



# Centralized and distributed architectures of scalable video conferencing services

Tien Anh Le, Hang Nguyen

**Abstract**—The Multipoint Control Unit-based centralized architecture and Application Layer Multicast-based distributed architecture are mainly used for data distribution in video conferencing services. With the contribution of Scalable Video Coding, the latest extension of Advanced Video Coding, video conferencing services are being further researched to support terminals' scalability. The main contribution of this research is to answer a fundamental question before a video conferencing service can actually be designed: which service architecture is more suitable for a SVC-based video conferencing service and in which condition? In order to do that, a framework for evaluating SVC-based video conferencing service has been built; intensive simulation results have been also obtained from simulation scenarios which have been designed from the analysis models of the two scalable video conferencing service architectures. The obtained results show that, with a much higher capacity of the MCU, the MCU-based architecture can only guarantee an almost similar video quality with the ALM-based architecture at a trade-off of at least three times higher end-to-end delay.

**Index Terms**—Scalable video coding; video evaluation platform; video conference; service architecture;

## I. INTRODUCTION

Nowadays, multimedia conferencing services are becoming an essential part in both our business and everyday activities. They can save us a lot of time and resource in comparison with the traditional face-to-face meetings. However, in order to fully replace the conventional end-to-end meeting methods, multimedia conferencing services have to provide their users the highest possible quality of experience and a very flexible connectivity to the conferencing service platform. Among the multimedia flows that build up a multimedia conferencing service, video is the most important as well as the most difficult medium that the multimedia conferencing service has to distribute to its participating users (e.g. multimedia conferencing services are usually called video conference). The video medium is apparently very important because it contributes the most to the users' quality of experience and transfers most of the conference information to participants. It is difficult to be distributed because the video stream is always the most heavy stream in the multimedia flow, so it is very difficult to transport the video stream to all users without a serious quality decrement and a heavy network congestion.

People are using different kinds of terminals to access to the multimedia conferencing service. These terminals are different in their computational capacities, screen

resolutions, and communication bandwidths. However, many service providers are still 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), so it is impossible to change the encoding parameters without using a trans-coder and/or a trans-rater. A much more flexible 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. Another issue for multimedia conferencing services is how we can effectively distribute the video bit-stream from its source to many other participants. Most commercial video conferencing services are now using Multipoint Control Unit (MCU) for mixing and distributing video bit-streams to all participants in the conferencing session. When a MCU is used, this middle-box collects video streams from all participating users of the conferencing session. It then mixes all of these incoming streams, making necessary trans-codings or trans-ratings before sending back the mixed streams to all participating users. This process may cause delays, a single point of failure and bottleneck for the entire communication session. Regarding multicast mechanisms, IP-Multicast[2] is the

Authors are with the Department of Wireless Networks and Multimedia Services, Telecom Sud Paris, France, 91011. Phone: +33 (0)1 60 76 66 63, Fax: +33 (0)1 60 76 45 78, E-mail: {Tien\_anh.Le, Hang.Nguyen}@it-sudparis.eu. This work was supported in part by CAM4Home, an European project.

first attempt to address this problem. However, many deploying problems are still preventing IP-Multicast from being supported worldwide[3]. An alternative solution is Application Level Multicast(ALM). The key concept of ALM is the implementation of multicasting functionality as an application service instead of a network service. It has excellent advantages over IP-Multicast: easier and possibly immediate deployment over the Internet without any modification of the current infrastructure and adaptable to a specific application. In order to evaluate the video conferencing services, designers and researchers are really in-need of an evaluation tool for video transmission which is specially tailored for the evaluation of SVC-based video conferencing service. Since most of the current video conferencing systems are now using the MCU-based centralized architecture or ALM-based distributed architecture to provide their conferencing service, an evaluation platform for these two scalable video conferencing services is required. In[4], an early attempt was made to evaluate the SVC transmission. However, it used an all-in-one solution which is incompatible to H.264/AVC evaluation platforms. It has not fully supported the extended NALUs. As a result, it couldn't support Sub-sequence Parameter Set (NALU 15). This NALU is very important since it contains decoding information for a sequence of NALUs. If this NALU is dropped along the transmission, many SVC frames will be effected and though cannot be decoded. Last but not least, it cannot provide any interface to popular simulation platforms (NS-2 or OverSim), so that it can only be used in field tests but not in theoretical analysis. So far, the research community depends on EvalSVC[5] 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. EvalSVC has supported up to latest extensions of the SVC codec. The main contribution of this research is to form a many-to-many simulation platform for video conferencing service architectures using the NS-2 interface of EvalSVC. Then, both the centralized MCU-based and ALM-based service architectures of the SVC-based video conferencing service will be measured and evaluated for the SNR scalability by using the EvalSVC platform. Analysis will be made on the obtained results.

## II. EVALUATION OF SCALABLE VIDEO CODING TRANSMISSIONS

In[5], the evaluating problems of Scalable Video Coding transmissions was first addressed and solved by using the EvalSVC platform. The most difficult problem is that the full SVC's Network Abstraction Layer Unit (NALU) extensions haven't been fully defined and standardized. However, it should be noticed that, the basic NALU extension types (e.g., types 14, 15, 20) have been spared for SVC extensions from the AVC NALU types.

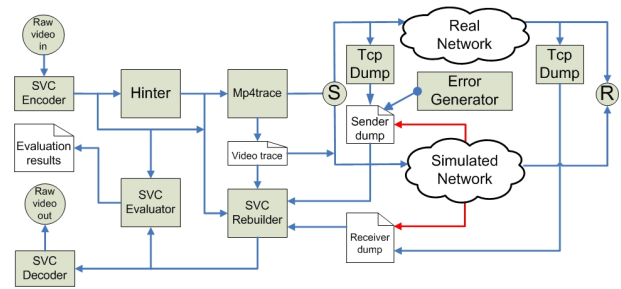


Fig. 1. EvalSVC's diagram[5].

So only these basic NALU extensions are supported in the 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[6], are out of EvalSVC's scope.

Among the three basic NALU extension types, NAL unit type 14 is used as a 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[7]. 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. Based on these IDs, we can choose to drop all packets of the same enhancement layer(s) according to a chosen scalability (whether it is quality, temporal, spatial or combined scalability).

Fig. 1 illustrates main components of our EvalSVC platform.

Basically there are three available metrics which can be used to evaluate the performance of the scalable video coding bit-stream. These three metrics can be further divided into subjective and objective quality measures. The most popular objective metric is peak signal-to-noise ratio (PSNR). PSNR can be used to assess the resulting video quality by calculating the Mean-square Error between the original raw video bit-stream before the encoding process at the input and the possibly corrupted received video bit-stream at the output, frame-by-frame. Normally, the luminance component of the video is used in PSNR comparison since it is the most important component of a video frame.

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

$$MSE(s, d) = \sqrt{\frac{1}{N_{col}N_{row}} \sum_{i=0}^{N_{col}} \sum_{j=0}^{N_{row}} [Y_S(n, i, j) - Y_D(n, i, j)]^2}$$

In which:

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

SSIM is another objective metric for measuring the similarity between two images[8]. 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 metric is calculated on various windows of an image. The measure between two windows  $x$  and  $y$  of common size  $M \times N$  is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (1)$$

In which:

- $\mu_x, \mu_y$  are the mean values of  $x$  and  $y$ ,
- $\sigma_x^2, \sigma_y^2$  are the variance values of  $x$  and  $y$ ,
- $\sigma_{xy}$  is the covariance of  $x$  and  $y$ ,
- $c_1, c_2$  are two variables to stabilize the division with weak denominator;

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[9].

All of the three metrics are used in EvalSVC as the main measures to evaluate the SVC transmissions. OverSim[10] 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. In[11], an interface between the EvalSVC platform and the multicast simulations using OverSim has been developed. Together with the NS-2 interface which has been already integrated, the entire EvalSVC platform can provide necessary simulation evaluations for the comparison between SVC-based video conferencing service architectures.

### III. VIDEO CONFERENCING SERVICE ARCHITECTURES

The multimedia conferencing services are mainly composed of a media distribution plan and a signaling plan[12]. In this research, we concentrate on the media distribution plan of the conferencing service in which the evaluation of video distribution architecture is in focus.

In Fig.2, a centralized video conferencing service architecture using MCU is demonstrated. MCU is a central device which has a larger bandwidth connection and more powerful computation capacity than others. The MCU collects video bit-streams from all participants, mix them into a common video frame (the active speaker's video occupies a big part of that common frame, other participants' videos can be displayed as small thumbnails around that big video) and send back to all participants. In a SVC-based video conference service architecture, participants

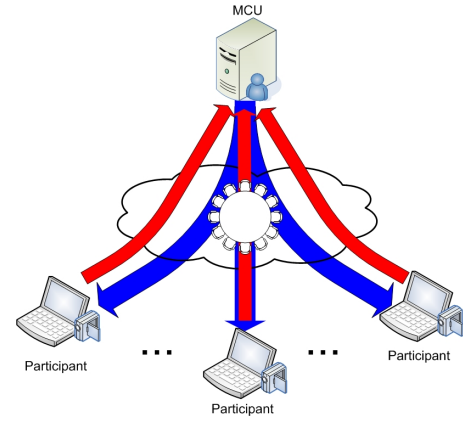


Fig. 2. Centralized video conferencing architecture.

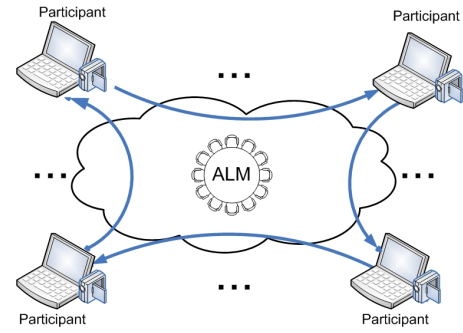


Fig. 3. Distributed video conferencing architecture.

can encode their video using Scalable Video Coding before sending them to the MCU. In order to support those incoming SVC bit-streams, a Scalable Video Conferencing Server (SVCS) was proposed to replace the conventional MCU in[13]. The SVCS collects SVC bit-streams from all participants. It can choose to discard enhancement layers to save its bandwidth and computational capacity. Then, the received bit-streams will be decoded and then mixed together to form a common video frame. That video frame will be encoded again using a SVC encoder. The output SVC bit-stream, comprising of a base layer and enhancement layer(s), is then sent to all other participants.

The distributed video conferencing service architecture is shown in Fig.3. In this architecture, each participant is also armed with a SVC encoder. All participants form an application layer multicast (ALM) tree to distribute their SVC bit-stream to the rest of the conference. The ALM algorithm has considered various conditions of the participants' terminals (such as bandwidth, delay, capacity, etc.) before building the multicast tree. As a result, the participant having a higher capacity will more likely become a forwarder who is capable of relaying the SVC bit-streams to another participant, dropping enhancement layers when necessary in order to meet the overall requirements of the conference session. The main difference between the centralized and distributed architectures from the user's point of view is that instead of receiving only one video stream, each terminal will receive video streams from all other participants. It then has the full control whether it wants to

receive a video bit-stream from a particular participant or not. Since the video bit-streams are SVC-based, participants also have the right to discard enhancement layers if they find that they are not capable of receiving all layers. So basically, the participants will have the full control on how many video bit-streams and how many layers they want to receive depending on their preferences and capacities.

Each conferencing service architecture has its own advantages and disadvantages. The MCU-based centralized architecture may provide a lower network load but with a higher delay and a decrease in video quality. The ALM-based distributed architecture can provide more flexible receiving options with possibly higher video quality. However, the ALM may cause delay and more traffic load on the network. We compare the performance and quality of SVC-based video conference service architectures using simulations and our EvalSVC platform.

We set up a simulation plan of a video conferencing service serving 16 participants simultaneously over the Internet. Two simulation scenarios are built reflecting the centralized and distributed architectures.

#### A. MCU-based centralized video conferencing service architecture

The NS-2 interface of EvalSVC is used to build and evaluate the first simulation scenario. In this scenario, the participating users connect to the MCU server using a simulated network topology generated by GT-ITM. The topology generated by GT-ITM emulate the real network environment on the Internet using a hierarchical transit-stub configuration of 1250 nodes and about 6000 physical links[14]. Each physical link will have random values of delay, bandwidth, and Packet Error Rate (PER). Each participating user connects to the network using a DSL access interface. The MCU uses a higher bandwidth and lower delay interface to connect to the network. Each participating user sends the same sample video of 1065 frames to the MCU. At the MCU, we prepare in advance a composite SVC video of 1065 frames as if it is mixed and encoded by the MCU after receiving 16 video streams from all 16 participating users. As soon as the MCU receives packages from all participating users, it will send the corresponding video frame of the prepared composite video back to all participants. The MCU uses a 10 Mbps access connection to receive and transmit video bit-streams. By using this two-step scenario, we can emulate the MCU's operations.

The performance evaluation is also two-fold. In Fig.4a, 16 SVC bit-streams are sent from participants to the MCU via different channels. The MCU will receive 16 possibly corrupted SVC bit-streams. The mixer composes all corrupted SVC bit-streams to make a composite SVC bit-stream and send back to all participants. Each participant will receive a composite SVC bit-stream. Of course this composite SVC bit-stream may be double-corrupted due to the quality of the channel. Since the evaluation metrics such as PSNR and SSIM are frame-based (e.g. the source and corrupted bit-streams must be identical in order for

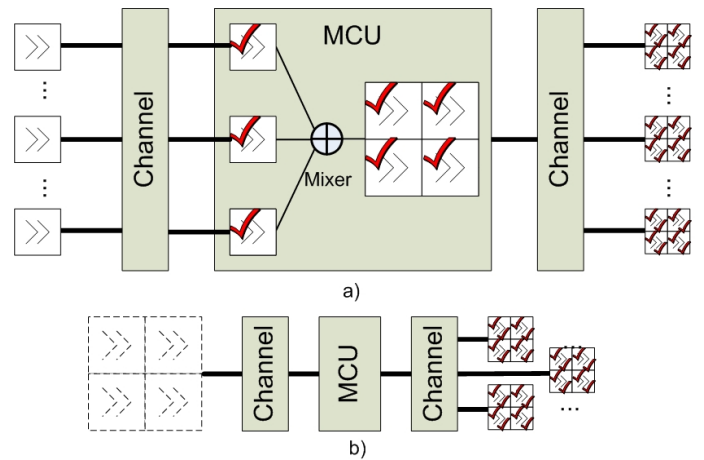


Fig. 4. MCU-based centralized video conferencing simulation scenario.

the evaluator to give out a reasonable result), an analysis simulation model is shown in Fig.4b. We assume that the video composition process is loss-less, so we can replace the transmission of 16 separated SVC bit-streams to the MCU by the transmission of a single loss-less composite SVC bit-stream. Afterward, the simulation and analysis models use an identical process to transmit the corrupted SVC bit-stream from the MCU back to the participants. The overall simulation scenario for the MCU-based centralized video conferencing service architecture is as follow. Firstly, 16 raw videos from 16 participants compose a loss-less composite raw video. Next, 16 raw videos are encoded by a SVC encoder at each participant and then separately sent over the channel to the MCU. The MCU will decode 16 possibly corrupted SVC bit-streams, and then form the corrupted raw composite video. That raw video will then be encoded again using a SVC encoder and then sent back to participants. Each participant will receive a double-corrupted SVC bit-stream, and then decode it to obtain a double-corrupted raw video. This video is compared to the loss-less raw video composed at the beginning of the process to evaluate its PSNR and SSIM. In this scenario, we use the NS-2 interface of the EvalSVC framework to build our simulation.

#### B. ALM-based distributed video conferencing service architecture

The simulation scenario for the ALM-based distributed video conferencing service architecture is in fact quite straight forward. We use NICE[15], a popular ALM algorithm to multicast the SVC bit-streams from each participant to all others. NICE only uses a delay-type cost function to build and to maintain its ALM tree (with a clustering, layering structure). By sending and receiving periodic heartbeat messages containing delays between nodes within a cluster, peers will decide whether it should elect a new cluster-leader. Changing cluster-leaders provokes changing and rebuilding the entire NICE tree. The underlay network topology is identical with the one used in the MCU-based centralized scenario. In this scenario, all

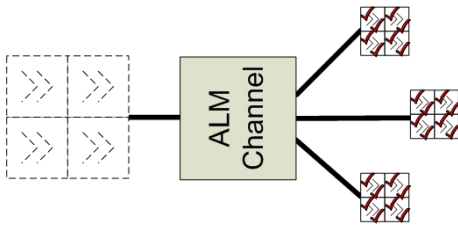


Fig. 5. ALM-based distributed analysis scenario.

peers will join together to form an overlay. By comparing distances among peers, a media distributing tree is built to multicast the SVC bit-stream. In this case, the bit-stream will not be sent directly, it may be relayed by forwarder(s) along the media distributing tree before reaching its final destination. According to the network conditions, enhancement layers can be dropped at the forwarder(s) in order to maintain the continuity of the video communication session. Participating nodes use a 1 Mbps connection to transmit, forward and receive video bit-streams. Afterward, each participant will receive SVC bit-streams from all members of the conference session. Since evaluation metrics such as PSNR and SSIM are only comparing identical bit-streams, in order to evaluate the performance between the distributed and centralized video conferencing service architectures, an analysis model as the one shown in Fig.5 is used. Video bit-streams are composed at senders before transmitting to the ALM network to prepare a lossless composite video for later evaluation. At each receiving node, all bit-streams are decoded and then composed to make a possibly corrupted composite video. The corrupted bit-stream will be compared with the original composite bit-stream for PSNR and SSIM. In this scenario, we use the Oversim interface of the EvalSVC framework to build our simulation.

#### IV. SIMULATION RESULT

The results are shown in Fig.6 and Fig.7. These results compare the average quality of the received video bit-streams in two architectures using PSNR and SSIM. SNR scalability is used to encode the video bit-streams since it is the best error-resistance SVC type[5]. We can see that, with a much higher capacity MCU, the centralized architecture can provide an almost similar video quality with the distributed architecture.

TABLE I  
AVERAGE END-TO-END DELAY [s].

Types	Uplink	Downlink	MCU e2e	ALM e2e
Packet	0.0327191	0.232448	0.265167	0.09171
Frame	0.0328203	0.256979	0.289799	0.108484

Table I shows the average end-to-end delay comparison between the MCU-based and ALM-based service architectures. It is clear that, from both packet level and frame level, the MCU-based architecture has almost three times the end-to-end delay than the ALM-based architecture.

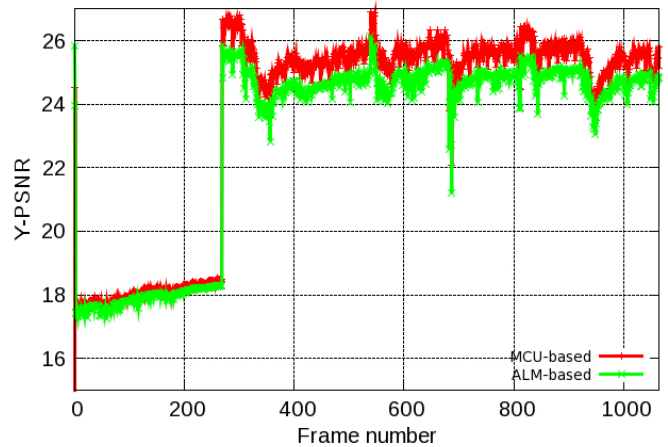


Fig. 6. Y-PSNR comparison of MCU-based and ALM-based scalable video conferencing service with SNR SVC encoding method.

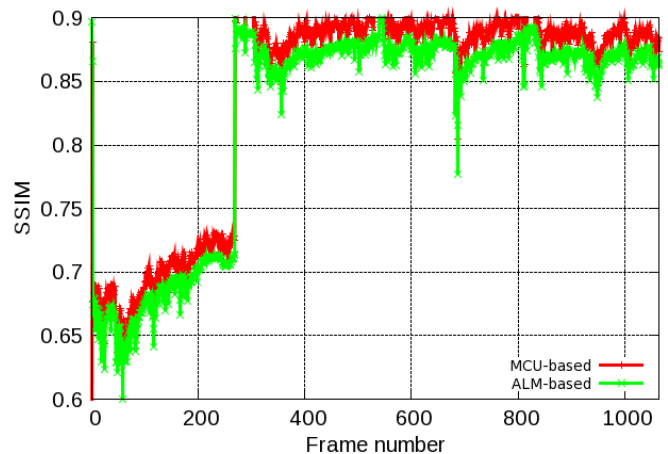


Fig. 7. SSIM comparison of MCU-based and ALM-based scalable video conferencing service with SNR SVC encoding method.

More specifically, the direction from nodes to the MCU (uplink) has a small end-to-end delay but the direction from the MCU to nodes (downlink) has a very high delay leading to its very high end-to-end delay. In this result, a very big delay created by the mixing, encoding/decoding processes at the MCU has not been considered. Therefore, the real end-to-end delay of the MCU-based architecture should be much higher. In this case, a very high price is required if we want to keep the end-to-end delay of the MCU-based scalable video conferencing service architecture below a recommended threshold of 200 ms[16]. Even though a 10 times higher bandwidth and about 16 times higher computational capacity have been applied at the MCU (when all participating peers is having a similar bandwidth and computational capacity with the ALM-based scenario), it can only guarantee a slightly higher quality of the received videos with the trade-off of a very high end-to-end delay in comparison with the ALM-based architecture.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have succeeded in comparing the performance and quality of the centralized, MCU-based and distributed, ALM-based scalable video conferencing architectures. Intensive simulation scenarios have been built based on the EvalSVC, a scalable video coding evaluation framework and its interfaces to NS-2 and OverSim simulation tools. The obtained results have shown that, when 10 times higher bandwidth, approximately 16 times higher computational capacity, and similar participants' capacities are applied, the MCU-based architecture may obtain a slightly higher average video quality at participants. However, a very high queuing with a high packet dropping rate at the MCU show a highly potential single point of failure in the network. Especially when the number of participants increases into hundreds or even thousands, the ALM-based architecture is foreseen to out-perform the MCU-based architecture. A very serious problem of the MCU-based architecture is that its end-to-end delay is about 3 times higher than the ALM-based architecture even without considering the very high delay of the mixing, encoding/decoding process at the MCU. Based on the result and simulation scenario obtained from this research, theoretical analysis can be built to provide a more solid background on the performance of the centralized MCU-based scalable video conferencing services.

## 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] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 2, pp. 85–110, 1990.
- [3] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, 2000.
- [4] A. Detti, G. Bianchi, W. Kellerer, and Others, "SVEF: an open-source experimental evaluation framework," in *In Proc. of IEEE MediaWIN 2009, Sousse, Tunisia*, 2009.
- [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. K. Wang, T. Schierl, and A. Eleftheriadis, "RTP payload format for SVC video," *draft, Internet Engineering Task Force (IETF)*, September 2009.
- [7] 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.
- [8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [9] I. Rec, "P. 800: Methods for subjective determination of transmission quality," *International Telecommunication Union*, 1996.
- [10] 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.
- [11] 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.
- [12] S. Firestone, T. Ramalingam, and S. Fry, *Voice and video conferencing fundamentals*, Cisco Press, 2007.
- [13] Eleftheriadis Alexandros, Civanlar M. Reha, and Shapiro Ofer, "Multipoint videoconferencing with scalable video coding," *Journal of Zhejiang University - Science A*, vol. 7, no. 5, pp. 696–705, 2006.
- [14] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings IEEE INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation*, 1996, vol. 2.
- [15] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2002, p. 217.
- [16] R. Itu-T and I. Recommend, "G. 114," *One-way transmission time*, vol. 18, 2000.