar functions for H264 AVC.

The functions firstly add a standard hint track container in the destination MP4 file. In this step, the function processes the normal attributes such as session name, network type, media name and port number [23]. The function then analyzes the samples from media tracks and generates hinting information for the hint tracks.

# 5 Implementation Platform

The real-time SVC streaming evaluation framework was deployed according to Fig. 3. This includes an off-the-shelf access router: Asus WL500 Wireless-G Broadband Router. The reason for selecting this device is that its original firmware can be easily replaced by a variety of different third-party opensource firmwares. We have opted for the Open-WRT [24], a small-scale Linux distribution that offers a dedicated SDK for developing applications or libraries. Packages are created and they can be either installed via the package management software or directly integrated in the firmware image.

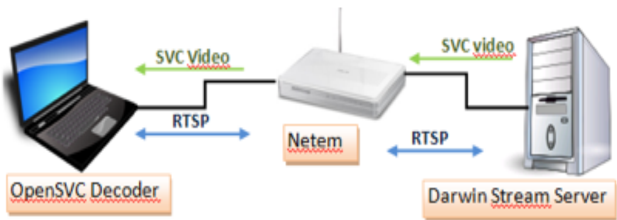


Figure 3. Real-time SVC Streaming Platform.

From a hardware point of view the WL-500Gp is based on the Broadcom SoC (system-on-chip) BCM5352EL. It has a MIPS32 processor running at 200 MHz, an IEEE 802.11b/g MAC/- PHY, an SDRAM controller, and a Fast Ethernet switch with five ports. The router offers in total 16 Mbytes of main memory from which the majority is used by the kernel, necessary daemons (SSH server, HTTP server), and a writable, temporary filesystem (tempfs).

The WL-500Gp is equipped with a 4 Mbytes flash memory, were the open-WRT distribution was installed. In order to include software for emulating network conditions and implement network adaptation, we upgraded the available memory by using an USB 4 Gbytes flash memory pen.

As described before, the emulation of network conditions was implemented by using the Network Emulation (NetEm) module of the Linux Kernel. In order for the NetEm to operate, the module for traffic control (tc) was added to the firmware.

NetEm module is not included in the OpenWRT packages. It was necessary to recompile the OpenWRT firmware source code with the Linux kernel modified to include the emulator, using the firmware modification toolkit

As streaming server, we used a standard Darwin Streaming Server [19] ideal to handle hinted mp4 containers, running in a PC with Linux Ubuntu. As a client, we chose a modified MediaPlayer with openRTSP and the openSVC decoder, running on top of a laptop equipped with a windows operating system.

## 6 SVC Realtime Streaming Tests

H.264/SVC and H.264/AVC target a wider range of video applications ranging from video at mobile devices, with bitrates possibly lower than 30Kbit/s, to HDTV; with bitrates of 20Mbit/s or above. The evaluation of the present framework was targeted for low bitrates, envisaging the mobile market and current Internet bitrates.

H.264/SVC bit streams were encoded from ITU-T YUV test sequences City, Crew, Paris and Soccer provided by Fraunhofer-HHI [25], whose description is present in Table I. The base layer is an AVC bit stream with QCIF resolution (176x144) and 30fps. The spatial enhancement is of a CIF resolution (352x288) and also 30fps. In this way there are 4 test sequences with 10s, each one consisting of 2 layers. These generated SVC streams with 300 frames having a Group of Picture (GOP) size of 16. Since the frequency of IDR impacts directly on the quality of the video, two different intra-periods were evaluated: 64, which give 5 IDRs, and 128, as shown in Table II.

It should be noticed that basic NALU extension types (e.g., types 14, 15, 20) have been reserved for SVC extensions from the AVC NALU types. So only these basic NALU extensions are supported in our extended Evalvid procedure of tests. Other NALU types, such as Payload Content Scalability Information (PACSI), Empty NAL unit and the Non-Interleaved Multi-time Aggregation Packet (NI-MTAP), which are being drafted in [23], are out of our evaluation scope.

In our scenario we have emulated a network with several packet loss rates, as described in Table II. In the tests there is a correlation between discarded packets. For instance, packets that are discarded at a rate of 1%, a quarter of them have their discard probability dependent on previous losses.

TABLE I. LIST OF THE TEST SEQUENCES USED IN THE TEST CASES

| **Sequence description** | ***Resolution Base(QCIF)*** (width x height) | ***Resolution Enhanced(CIF)*** (width x height) | ***Frame rate (f/s)*** |
|---|---|---|---|
| Soccer | 176x144 | 352x288 | 30 |
| | Football, fast motion, camera motion | | |
| Paris | 176x144 | 352x288 | 30 |
| | Test sequence, high contrast, saturated color | | |
| Crew | 176x144 | 352x288 | 30 |
| | NASA crew leaving building, flashlights | | |
| City | 176x144 | 352x288 | 30 |
| | Aerial view, slow motion, camera zooming | | |

TABLE II. QOE AFFECTING PARAMETERS VALUES

| **Parameter** | **Set of values** |
|---|---|
| Packet loss rate (%) | 0, 0.1, 0.3, 0.6, 2, 5, 10, 15 |
| IDR frequency | 64, 128 |

### 6.1 Subjective Tests (MOS)

Subjective tests were based on ITU-R BT-500, which describes the test conditions and the test setup for subjective visual tests. The tests were prepared and conducted at IEFP CFPS, Santarem, Portugal.

All tests used the Single Stimulous Continuous Quality Scale (SSCQS) test method that it is described in ITU-R BT-500-11 [26]. This choice was made because SSCQS is considered a very reliable and widely used method.

Before the actual tests begin, each subject passed a training phase to become familiar with the testing procedure. To adjust the user perception of quality, from a poorly encoded video to a perfect reconstruction, 3 video sequences were generated to reflect the whole range of possible qualities during tests. They were just used to allow each subject to set their personal range from 'bad' to 'perfect'.

Ten subjects were involved in each test, which provides meaningful results. All data was then statistically processed to obtain the Mean Opinion Score (MOS) by averaging the votes of all subjects. In addition the Standard Deviation and the 95% Confidence Interval (CI) were computed.

### 6.2 Objective Tests (PSNR)

Objective tests were based on the popular peak signal-to-noise ratio (PSNR). In spite of the ongoing debate and discussion, it is still widely used. Hence, we can say that PSNR is considered to be a reference benchmark for developing inferring video quality. The validity of PSNR is particularly agreed when used carefully; it is only conclusively valid when it is used to compare results from the same codec and content, which was the case in our test scenario. Given this fact, and considering that our objective tests are partially supported by the Evalvid platform, we have also used its PSNR to MOS conversion function [27].

## 7 Results

Several functional tests were performed to demonstrate real-time SVC streaming. In Fig. 4 a snapshot of the Showcase client is shown. It make evident that the client is receiving a base layer and an enhanced spatial layer, streamed by a standard DSS server.

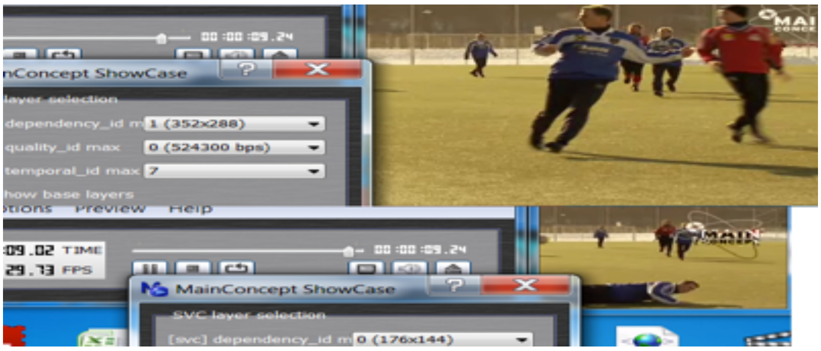


Figure 4. Real-time SVC Streaming, two spatial layers CIF and QCIF.

Performance tests were also conducted to infer differences, in our evaluation framework, between PSNR mapped MOS and subjective MOS, considering different packet dropping levels for two IDRs.

We discovered that the MOS graphs obtained by conversion from PSNR, using our framework, with video traces and reconstruction, don't differ significantly from the subjective MOS observed in a real time streaming video situation, with real subjects evaluating the streaming SVC videos quality, as can be confirmed in Fig. 5 (a, b). Both measured and converted MOS are consistent and decrease with increasing values of packet loss ratio, and better MOS are obtained for 64 IDR frequency, as expected.

A human cannot usually easily detect the loss of a few frames, or off-position comparisons, and this in part justify the underestimation of MOS obtained by PSNR mapping. This is to up to approximately a packet loss ratio of 10%. Above this value, MOS converted from PSNR is too optimistic, but human subjects reach a quality threshold below which regard experienced quality to low.

## 7 Conclusions

In this work a real-time streaming evaluation framework based on the SVC extension of H.264/AVC was experimented.

An easy, intuitive and real-time method of SVC streaming, which can be used with legacy AVC streaming servers and SVC clients, was demonstrated. This method does not need to resort to aggregators or extractors to make the SVC streaming. The focus of this work was on modifications in the hint tool, creating a standard MP4 container for SVC. The resulting MP4 files strictly comply with the ISO/IEC 14496-14 standards and it is readable by standard (AVC) streaming servers without any modification.

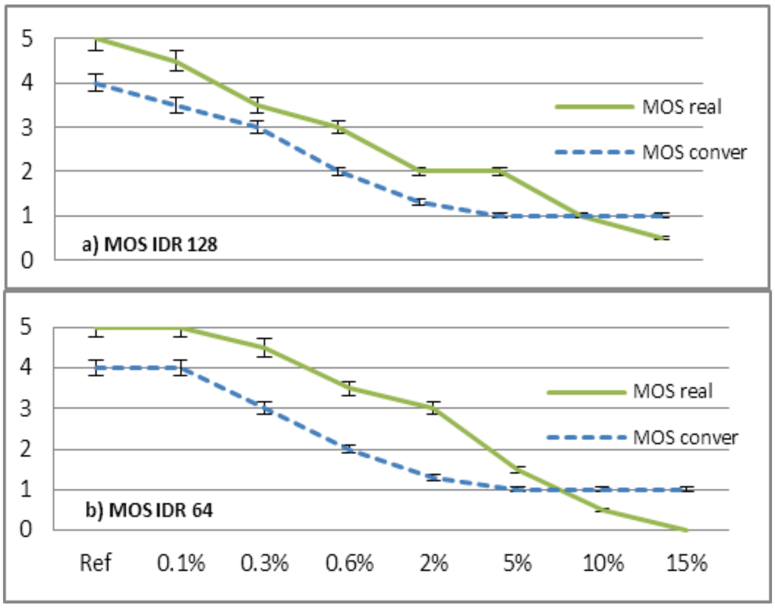


Figure 5. MOS comparison between MOS converted from PSNR and directly measured MOS, for two IDR frequencies. a) 128, b) 64

With this framework, real QoE subjective MOS tests were performed, and results were compared with MOS obtained with objective PSNR tests. Network conditions were emulated using Netem and different use cases for QoE of subjective MOS tests results for SVC streaming in real networks were presented. In the end, the MOS results obtained from PSNR are consistent with the ones obtained with subjective evaluation, proving the validity of our evaluation framework.

For future work we are developing a real-time multi-path SVC streaming, framework using the developed mp4 container system. We are also investigating issues related to MANE (Media Aware Network Element) adaptation, by using a feedback mechanism in our modified Evalvid evaluation framework. Finally, we intend to improve the accuracy of our QoE emulation framework by considering different objective measures and its mapping to MOS.

## References


1. T. Wiegand, J. Ohm, G. Sullivan, and A. Luthra. Special Issue on Scalable Video Coding – Standardization and Beyond. IEEE Transactions on Circuits and Systems for Video Technology, 17(9), September 2007.
2. ISO/IEC 14496-10. Information technology. Coding of audio-visual objects. Part 10: Advanced Video Coding. Dec. 2005.

3. T. Schierl, K. Gruneberg, and T. Wiegand, “Scalable video coding over RTP and MPEG-2 transport stream in broadcast and IPTV channels,” IEEE Wireless Commun. Mag., vol. 16, no. 5, pp. 64–71, Oct. 2009.
4. S. Wenger, Y. K. Wang, T. Schierl, “Transport and Signaling of SVC in IP Networks”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 9, pp. 1164-1173, Sep. 2007.
5. Information Technology - Coding of Audio-Visual Objects - Part 14: MP4 File Format, ISO/IEC 14496-14, 15 Nov. 2003.
6. P. Amon, T. Rathgen, and D. Singer, “File Format for Scalable VideoCoding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 9, pp. 1174-1185, Sep. 2007.
7. M. Ransburg, E. M. Graci´a, T. Sutinen, J. O. Murillo, M. Sablatschan, and H. Hellwagner, “Scalable Video coding impact on networks,” in SVCVision workshop of Mobimedia’ 10, Lisbon, Portugal, Sep. 2010.
8. J. Klaue, B. Rathke, and A. Wolisz, “Evalvid–A framework for video transmission and quality evaluation,” Computer Performance, pp. 255–272, sept 2003.
9. M. Blestel, M. Raulet, “Open SVC Decoder: a Flexible SVC Library”, Proceedings of the international conference on Multimedia, Firenze : Italy, 2010
10. International Telecommunication Union, .Subjective video quality assessment methods for multimedia applications., rec. ITU-T P.910, 1999.
11. International Telecommunication Union, .Methodology for the subjective assessment of the quality of television pictures., rec. ITU-R BT.500-11, 2002.
12. K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, “A subjective study to evaluate video quality assessment algorithms,” in Proc. SPIE - Human Vision and Electronic Imaging, 2010.
13. Tien A. Le, Hang Nguyen, and Hongguang Zhang, “EvalSVC - an evaluation platform for scalable video coding transmis-sion," in 14th International Symposium on Consumer Elec-tronics, Braunschweig, Germany, June 2010.
14. SVC4QoE, http://www.svc4qoe.com/
15. J Chen, D Wu, H Gao,J Y Tham, “A novel scheme of generating streamable MP4 files for scalable video coding”, Internet, 2009. AH-ICI 2009. First Asian Himalayas International Conference on, pp 1-5, Nov. 2009
16. MPEG4IP, http://mpeg4ip.sourceforge.net/
17. GPAC, http://gpac.wp.institut-telecom.fr/
18. “JSVM Software Manual – JSVM 9.12”, ISO/IEC Moving Pictures Experts Group (MPEG) and the ITU-T Video Coding Expert Group (VCET), 26 Mar. 2008.
19. DSS, http://developer.apple.com/darwin/projects/streaming/
20. Netem: http://linux-net.osdl.org/index.php/Netem
21. LIVE555, http://www.live555.com/
22. openSVC, http://sourceforge.net/projects/opensvcdecoder/
23. S. Wenger, Y. K. Wang, T. Schierl, and A. Eleftheriadis, “RTP payload format for SVC video," draft, Internet Engineering Task Force (IETF), September 2009.
24. OpenWrt, http://openwrt.org
25. YUV repository, http://www.tnt.uni-hannover.de
26. ITU-T Study Group 9 Contribution 80, Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality, June 2000.
27. J. R. Ohm, Multimedia communication Technology, Springer, USA, 2004