End-to-end transmission of scalable video contents: performance evaluation over EvalSVC - a new open-source evaluation platform

Tien Anh Le, Hang Nguyen

End-to-End Transmission of Scalable Video Contents: Performance Evaluation over EvalSVC - A New Open-source Evaluation Platform

Tien Anh Le, *Student Member, IEEE,* and Hang Nguyen, *Member, IEEE*

Abstract—Scalable Video Coding (SVC) is a new standard dedicated to the transmission of video contents. Nevertheless, multimedia service research and industry community have not been able to fully utilize the entire potential of this video coding standard because of the lack of a platform for evaluating the end-to-end transmission of SVC-contents. EvalSVC aims to foster SVC-based applications and research in multimedia services. It is capable of evaluating the end-to-end transmission of SVC bit-streams encoded with enhanced features (spatial, temporal, SNR, combined scalability). The output results are both objective and subjective metrics of the video transmission. Interfaces with real networks and an overlay simulation platform are presented. Through these interfaces, the transmission performance of different types of SVC scalability and AVC bit-streams on bottle-neck conditions and overlay network will be evaluated. This evaluation is new because it is conducted on the end-to-end transmission of SVC contents and not on the coding performance. The bottle-neck network environment is set up for evaluation because this realistic and very common network condition usually causes serious problems for video transmission and multimedia services. The overlay network is used for evaluation because it is a future transportation environment for multimedia content delivery. The results show that, both in bottle-neck network conditions and on the overlay network environment, SVC transmission outperforms AVC transmission and SNR scalability has the highest transmission performance. Through these end-to-end transmission performance evaluations over EvalSVC, we can conclude that, indeed SVC achieves its objectives of improving video transmission over realistic network conditions.

Index Terms—Scalable video coding, video evaluation platform, simulation platform, QoS, SVC performance evaluation, evaluation tools, bottle-neck, overlay network, SNR scalability, spatial scalability, temporal scalability

I. INTRODUCTION

H.264/MPEG-4 Advance Video Coding (AVC)[1] is a very famous video compression standard. Its com-

Tien Anh Le and Hang Nguyen are with the Department of Mobile Networks and Multimedia Services, Institute Telecom-Telecom sud Paris, 9 rue Charles Fourier, Evry, 91011 FRA.

pression ratio has enabled many video communication services (such as video conferencing, video surveillance or video-phony...). However, a fatal limitation of this standard is that it is not scalable enough for many 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). However, those senders and receivers may have different screen sizes, different computational capabilities, network conditions (such as bandwidth, delay, jitter...) which might be changed during the communication session. In those cases, in order for the AVC encoded bit-stream to be consumed adaptively at each and every receiver, there must exist middle-boxes in the communication network to convert the incoming AVC encoded bit-streams into various output bit-streams which are suitable for each receiver. This causes a huge delay in the communication session and single points of failure in the communication network. Otherwise the bit-stream will be stuck at bottle-necks and the entire video communication session will be broken. All of these problems make AVC not scalable enough for many video communication services.

People are now working and entertaining in a "3-screen" world. These screens are different in their computational capacities, screen resolutions, and communication bandwidths. A much better solution than AVC is to use Scalable Video Coding (SVC). SVC has been standardized as an extension of the AVC standard since 2007[2]. The main idea of this extension is to apply multi-layer coding into the AVC codec. This is not a totally new idea since people had attempted to implement this idea from prior international video coding standards such as H.262/MPEG.2 Video, H.263 and MPEG.4 Visual[2]. However, the most challenging problem is that, the scalability used to come with a huge increase in the computational complexity. SVC has succeeded in providing scalability at an affordable computational

cost. 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. This base-layer is usually protected while being transmitted over the network by QoS assured transmission methods or Forward Error Correction (FEC) algorithms. 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.

Video services using SVC have been launched since the standardization of the SVC codec. SVC codec is dedicated to improve the transmission performance and not the encoding performance of the video contents. However, few works have been conducted on the end-to-end transmission performance evaluation of SVC contents. In order to evaluate the end-to-end transmission performance of SVC video contents, 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. So far, the research community depends on Evalvid[3] for measuring the evaluation of AVC content transmission. Evalvid can only support up to the H.264 video codec. It cannot take SVC video as a possible input to the evaluation process. Moreover, Evalvid is limited in its interface to only real and Network Simulator (NS-2) based network environment. Evalvid's outputs, limited to only two metrics: Peak Signal to Noise Ratio (PSNR) and Mean Opinion Score (MOS), do not reflect well the performance of the video transmission. In our new EvalSVC platform, we manage to overcome all of these limitations by supporting the evaluation of SVC transmission. In addition to the conventional metrics of PSNR and MOS, we also use Structural Similarity Index (SSIM). SSIM is a method for measuring the similarity between two images. It was designed to improve the traditional methods like Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE), which have proved to be inconsistent with human eye perception.

In our new EvalSVC platform, more interfaces with the bottle neck and overlay networks are developed. Since bottlenecks can be found everywhere on the network and it is actually a serious problem for the multimedia services, the interface to the bottle-neck network is really necessary for SVC transmission evaluation. The interface to the overlay network is required because overlay network is the future transportation architecture of the multimedia services (such as peer-to-peer communication and cloud-based multimedia services).

EvalSVC is an evaluation platform for SVC content transmission. Designed with different interfaces to both real and simulated networks, EvalSVC can be used to evaluate the transmission performance of SVC contents over different network environments. It is a need for the industrial and research communities because it can help the researchers and SVC-based service developers to evaluate the performance of their services and fine-tune them from different quality metrics. This EvalSVC tool is of great interest for the industrial and research communities because it is built and distributed as an open-source platform (http://code.google.com/p/evalsvc/) so that the researchers and developers can freely investigate, use, develop, and enrich the platform based on their requirements and needs. To reach the purpose of evaluating the SVC contents transmission, EvalSVC has the functionalities of a real SVC video streaming server. It can be fully integrated with different network environments using both real video dumping files and popular network simulation platforms such as the Network Simulator (NS-2). We even have interfaces to Oversim, an overlay simulation platform and a hybrid transmission environment between simulated and real networks to support richer evaluation scenarios [4], [5]. The input of the EvalSVC platform can be all different types of SVC-encoded contents (temporal, spatial, SNR scalability) as well as AVC encoded video contents. After a transmission over the network defined by the network environment, the output results can be the objective QoS-related parameters of the under-layer networks (such as loss-rate, delays, jitter...) and three different subjective and objective metrics. The subjective metric is the Mean Opinion Score (MOS). The objective metrics are Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM). Users can judge the performance of the video content delivery based on a single metric, any combination of them or all of the metrics. An evaluation session using EvalSVC starts with the raw video taken from a file or real-time captured by a camera. This raw video will then be encoded by the SVC encoder to form a SVC bit-stream. The SVC encoded bit-stream is packetized into RTP packets accompanied by a hint

track. EvalSVC will stream the hinted video contents from the Sender node to the real/simulated network. A video trace file, a sender and a receiver dumping files will be generated. Using information from all of these files, and the original bit-stream, the SVC Re-builder at the receiver's side will reconstruct the received bit-stream and feed it to the SVC Evaluator for generating the video transmission results. The reconstructed SVC video can also be delivered to the SVC decoder to get the output video play-out at the receiver side.

In comparison with the conventional evaluation platform (such as EvalVid), the main advantage of the newly proposed EvalSVC is its capability to support SVC-encoded contents. Moreover, EvalSVC has enriched interfaces to the overlay simulation platform (built by Oversim). In this paper, we introduce also the EvalSVC's interfaces to real networks such as real Internet or real wireless networks (GPRS, UMTS, WiMAX, LTE, WiFi...), to a hybrid network environment, and to the Overlay simulation platform as the samples of possible interfaces to the platform. In [4], these interfaces to real networks and hybrid networks have been applied for evaluating SVC video transmission over a real mobile WiMAX network and a hybrid network composed of a simulated Internet topology and a real mobile WiMAX network. The real interfaces help EvalSVC to obtain real measurements of the video transmission session on real networks. The Oversim interface helps EvalSVC to simulate an overlay network environment so that distributed SVC-based video content delivery can be evaluated with our EvalSVC. More interfaces can be added to this open source platform upon needs and requirements of the research and industrial communities.

Since our EvalSVC platform is open to the public [6], [7], many people have used our EvalSVC platform for their research and development works. We have been contacted by many EvalSVC's users from both the academy and the industry. For example, we have received interesting questions and requests for support from industrial laboratories and companies such as NEC Labs. America-USA, CTI-Greece, and from universities such as UC Davis-University of California-USA, Kumamoto university-Japan, Lisbon University Institute-Portugal, ITS Surabaya and Universitas Nusa Cendana-Indonesia. We have received quite a high citation by quality external research works such as [8], [9], [10], [11], [12] as well as our college's research works such as [13], [14], [15], [16].

Regretting of not having enough resources to fully support the public's concerns about the EvalSVC platform, the first contribution of this paper is to give a full and detailed descriptions of this tool-set so that people can handle EvalSVC, integrate it into their research work, enrich it upon their needs and requirements to make it a real community's platform for researchers and developers of SVC-based multimedia services.

The second contribution of this paper is to conduct the performance evaluation of SVC contents end-to-end transmission over different network environments. SVC is dedicated to improve the performance of video transmission. Thus, the purpose of the second part of this paper is to verify on realistic network conditions such as bottleneck conditions and overlay networks whether SVC end-to-end transmission really archives its objectives of improving video transmission. The conventional evaluation of SVC and AVC encoding schemes were only to compare AVC's and SVC's source encoding performance. Choosing a different approach from the conventional research works on the evaluation of SVC encoding performance, we manage to provide an evaluation platform for SVC transmission with the availability of EvalSVC. We can now evaluate the entire end-to-end transmission of SVC contents on different types of network conditions. It means that, even though in theory, we can partly understand that SVC should be better than AVC with its scalability feature at the encoding step, but it is only through the performance evaluation results and metrics provided by the EvalSVC platform that we can confirm if the SVC end-to-end transmission is really better than AVC end-to-end transmission in different network conditions.

EvalSVC is applied to evaluate the video services in different scenarios. In the first scenario, we want to compare the end-to-end transmission performance of AVC and different types of SVC scalability contents in a bottle-neck network environment. This evaluation is new because it is a transmission evaluation of different types of SVC scalability and AVC in a bottle neck environment and not their encoding evaluation. A bottle neck can be defined as the difference between the core network's bandwidth (usually with broadband links) and the available bandwidth of the users' terminals (usually limited or narrow bandwidth link). Sometimes, due to the limited computational capacity or a small buffer memory at the terminal, the multimedia applications running on that terminal have to suffer an even smaller bandwidth than the maximum available bandwidth of the users' terminals. This is a very common and serious problem for multimedia services because of the asymmetric bandwidths between the core and the access networks. It is getting worse because users are using different types of terminals (computers, tablets, smart phones) and these terminals increase the asymmetric bandwidths between the core network and the access terminals/networks. A

big multimedia flux coming from a high bandwidth link can not go through narrow bandwidth link. Thus, congestion can happen and the encoded video contents may be blocked on the bottlenecks (the conjunction between the high-bandwidth link and the narrow bandwidth link) before they can arrive to the receivers, creating delay, jitter, packet loss... We select this bottle-neck condition because it is a very common problem of today's network and multimedia services. By evaluating the performance of AVC and different types of SVC scalability content delivery over a bottle neck network, we want to find out if AVC actually has a big problem with congestions in the network and whether SVC can be a solution to this problem. If SVC transmission is better, the purpose is to find which type of SVC scalability would be the best. We choose to evaluate the multimedia services under a bottle neck effect because it is very common in real networks. To evaluate the scalable capability of the SVC contents in a bottle neck network, we use EvalSVC to evaluate the SVC transmission performance against the very popular conventional AVC transmission.

In the second scenario, we evaluate the transmission performance of AVC and different types of SVC scalability contents (temporal, spatial, SNR scalability) in an overlay network environment. This evaluation is original because it is the transmission evaluation of AVC vs. different types of SVC scalability, and not their encoding evaluation. Overlay network is a computer network which is built on the top of another network. For example, distributed systems such as cloud computing, peer-to-peer networks are overlay networks because their nodes run on top of the Internet. Since this type of network is the future transportation architecture of the multimedia services, the purpose for us to select it as a network setting for evaluating the performance of different types of scalability of SVC contents is to find out whether SVC transmission is really better than AVC transmission over the overlay network. And if SVC transmission is better, which type of SVC scalability would be the best suitable for the SVC-based multimedia content delivery via the overlay networks, the future of distributed multimedia delivery.

The rest of the paper is organized as follows. In section II, we will introduce the main ideas of the EvalSVC platform. Available interfaces from the EvalSVC platform to real and simulated networks are described in section III. Performance comparison of AVC vs. different types of SVC scalability end-to-end transmission on EvalSVC in a bottleneck network environment is presented in section IV and performance comparison of AVC vs. different types of SVC scalability end-to-end transmission on EvalSVC over an overlay network environment is introduced in section V. Conclusion is made in section VI.

II. PROPOSED EVALSVC PLATFORM DESCRIPTION

A. Architecture description

Our work manages to develop a video transmission evaluation framework supporting SVC's NALU extension types. The Network Abstraction Layer (NAL) is the interface between the SVC encoder and the real network protocol, which will be used to transmit the encoded bitstream. The NAL encoder encapsulates the output slices from the VCL encoder into Network Abstraction Layer Units (NALU), which are suitable for transmission over packet networks or used in packet oriented multiplex environments[17]. 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 protocol (H.320, MPEG-2, and RTP)[18] in which RTP is mostly used because of its popularity. The most difficult problem is that those extending types haven't been fully defined and standardized by IETF. In order to evaluate the SVC transmissions, it is required that Hinter and SVC Rebuilder blocks in Fig.1 must support SVC new NALs so that the sender and receiver can packetize and rebuild the SVC frames. Therefore, these blocks are essential for SVC evaluation. 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[17], 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[19]. Meanwhile, a SVC's NAL header can be 1, 2, or 3 octet length[20]. The first octet of SVC's NAL header is identical with AVC (Fig.2). 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[21]. 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.

B. Functional blocks

Figure1 illustrates basic components of our EvalSVC platform. Some external tools are also integrated into EvalSVC to support the data-flow of the entire framework.

- Raw video in: This is the input video. Normally the YUV or CIF formats are used as they are acceptable by SVC encoders as well as common video capturing devices.
- SVC encoder/decoder: We use JSVM[22] as our main SVC codec.
- *Hinter*: This component is derived from the mp4box tool of the GPAC library[23]. The main role of this component is to packetize SVC's NALU into RTP 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. The difficulty that the SVC Hinter must support new SVC's NALUs as described in Fig.2. Because these new NALUs are being standardized at the time we were developing the EvalSVC platform, therefore, it is difficult for us to support the right NALUs while still keep the possible extensions for other NALUs which may be standardized later. Nevertheless, the rapid development of SVC-based services requires the instant support of the Hinter on new NALUs. We have developed the Hinter so that it supports the basic NALUs which are required for SVC transmission and reserves some units for possible extensions.
- 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 Hinter. 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 time-scale. 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. Tcpdump 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 dumping 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 end-to-end delay, jitter, loss rate, sender's and receiver's bit-rate will also be calculated.
- Error Generator: Normally, an optimal transmission condition can be obtained by using a direct connection between a sender and a receiver. We can use the Error Generator to modify the dump and trace files according to a pre-defined error distribution function.
- 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.
- Sender/Receiver nodes: Real or simulated nodes on the transmission network. They are the departure and destination of the video transmission.

Fig. 1.   EvalSVC's diagram.

Fig. 2.   SVC NALU's header.

A sample evaluation session using EvalSVC starts with the raw video taken from a file or real-time captured by a camera. This raw video will then be encoded by the SVC encoder to form a SVC bit-stream. The SVC encoded bit-stream is fed into the Hinter to be packetized into RTP packets. A hint track will also be added to the original bit-stream. Mp4trace will send the hinted file (using streaming or camera mode) from the Sender node to the real/simulated network. A video trace file, a sender and a receiver dumping files will be generated. Using information from all of these files, and the original bit-stream, the SVC Re-builder will reconstruct the received bit-stream and feed it to the SVC Evaluator for generating the video transmission results. The reconstructed SVC video can also be delivered to the SVC decoder to get the output video play-out at the receiver side.

C. Evaluation metrics

1) Objective metric 1 - Peak signal-to-noise ratio:
Peak signal-to-noise ratio (PSNR) is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. More specifically, here we use Y-PRNR to compare the PSNR of the luminance (Y) component of the videos. For objective measurement, EvalSVC uses the Peak Signal to Noise Ratio (PSNR) frame by frame. In YUV video, since the human's eyes are more sensible with the luminance component of the video than with color components, EvalSVC calculates PSNR of the luminance component Y of source image S and destination image D.

$$Y - PSNR(s,d) = 20 \log_{10} \left(\frac{V_{peak}}{MSE(s,d)} [dB] \right)$$

$$MSE(s,d) = \sqrt{\frac{1}{N_{col} N_{row}} . A}$$

$$A = \sqrt{\sum_{i=0}^{N_{col}} \sum_{j=0}^{N_{row}} [Y_S(n,i,j) - Y_D(n,i,j)]^2}$$

(1)

In which:

- $V_{peak} = 2^k - 1$
- k=number of bits per pixel (luminance component)

2) Objective metric 2 - Structural SIMilarity Index:
Structural SIMilarity Index (SSIM) is a method for measuring the similarity between two images. It was designed to improve the traditional methods like Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE), which have proved to be inconsistent with human eye perception.

$$SSIM(x,y) = \frac{(2\mu_x \mu_x + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

(2)

In which:

- μ_x, μ_y are the mean values of x and y,
- σ_x^2, σ_y^2 are the variance values of x and y,
- σ_{xy} is the covariance of x and y,
- c_1, c_2 are two variables to stabilize the division with weak denominator;

3) Subjective metric: For subjective measurement, we use Mean Opinion Score (MOS)[24], which scales the human quality impression on the video from bad (0) to excellent (5). After having the PSNR measurements, the corresponding MOS scales can be found by using the conversion table I.

TABLE I
CONVERSION BETWEEN PSNR AND MOS[25].

PSNR [dB]	MOS
> 37	5 (Excellent)
31 - 37	4 (Good)
25 - 31	3 (Fair)
20 - 25	2 (Poor)
< 25	1 (Bad)

III. PROPOSED EVALSVC'S INTERFACES

A. Proposed 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[17]. 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...)[18] 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).

B. Proposed interface to Overlay simulation platform

Oversim[26] is a 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
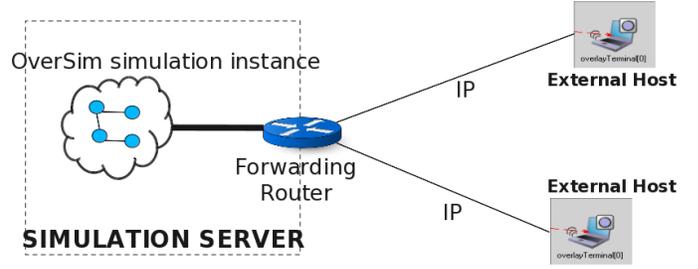

Fig. 3. Hybrid network settings.

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 favourable 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 available on-line scalable video coding trace library[27]. 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 and MOS.

C. Hybrid network environment

Previously, EvalSVC either supports simulation network or real network but not two types of environment at the same time. There are many cases when we need to simulate a heterogeneous network in which several terminals are using different types of access networks to participate into an overlay network. Figure3

demonstrates a hybrid network environment in which the evaluation of the SVC transmission can be done on a hybrid environment of simulated and real networks. While the simulation server applies the INET underlay Internet generated by GT-ITM, the external hosts apply a single-host underlay to connect to the simulation server via real forwarding routers. Using this kind of hybrid network environment, evaluations can be made when only a limited number of real terminal equipments are provided.

IV. PERFORMANCE COMPARISON OF AVC VS. DIFFERENT TYPES OF SVC SCALABILITY END-TO-END TRANSMISSION ON EVALSVC IN A BOTTLENECK NETWORK ENVIRONMENT

Our first evaluation scenario is to compare the end-to-end transmission performance of different types of SVC scalability and AVC performances in bottle-neck environment. We try to find out the best SVC method which has the highest resistance against the bottleneck condition of the network since bottleneck is a very common condition on today's network [28]. The evaluation process and platform's architecture are demonstrated in Fig.4. A sample evaluation session using EvalSVC starts with the raw video taken from a file or real-time captured by a camera. This raw video will then be encoded by the SVC encoder to form a SVC bit-stream. The SVC encoded bit-stream is fed into the Hinter to be packetized into RTP packets. A hint track will also be added to the original bit-stream. Mp4trace will send the hinted file (using streaming or camera mode) from the Sender node to the real/simulated network. A video trace file, a sender and a receiver dumping files will be generated. Using information from all of these files, and the original bit-stream, the SVC Re-builder will reconstruct the received bit-stream and feed it to the SVC Evaluator for generating the video transmission results. The reconstructed SVC video can also be delivered to the SVC decoder to get the output video play-out at the receiver side. For example, a cif-size raw file with 1065 frames is encoded using SNR SVC. The output bit-stream is sent via a real direct IP connection from a sender to a receiver. We manually generate errors by erasing entries at the sender's and receiver's dumping files. At the receiver, the received bit-stream is re-constructed by using the re-builder component of EvalSVC. Since the JSVM decoder cannot decode a corrupted bit-stream, we need to extract the uncorrupted base-layers out of the corrupted bit-stream for it to be decoded by the decoder. We can also use EvalSVC to evaluate the transmission of different kinds of SVC streams on a simulated network using NS-2.

TABLE II
SIMULATION PARAMETERS OF THE SVC TRANSMISSION OVER BOTTLENECK NETWORK.

| Parameters	Values
Purpose	Compare performance of AVC and different types of SVC on bottleneck condition
Encoding	• AVC, • SVC: Temporal, Spatial, SNR
Video size	CIF
Transmission network	Bottleneck condition
Network simulation tool	NS-2
Network condition	• Link 1: 400 kbps bandwidth, 1ms delay, • Link 2: 100 Mbps bandwidth, 1ms delay.
Video quality related measurements	• Objective: Y-PSNR, SSIM, • Subjective: MOS
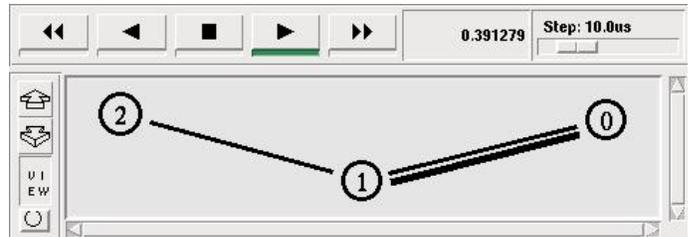
Fig. 5. NS-2 based simulation diagram of video transmission over a bottleneck network.

In the first simulation scenario (Fig.5), we try to find out if SVC transmission is really better than AVC transmission in the bottle neck condition of the network and if it is the case, which type of SVC scalability would be the best. 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 use EvalSVC to evaluate the end-to-end transmission performance of these 3 streams with all three different metrics (PSNR, SSIM, and MOS).
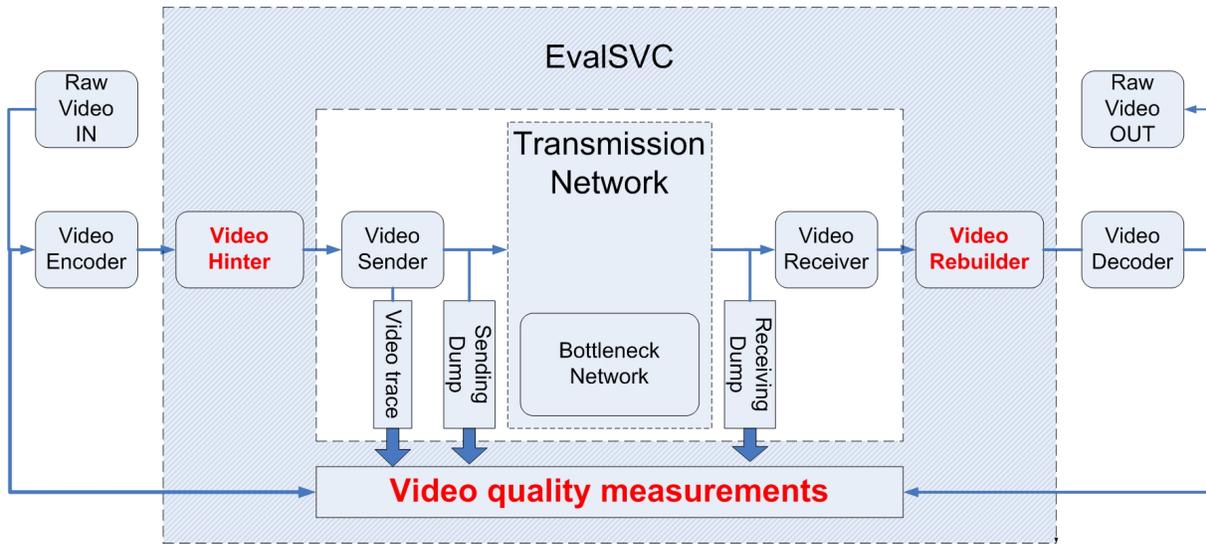
Figure 6 shows that, when bottom-neck occurs, SNR

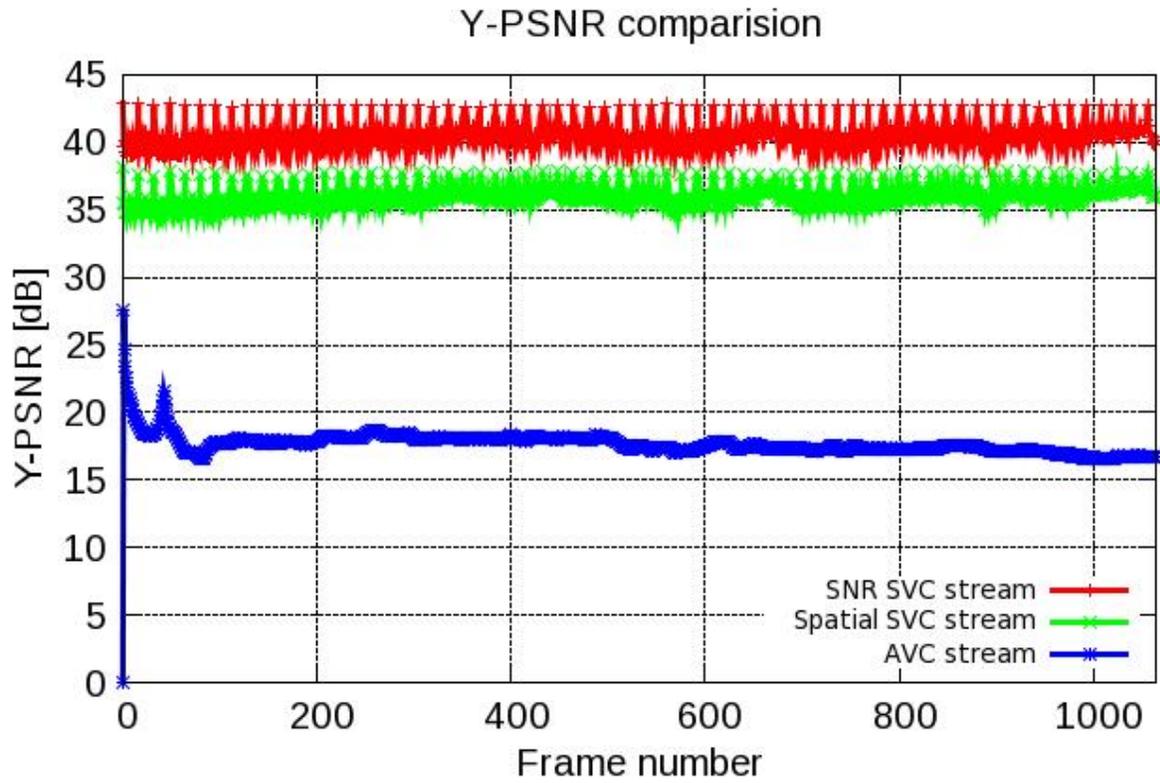Fig. 4. EvalSVC with the evaluation of the SVC transmission on bottleneck network.

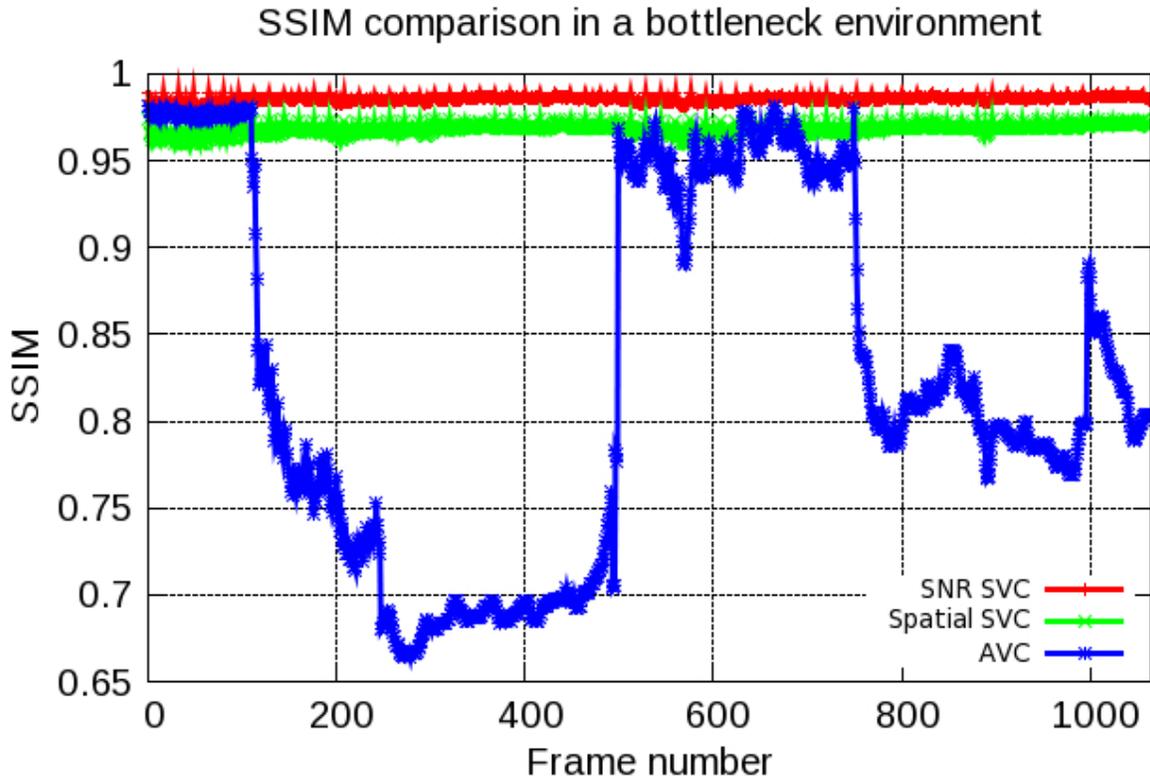Fig. 6. Y-PSNR comparision among AVC, SNR SVC and Spatial SVC streams.

Fig. 7. Structural similarity index comparison among SNR, spatial SVC and AVC.

SVC has the best and AVC has the worst Y-PSNR performance. MOS grades of AVC, spatial SVC, and SNR SVC streams are 1.02, 4.07, and 5, respectively (for subjective measurement, Mean Opinion Score (MOS) is used, which scales the human quality impression on the video from bad (0) to excellent (5)). There is also a convert table between PSNR and MOS values[24]. We can conclude that, AVC is very sensible to bottleneck, a single bottleneck in the transmission route can easily block the entire communication session. Meanwhile, all SVC streams can perform quite well (MOS > 4) under the bottleneck condition of the network, among those, SNR SVC has the best performance. Figure 7 shows the performance comparison of the three bit-streams using the structural similarity (SSIM) index. 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. With a structural similarity, the SSIM metric can better reflect the video transmission quality for human visual system. As it can be analysed from Fig. 7, at the beginning, the quality of the AVC is good. Then, after the first 100 frames, the quality is suddenly dropped down due to transmission problems until the 500th frame. During that period, the video quality is very bad and the received video content is absolutely not similar to the sending

video content and therefore almost not recognisable by the users. The quality recovers by the 500th frame and then continues to be acceptable until frame 750, and then it becomes very bad again until the end. We cannot see this video quality evolution if we only use the an average metric like PSNR in Fig. 6. This big variation in video quality will be perceived very badly by the users. For example, a part of the movie or an important discussion of the video communication session can be missed. From the end user Quality of Experience (QoE) point of view, a change from a high video quality to a very bad video quality brings a very bad user experience. We can see from Fig. 7 with the SSIM metric that AVC transmission creates very bad QoE because of its high variation in video transmission quality. Different types of SVC scalability transmission is better not only because it has a higher PSNR performance but also because it has a lower variation in SSIM.

Figure 8 shows the frame loss flags of different types of video encoding methods. A value of 1 (one) means the corresponding frame is lost while a value of 0 (zero) means the frame is successfully received. According to the frame loss flag of the AVC transmission, we can find that, the AVC video starts to loose its frames from frame number 100. The frames from 200 to about 500 are heavily lost. The density of the frame loss flags are lower
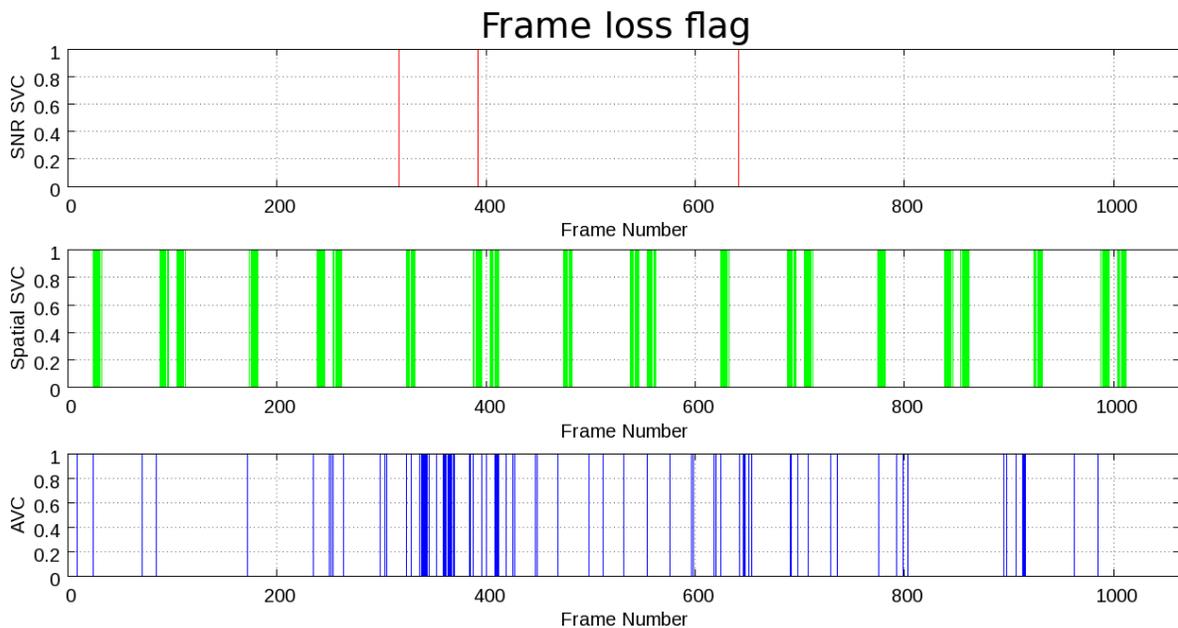
Fig. 8. Frame loss flag among AVC, SNR SVC and Spatial SVC transmission on bottleneck condition.

from frames 600 to about 750. Frame numbers from 800 to 900 suffer a big loss. Since the SSIM value is more sensible with the loss of the B frames, we can recognize its variance corresponding to the density of the AVC's frame loss flag. Meanwhile, the Y-PSNR result in Fig.6 shows that the Y-PSNR is not sensible with the frame loss.

From the results in Fig.6 and Fig.7, we can conclude that SVC has a better resistance than AVC in the bottleneck conditions of the network and SNR SVC has the best end-to-end transmission performance. Since bottleneck condition is very common in overlay network, we recommend using SVC, and more specifically SNR SVC, to build video services on overlay network.

V. PERFORMANCE COMPARISON OF AVC VS. DIFFERENT TYPES OF SVC SCALABILITY END-TO-END TRANSMISSION ON EVALSVC OVER AN OVERLAY NETWORK ENVIRONMENT

We evaluate here the end-to-end transmission performance of AVC vs. different types of SVC scalability contents on the overlay network. We use a popular overlay network (NICE) to transmit the video contents in this evaluation scenario [29]. In order to evaluate the transmission performance of different types of SVC on the overlay environment, the evaluation process and platform architecture are demonstrated in Fig.9. EvalSVC makes use of the trace files of actual SVC bit-streams. Instead of sending the real video which has big size and often has copyrighted contents, trace files containing

frame sizes and sending time-scales will be used. We can make use of the on-line available scalable video coding trace library[27]. 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.

Figure 10 shows the Y-PSNR measurement of SVC and AVC video transmission over the OverSim interface. We can see that, regarding the Y-PSNR on an ALM environment, SNR-SVC has the best performance followed by spatial-SVC, and temporal-SVC (it should be noted that the temporal-SVC bit-stream has the smallest number of frames simply because many B frames have been dropped for scalability). AVC still owns the worst
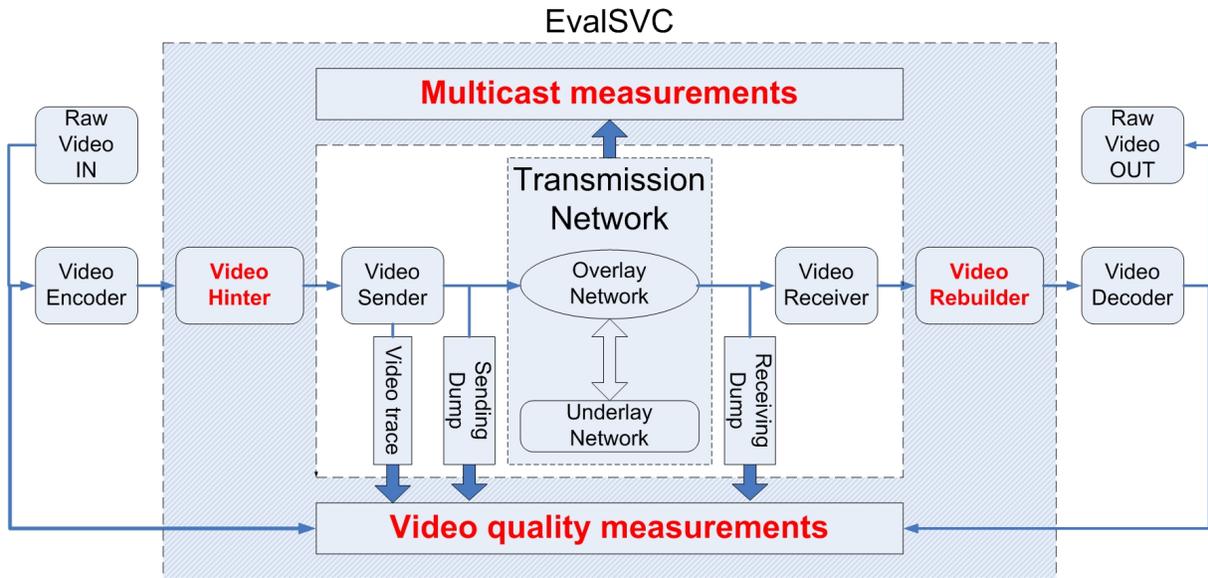
Fig. 9. EvalSVC and the performance evaluation of different types of SVC transmission on overlay network.

TABLE III
SIMULATION PARAMETERS OF THE DIFFERENT TYPES OF SVC
TRANSMISSION ON OVERLAY NETWORK CONSTRUCTED FROM
USING CONVENTIONAL COST FUNCTION.

Parameters	Values
Purpose	Compare video quality related performance of SVC transmissions on overlay network constructed by using a popular cost function
Encoding	SVC: Temporal, SNR, and Spatial
Video size	CIF
Transmission network	Overlay network
Service	Application Layer Multicast of SVC video
Network simulation tool	Oversim
Number of peer	1-1024 peers
Underlay network	Internet topology generated by GT-ITM
Cost functions	NICE's popular cost function
Video quality related measurements	- Y-PSNR - SSIM

performance. Regarding the SSIM measurement among the same set of video over the ALM environment [5], Fig.11 show that SNR-SVC, and spatial-SVC still outperform temporal-SVC and AVC.

The conclusion to withdraw from these results is that SVC transmission performs better than AVC transmission on overlay network and SNR SVC has the best quality when transmitted on overlay network.

VI. CONCLUSION

In this paper, we have introduced EvalSVC, a new evaluation platform for end-to-end transmission of Scalable Video Coding contents. The first 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. We have proposed new interfaces between the EvalSVC platform and the bottleneck and overlay network environments. Output measurement results are also provided in three different metrics: MOS, PSNR and SSIM to better reflect the SVC-contents end-to-end transmission performance.

The first contribution of this research work is to provide all necessary information of the EvalSVC platform's architecture to help the user to handle the platform more easily. We leave the platform as an open-source evaluation tool[7] on line at http://code.google.com/p/evalsvc/ for the industrial and research community. Therefore, it is easy for them to handle the source codes, apply and integrate into their systems, develop and extend the functionalities if necessary.

The second contribution is to conduct end-to-end transmission performance evaluation of SVC contents. Two main evaluation scenarios have been conducted. The first scenario is the end-to-end transmission performance comparison of AVC vs. different types of SVC scalability contents on EvalSVC in a bottleneck network environment. The second scenario is the end-to-end transmission performance comparison of AVC vs. different types of SVC scalability contents on EvalSVC over an overlay network environment. Using our newly developed EvalSVC platform and the end-to-end transmission
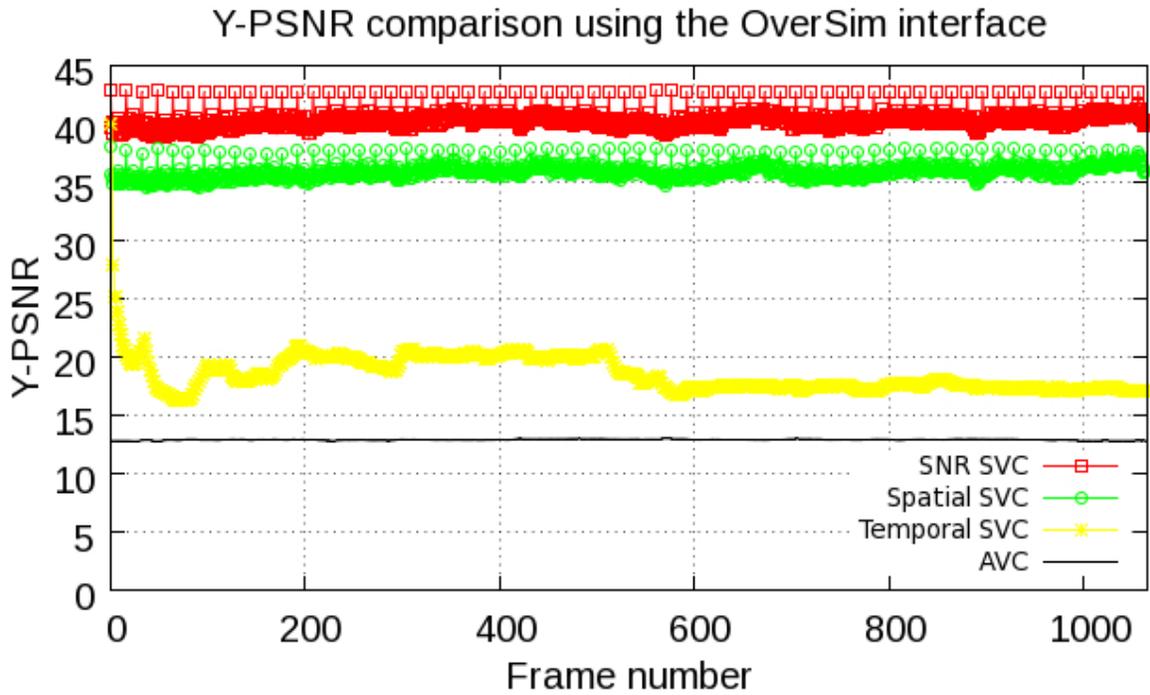
Fig. 10.   Y-PSNR comparison among SNR, spatial, temporal SVC and AVC transmission performance over the OverSim interface.
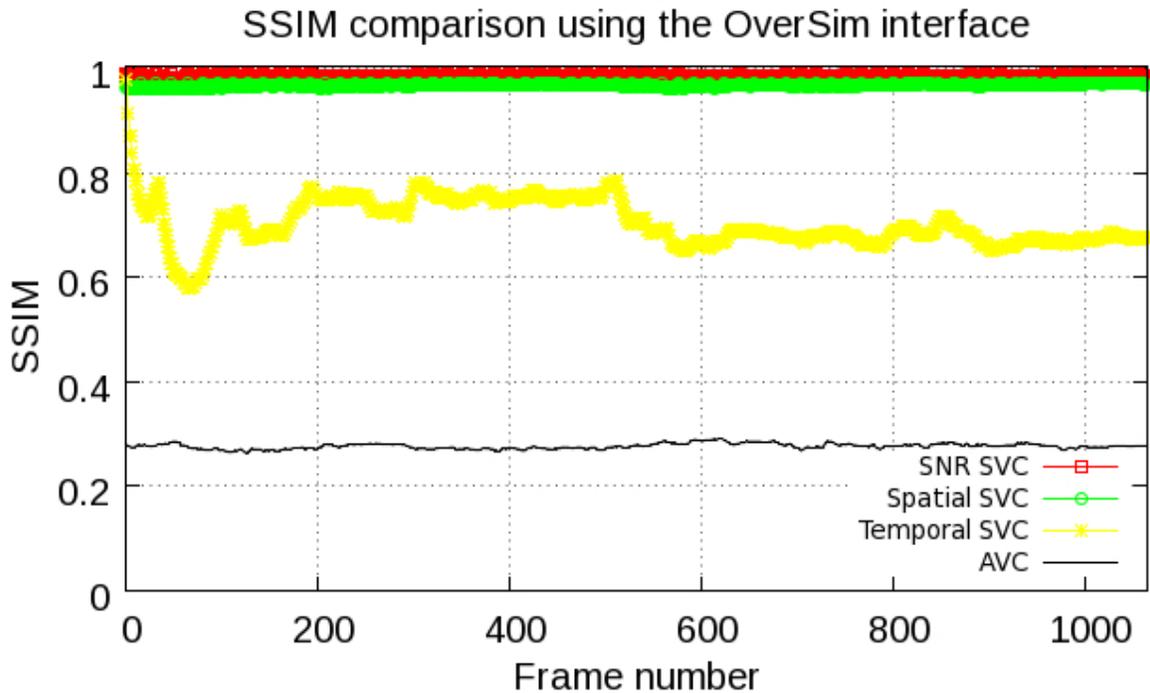

Fig. 11.   SSIM comparison among SNR, spatial, temporal SVC and AVC transmission performance over the OverSim interface.

evaluation scenarios, we found that, SVC transmission outperforms AVC transmission both in bottleneck conditions and on overlay network. Among different types of SVC's scalability, through end-to-end transmission evaluation results, we found that, SNR scalability has the highest PSNR, MOS as well as SSIM performance while transmitted both in bottleneck conditions and on the overlay network. Through these transmission performance evaluations, we can conclude that, indeed SVC achieves its objectives of improving video transmission over realistic network conditions.

For future work, we plan to apply EvalSVC to investigate more SVC transmission scenarios and with different types of multimedia services.

REFERENCES

[1] J. V. Team, "Advanced video coding for generic audiovisual services," *ITU-T Rec. H*, vol. 264, pp. 14496–10.

[2] 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.

[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] Tien A. Le and Hang Nguyen, "Application-aware cost function and its performance evaluation over scalable video conferencing services on heterogeneous networks," in *2012 IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks (IEEE WCNC 2012 Track 3 Mobile & Wireless)*, Paris, France, Apr. 2012.

[5] Tien A. Le, Hang Nguyen, and Hongguang Zhang, "Scalable Video transmission on overlay networks," in *Second International Conferences on Advances in Multimedia*, Athens, Greece, June 2010, pp. 180–184.

[6] Tien A. Le, Hang Nguyen, and Hongguang Zhang, "EvalSVC - An evaluation platform for scalable video coding transmission," in *Consumer Electronics (ISCE), 2010 IEEE 14th International Symposium on*, 2010, pp. 1–6.

[7] Tien A. Le, Quang H. Nguyen, and Anh M. Nguyen, *EvalSVC tool-set: http://code.google.com/p/evalsvc/*, 2009.

[8] C. Bouras, S. Charalambides, K. Stamos, S. Stroumpis, and G. Zaoudis, "Power Management for SVC Video over Wireless Networks," in *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 International Conference on*, Oct. 2011, pp. 270–276.

[9] P. McDonagh, C. Vallati, A. Pande, P. Mohapatra, P. Perry, and E. Mingozzi, "Quality-Oriented Scalable Video Delivery using H. 264 SVC on an LTE Network," .

[10] K. Rantelobo, G. Hendrantoro, A. Affandi, and H. A. Zhao, "A New Scheme for Evaluating Video Transmission over Broadband Wireless Network," *Future Wireless Networks and Information Systems*, pp. 335–341, 2012.

[11] A. Pande, V. Ramamurthi, and P. Mohapatra, "Quality-oriented Video delivery over LTE using Adaptive Modulation and Coding," .

[12] R. Skupin, C. Hellge, T. Schierl, and T. Wiegand, "Packet level video quality evaluation of extensive H. 264/AVC and SVC transmission simulation," *Journal of Internet Services and Applications*, pp. 1–10, 2011.

[13] T. A. Le and H. Nguyen, "Centralized and distributed architectures of scalable video conferencing services," in *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on*. IEEE, 2010, pp. 394–399.

[14] T. A. Le, H. Nguyen, and H. Zhang, "Scalable Video transmission on overlay networks," in *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on*. IEEE, 2010, pp. 180–184.

[15] M. Hu, H. Zhang, T. A. Le, and H. Nguyen, "Performance evaluation of video streaming over mobile WiMAX networks," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*. IEEE, 2010, pp. 898–902.

[16] Quoc T. Tran, Tien A. Le, and Hang Nguyen, "WiMAX-based overlay conferencing service," in *Intelligence in Next Generation Networks (ICIN), 2011 15th International Conference on*, Oct. 2011, pp. 17–22.

[17] S. Wenger, Y. K. Wang, T. Schierl, and A. Eleftheriadis, "RTP payload format for SVC video," *draft, Internet Engineering Task Force (IETF)*, Sept. 2009.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] I. Rec, "P. 800: Methods for subjective determination of transmission quality," *International Telecommunication Union*, 1996.

[25] 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.

[26] 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.

[27] 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.

[28] Tien Anh Le; Hang Nguyen; Hongguang Zhang, "EvalSVC - an evaluation platform for scalable video coding transmission," in *14th International Symposium on Consumer Electronics (ISCE 2010)*, Braunschweig, Germany, June 2010, pp. 85–90.

[29] D. Constantinescu and A. Popescu, "Implementation of Application Layer Multicast in OverSim," in *4th Euro-FGI Workshop on" New Trends in Modelling, Quantitative Methods and Measurements"*. Citeseer.