



HAL
open science

Scalable Video Transmission on Overlay Networks

Tien Anh Le, Hang Nguyen

► **To cite this version:**

Tien Anh Le, Hang Nguyen. Scalable Video Transmission on Overlay Networks. MMEDIA 2010: Second International Conferences on Advances in Multimedia, Jun 2010, Athens, Greece. pp.180 -184, 10.1109/MMEDIA.2010.32 . hal-00625803

HAL Id: hal-00625803

<https://hal.science/hal-00625803>

Submitted on 22 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Scalable Video transmission on overlay networks

Tien Anh Le*, and Hang Nguyen*

**Department of Wireless Networks and Multimedia Services*

Telecom Sud Paris, Evry, France

Email: {Tien_anh.Le, Hang.Nguyen}@it-sudparis.eu

Abstract—Scalable Video Coding is the multi-layer extension of Advanced Video Coding with the advantage of providing visual services for customers with heterogeneous network conditions and terminals' capabilities. In this research, advanced features of Scalable Video Coding are investigated and compared with Advanced Video Coding. A new video transmission evaluation platform is proposed to support the latest Network Abstract Layer Units of Scalable Video Coding. A new interface between the scalable video evaluation platform and an overlay simulation platform is developed so that the transmission performance of Scalable Video Coding bit-streams on an overlay network will be evaluated. Both structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR) measurements are applied to evaluate the performance of the video transmission session. New measurement results are also provided so that other SVC-based service designers can select the right video scalability for their service.

Keywords-Scalable video coding; video evaluation platform; QoS Metrics and Measurement; overlay simulation;

I. INTRODUCTION

People are now working and entertaining in a "3-screen" world. These screens are different in their computational capacities, screen resolutions, and communication bandwidths. However, many service providers are providing their multimedia services based on single layer video coding (such as JPEG2000, Advance Video Coding (AVC)...). A fatal limitation of the single layer video coding is that it is not scalable enough for multimedia services. Once a source video stream has been encoded with AVC, that encoded bit-stream will remain the same throughout the communication process. Encoding parameters of the encoded bit-stream (such as bit-rate, frame-rate, screen size, SNR...) will be determined at the beginning of the communication session by senders and receivers (mostly by receivers). A much better solution is to use Scalable Video Coding (SVC). SVC has been standardized as an extension of the AVC standard since 2007[1]. The main idea of this extension is to apply multi-layer coding into the AVC codec. SVC encodes an input video stream into a multi-layer output bit-stream comprising of a base layer and several enhancement layers. Within those layers, the base-layer is encoded with a basic quality to guarantee that it can be consumed by the weakest receiver of the entire communication group. For the purpose of backward compatibility, the base-layer must be recognized by all conventional H.264 decoders. Enhancement layers,

when received at the receivers together with the base-layer, can enhance the overall-quality of the bit-stream. Especially, when all enhancement layers are received in-order at the receiver together with the base layer, the bit-stream will achieve its original encoded quality. However, when real conditions (such as bandwidths, delays, or displaying screen sizes) do not allow, upper layers can be discarded along the transmission link or at any middle box (relaying entities) for the bit-stream to be fit-in with those conditions without corrupting the video communication session.

Beside scalable video coding, modern multimedia services often rely on Application Layer Multicast (ALM)[2] to serve a large group of subscriber with heterogeneous network and terminal capacities (e.g., IPTV, video conferencing services...). The key concept of ALM is the implementation of multicasting functionality as an application service instead of a network service. ALM algorithms can be optimized for a specific application and of course it can adapt to the SVC-based services.

Video services using SVC have been launched since the standardization of the SVC codec. In order to evaluate those services, designers and researchers are really in-need of a video transmission evaluation tool which is specially tailored for the evaluation of SVC encoded video transmissions over a real or simulated network and which has the interface to an Application Layer Multicast simulation platform. So far, the research community depends on Evalvid[3] for measuring the objective QoS-related parameters of the underlay networks (such as loss-rate, delays, jitters...), as well as evaluating both the subjective (using Mean Opinion Score - MOS) and objective (Peak Signal to Noise Ratio - PSNR) video quality metrics. Evalvid has supported only up to the H.264 video codec. In[4], the connecting interface of Evalvid was extended to replace its simple error simulation model by a more general network simulator (NS-2) so that researchers and designers can plug-in their own network architectures to evaluate. However, none of the above has taken SVC and its metrics into the evaluation. An SVC evaluation platform was first proposed in[5]. In this research, an interface between EvalSVC and Oversim has been developed as an evaluation bridge between Scalable Video Coding services and Application Layer Multicast algorithms.

The rest of the paper is organized as follows. In section II, we will take a look at the new RTP payloads of SVC to

see what should be done to support them in the evaluation platform. Section III will introduce the proposed features of the scalable video coding transmission platform EvalSVC. In section IV, the newly developed interface between EvalSVC and Oversim will be described with some demonstrative simulation scenarios and results. Section V will conclude and open some possible future works.

II. EXTENSIONS IN SCALABLE VIDEO CODING PAYLOADS

Scalable Video Coding was standardized as an extension of H.264/AVC. Deriving from H.264/AVC, it maintains the concepts of using a Video Coding Layer (VCL) and a Network Abstraction Layer (NAL)[1].

A. Scalable video slices

In H.264/AVC, each video frame to be encoded will be partitioned into smaller coding units called macro-blocks[1]. A macro-block will cover a rectangular picture area of luminance samples. Outputs of the VCL are slices: a bit string that contains the macro-block data of an integer number of macro-blocks (making a full frame) which are normally organized into slices according to the frame scanning order; and the slice header (containing the spatial address of the first macro-block in the slice, the initial quantization parameter, and similar information)[6]. In both H.264/AVC and SVC, there are three main types of slices:

- I slice: intra-picture coding using intra-spatial prediction from neighboring regions. This type of slice is self-contained and can be decoded without the reference to any other slice.
- P slice: intra-picture predictive coding and inter-picture predictive coding with one *prediction* signal for each predicted region. This type of slice can only be decoded with reference information from previous I or P frame.
- B slice: inter-picture bi-predictive coding with two prediction signals that are combined with a weighted average to form the region prediction. This type of slice can only be decoded with reference information from the previous and successive I or P frame.

B. Interface to real networks

If the VCL is the interface between the encoder and the actual video frames, the Network Abstraction Layer (NAL) is the interface between that encoder and the actual network protocol, which will be used to transmit the encoded bit-stream. The NAL encoder encapsulates the output slices of the VCL encoder into Network Abstraction Layer Units (NALU), which are suitable for transmission over packet networks or used in packet oriented multiplex environments[7]. In order to generate proper NAL units, we must pre-define the network protocol that we want to use to transmit the video bit-stream. H.264/AVC and SVC support encapsulating VCL slices into a number of network

protocols (H.320, MPEG-2, and RTP...)[8] in which RTP is mostly used because of its popularity.

SVC extended the H.264/AVC standard by providing scalability. There are three main kinds of scalability that SVC can support: Temporal, spatial, quality (SNR), and combined scalability.

III. A SCALABLE VIDEO EVALUATION PLATFORM

Our work manages to develop a video transmission evaluation framework supporting SVC's NALU extension types. The most difficult problem is that those extending types haven't been fully defined and standardized by IETF. However, it should be noticed that, the basic NALU extension types (e.g., types 14, 15, 20) have been spared for SVC extensions from AVC NALU types. So we are going to support only those NALU extensions in our EvalSVC framework since they have already reflected the main concepts of SVC. 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[7], are out of our scope.

A NAL unit comprises of a header and a payload. In AVC, the NALU's header is 1 byte length[9]. Meanwhile, a SVC's NAL header can be 1, 2, or 3 octet length[6]. The first octet of SVC's NAL header is identical with AVC. It contains 3 fields of which 2 first fields (F, NRI) are spared for signaling wire-line/wireless gateway, and the importance of that NALU. The last field in the first octet of the SVC's NAL header is *NALU Type* specifying the NAL unit payload type. NAL unit type 14 is used for prefix NAL unit, NAL unit type 15 is used for subset sequence parameter set, and NAL unit type 20 is used for coded slice in scalable extension. NAL unit types 14 and 20 indicate the presence of three additional octets in the NAL unit header. NALU types 15 contents header information which is not necessary to be repeatedly transmitted for each sequence of of picture[10]. This sub-sequence parameter set can be transmitted on an "out-of-band" transmission for error resilience. We will need this information about the NALU types when we reconstruct the possibly corrupted SVC bit-stream at the receiver side. PRID (priority ID) specifies a priority identifier for the NALU. A lower PRID indicates a higher priority. DID (dependence ID) indicates the inter-layer coding level of a layer representation. QID (quality ID) indicates the quality level of an MGS layer representation. TID (temporal ID) indicates the temporal level of a layer representation.

Fig. 1 illustrates main components of our EvalSVC platform. Some external tools are also integrated into EvalSVC to support the data-flow of the entire framework.

- SVC encoder/decoder: We use JSVM[11] as our main SVC codec.
- Hinter: This component is derived from the mp4box tool of the GPAC library[12]. The main role of this component is to packetize SVC's NALU into RTP

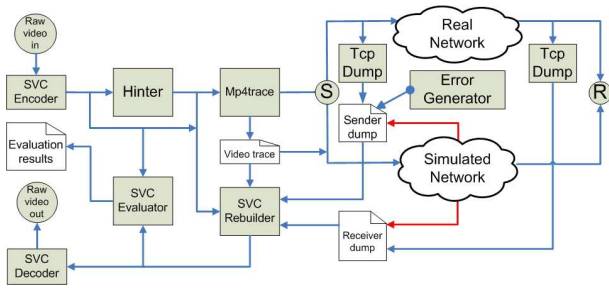


Figure 1. EvalSVC's diagram[5].

packets and add a hint track to the SVC bit-stream. We can consider the hint track as an in-band signaling for the SVC bit-stream. Another option is to distribute the hint track in the format of a SDP file via a separate channel as out-band signaling.

- **Mp4trace:** This component acts as a video sender. Its main part is to send the hinted SVC bit-stream out to the network using the packetization information it has from the Hinder. It also logs the sequence numbers, types, and sizes of the video frames, and the number of UDP packets used to transmit each frame (since the frame size may exceed the UDP/RTP maximum payload sizes), and its sending timescale. Mp4trace can work in streaming mode or camera mode.
- **Networks:** 2 kinds of networks can be used in EvalSVC, real and simulated networks. Real network's conditions can be obtained by using real IP connections over the Internet. Tcpcdump can be used to trace the real network traffic at both ends and to form the sender's and receiver's dumping files. We can also use NS-2 simulated network to form the sender's and receiver's dumping files. Using a NS-2 based simulation network, one can test a new SVC video transmission algorithm, or evaluate the performance of SVC video transmission over a conventional network model (supported by NS-2). A simulated network can comprise of many relaying nodes. Since the SVC bit-stream comprises of multiple layers, enhancement layers can be discarded at the relaying nodes according to the simulation scripts.
- **SVC Re-builder:** Being the heart of EvalSVC, the Re-builder will collect all data from sender's, receiver's dumpings and video trace files, take both the SVC encoded bit-stream and the hinted file at the sender into account and reconstruct a possibly-corrupted output SVC bit-stream at the receiver. The SVC re-builder must understand SVC NALU headers in order to properly rebuild the corrupted SVC bit-stream. When encountering a missing packet, or a missing frame, the SVC re-builder has two options. It can truncate the SVC video frame or fill that frame with zero (or a default value). Other QoS measurements of the network such as

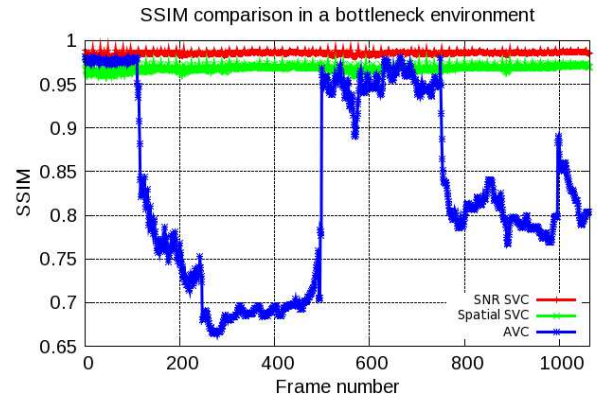


Figure 2. Structural similarity index comparison among SNR, spatial SVC and AVC.

end-to-end delay, jitter, loss rate, sender's and receiver's bit-rate will also be generated.

- **SVC Evaluator:** This component will compare the bit-stream from the output of the SVC Re-builder with the original bit-stream from the sender. Objective and subjective quality evaluation (PSNR, MOS) of the SVC video transmission will be carried out at this component.

We can also use EvalSVC to evaluate the transmission of different kinds of SVC streams on a simulated network using NS-2. We try to find out the best SVC method which can afford the most with the bottleneck condition of the network. To simulate the bottleneck condition, 3 nodes are built using NS-2: node 0 (the sender), node 1 (the relay), and node 2 (the receiver). The first link (link 1), connecting node 0 and node 1, has a bandwidth of 400 kbps, 1 ms delay. The second link (link 2), connecting node 1 and node 2 has a bandwidth of 100 Mbps, 1 ms delay. This network configuration will create a bottleneck on link 1. Firstly, a CIF-size AVC stream is sent from node 0 to node 2 via node 1. In the second and third simulations, a SNR SVC stream and a Spatial SVC stream (both CIF-size) are sent respectively via the same route from node 0 to node 2. We do not use the temporal SVC in our simulation since a temporal SVC stream is identical with an AVC stream. We use EvalSVC to evaluate the Y-PSNR performance of these 3 streams. Fig. 2 shows the performance comparison of the three bit-streams using the structural similarity (SSIM) index. SSIM is a method for measuring the similarity between two images. It was designed to improve on traditional methods like peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proved to be inconsistent with human eye perception. The SSIM measurement on the same bottle-neck simulated conditions gives us the similar result with Y-PSNR when compared SNR, spatial scalability and AVC. However, the SSIM measurement shows that, the AVC bit-stream is not always worse than the SVC bit-streams. Some frames (e.g.,

1-100, 500-750) do have a high similarity with the original bit-stream. We cannot see this similarity if we only use the PSNR measurement.

IV. PROPOSED INTERFACE TO OVERSIM

Oversim[13] is an simulation platform for overlay networks. In comparison to NS-2, it can provide better peer-to-peer and overlay simulation features. We can easily simulate application layer multicast algorithms (such as NICE, Narada...) with an almost unlimited number of peers within a multicasting group. Nowadays, more and more visual services (IPTV, video conferencing...) are being provided on multicast overlay networks. Our evaluation platform has an interface to the Oversim platform so that a scalable video bit-stream generated from our platform can be multicasted from a source node over the overlay simulated network generated by Oversim. Then, at each receiving peers within that multicast group, a possibly corrupted bit-stream will be reconstructed and compared with the original bit-stream. SSIM and PSNR measurements will be carried out at any peer or all peers of that multicast group when necessary. This feature is favorable for visual service designers and researchers of application layer multicast algorithms to verify and evaluate their proposals. EvalSVC makes use of the trace files of actual SVC bit-streams. Instead of sending the real video which has big sizes and often has copyrighted contents, trace files containing frame sizes and sending timescales will be used. We can make use of the online available scalable video coding trace library[14]. We can also generate a trace file from a specific SVC bit-stream by using the *mp4trace* block. According to that trace file, an application running on a randomly chosen source peer of the multicast group will generate the SVC traffic and transmit it through the simulated overlay. At the same time, it creates a sender's dumping file and store them at the sender's side. The video packets are then transmitted on the multicast group to other peers. Each peer will generate a receiver dumping file and write an entry to that file whenever it receives a packet from the sender via the multicast group. After the video transmission session ends, receiver's dumping files are collected from all receiving peers. The information from the sender dumping/trace files, the original/hinted bit-streams and the receivers' dumping files at receiving peers, possibly corrupted bit-streams are reconstructed at each receiver. These files can be decoded using a common Scalable Video Decoder and then compared with the original raw video at the sender using common methods such as Y-PSNR and SSIM.

Fig. 3 shows the Y-PSNR measurement of SVC and AVC video transmission over the OverSim interface. We can see that, regarding the Y-SNR on an ALM environment, SNR-SVC has the best performance followed by combined-SVC, spatial-SVC, and temporal-SVC (it should be noted that the temporal-SVC bit-stream has the smallest number of frames

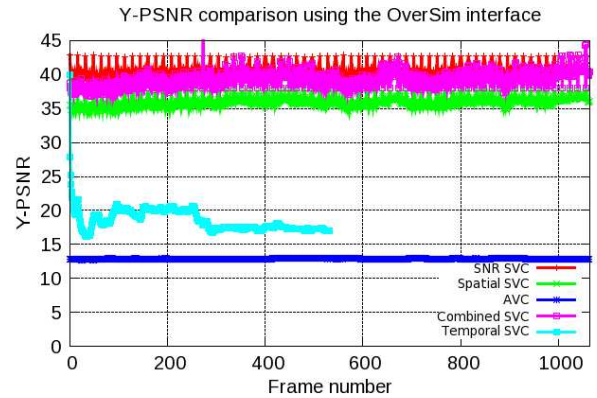


Figure 3. Y-PSNR comparison among SNR, spatial, combined, temporal SVC and AVC transmission performance over the OverSim interface.

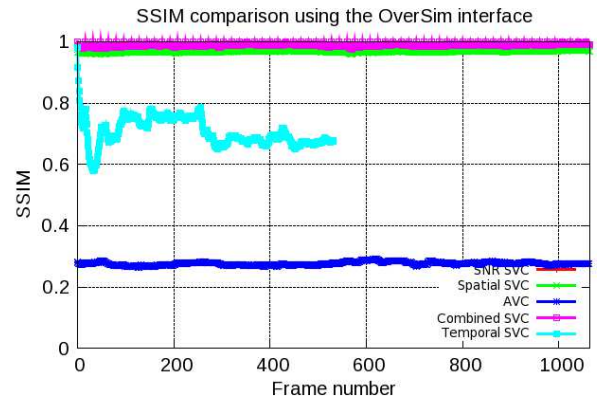


Figure 4. SSIM comparison among SNR, spatial, combined, temporal SVC and AVC transmission performance over the OverSim interface.

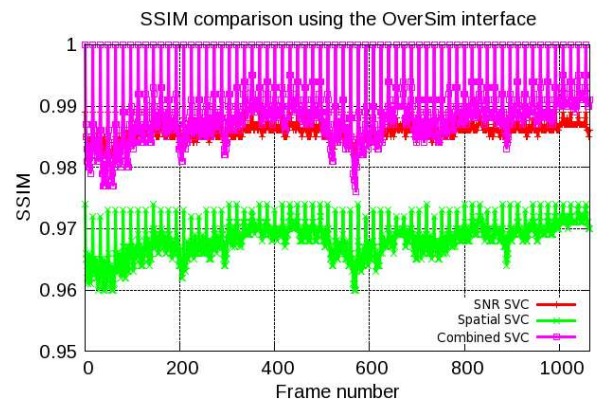


Figure 5. A closer view of the SSIM comparison among SNR, spatial, combined SVC transmission performance over the OverSim interface.

simply because many B frames have been dropped for scalability). AVC still owns the worst performance. Regarding the SSIM measurement among the same set of video over the ALM environment, Fig. 4 show that SNR-SVC, combined-SVC, and spatial-SVC still outperform temporal-SVC and AVC. Fig. 5 displays a closer look to the result shown in Fig. 5. Here we can find an interesting different result than the one we have from the Y-PSNR measurement in Fig. 3, on the same ALM environment, SSIM measurement shows that the combined-SVC has a better performance than the SNR-SVC. It is interesting because it shows that even though a combined-SVC bit-stream may have a lower SNR, the combined scalability can help it to achieve a better similarity to the original bit-stream than the SNR-SVC bit-stream.

V. CONCLUSION AND FUTURE WORK

In this paper, we have introduced EvalSVC, our evaluation platform for Scalable Video Coding video transmission, proposed a new interface between the EvalSVC and Oversim for bridging and evaluating the works between SVC-based services and ALM-based overlay algorithms. New measurement results are also provided in both PSNR and SSIM so that other SVC-based service designers can select the right scalability for their service. The main purpose of this work is to fill the gap between the design, evaluation and implementation processes of variable visual services based on Scalable Video Coding.

REFERENCES

- [1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.
- [2] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A survey of application-layer multicast protocols," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 3, pp. 58–74, 2007.
- [3] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid-a framework for video transmission and quality evaluation," *Lecture notes in computer science*, pp. 255–272, 2003.
- [4] C. H. Ke, C. K. Shieh, W. S. Hwang, and A. Ziviani, "An evaluation framework for more realistic simulations of MPEG video transmission," *Journal of Information Science and Engineering*, vol. 24, no. 2, pp. 425–440, 2008.
- [5] Tien A. Le, Hang Nguyen, and Hongguang Zhang, "EvalSVC - an evaluation platform for scalable video coding transmission," in *14th International Symposium on Consumer Electronics*, Braunschweig, Germany, June 2010.
- [6] S. Wenger, Y. Wang, and M. M. Hannuksela, "RTP payload format for H. 264/SVC scalable video coding," *Journal of Zhejiang University-SCIENCE A*, vol. 7, no. 5, pp. 657–667, 2006.
- [7] S. Wenger, Y. K. Wang, T. Schierl, and A. Eleftheriadis, "RTP payload format for SVC video," *draft, Internet Engineering Task Force (IETF)*, September 2009.
- [8] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.
- [9] S. Wenger, A. G. Teles, and G. Berlin, "H. 264/avc over ip," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.
- [10] Y. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and transport interface of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 149, 2007.
- [11] J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model JSVM-8," *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q. 6, JVT-U*, 2006.
- [12] J. L. Feuvre, C. Concolato, and J. C. Moissinac, "GPAC: open source multimedia framework," in *MULTIMEDIA'07: Proceedings of the 15th international conference on Multimedia*, 2007.
- [13] I. Baumgart, B. Heep, and S. Krause, "OverSim: A flexible overlay network simulation framework," in *Proceedings of 10th IEEE Global Internet Symposium (GI'07) in conjunction with IEEE INFOCOM*. Citeseer, 2007, vol. 7, pp. 79–84.
- [14] M. Reisslein, L. Karam, and P. Seeling, "H. 264/AVC and SVC Video Trace Library: A Quick Reference Guide <http://trace.eas.asu.edu>," 2009.