Scalable Video Streaming with automatic content adaptation

Livio Lima, Nicola Adami, Riccardo Leonardi
DEA - University of Brescia, via Branze 38, 25123 Brescia, Italy
Email: {firstname.surname}@ing.unibs.it

***Abstract*—Scalable Video Coding technology enables flexible and efficient distribution of videos through heterogeneous networks. In this regard, the present work proposes and evaluates a method for automatically adapting video contents, according to the available bandwidth. Taking advantage of the scalable video streams characteristics, the proposed solution uses bridge firewalls to perform adaptation. In brief, a scalable bitstream is packetized by assigning a different Type of Service field value, according to the corresponding resolutions. Packets corresponding to the full video resolution are then sent to clients. According to the given bandwidth constraints, an intermediate bridge node, which provides Quality of Service functionalities, eventually discards high resolutions information by using appropriate Priority Queueing filtering policies. A real testbed has been used for the evaluation, proving the feasibility and the effectiveness of the proposed solution.**

I. Introduction

The distribution of a given video content to several clients, characterized by having different rendering capabilities and connected by means of links with different bandwidth restriction, is in general a heavy task. Till today, this goal has been achieved mainly using two alternative methods: by using video transcoders, properly placed in the nodes of the distribution network or by encoding and successively distributing different coded version of the same content. These methods are clearly inefficient, in the first case a high computational power is required for successively adapting the content while in the second there is a waste of storage space and channel bandwidth. Additionally, these solutions are not effective in case of dynamic bandwidth variation.

The recently developed Scalable Video Coding (SVC) methods [1], [2], are a key technology to overcome these limitations. SVC codecs generate a bitstream with a unique feature, the possibility of extracting decodable sub-streams corresponding to a scaled version, i.e. with a lower spatio-temporal resolution or a lower quality, of the original video. Moreover, this is achieved providing coding performances comparable with those of single point coding methods and requiring a very low sub-stream extraction complexity, actually comparable with read and write operations. Scalability is then suitable to ease video content adaptation when there are bandwidth fluctuation or when the bandwidth required to trasmit the requested resolution is not available. In these situations, only a bit-stream subset can be transmitted, or forwarded by one of the node in the network to the clients.

Scalable Video Coding is a relatively new technology and a commonly adopted delivery method has not been defined yet. However, several solutions have been proposed, concerning different aspects of a complete scalable video streaming chain. In [3], a MPEG4-FGS scalable stream, with one spatial resolution and multiple quality layers, is transmitted using a client-server collaborative system with the aim of avoiding congestion. The client estimates the rate of occupancy of its receiving buffer, which is assumed to depend on the congestion level. The estimation is then transmitted to the server, on a feedback channel, which dynamically adapts the quality of transmitted video in order to avoid congestion at the client's side. Similar approaches have been proposed by Nguyen et al. [4] and by Hillestad et al. [5] in the context of wireless video streaming. In [6] the use of Differentiated Services (DiffServ) [7] of IP protocol is used to provide QoS with MPEG4-FGS and H.264-SVC. The main drawback is that only two classes of service are used, Expedited Forwarding (EF) for base layer and Assured Forwarding (AF) with three group of priority to differentiate the types of pictures (I, P and B) in the enhancement layer. In [8] a real-time system based on the scalable extension of H.264 (H.264-SVC) scalable and MPEG-21 Digital Item Adaptation (DIA) is proposed. In particular QoS is obtained using Adaptation QoS (AQoS) and Universal Constrain Description (UCD) tools of MPEG-21 DIA. The main drawback of this approach is the complexity of MPEG-21 descriptors determination, which depends on the content itself, needed for the configuration of the adaptation nodes.

This work also aims at providing solutions for scalable video content adaptation by considering a real client-server application framework. The network architecture, here considered, is composed by a server that can store and send the video contents, different clients and a bridge that adapts the transmitted stream according to the available bandwidth. The key aspect of the proposed application, with respect to the work described in [8], is the way video adaptation is realized. Instead of using dedicated extractors for scaling the distributed stream, the system relay on the packet filtering policies realized by network Quality of Service (QoS).

The presentation is organized as follows. Section II provides some hints on Scalable Video Coding focusing on the generated bitstream structure. Section III describes the structure and the elements of the considered distribution network. In Section IV the obtained results are presented.

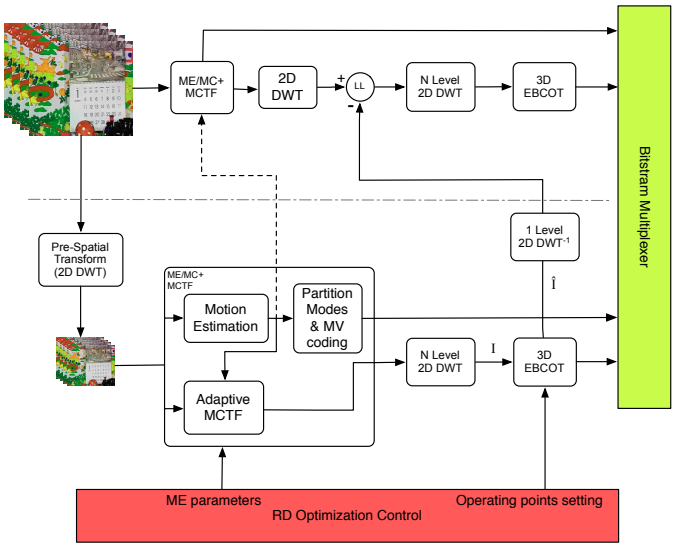

Fig. 1. Wavelet based Scalable Video Codec architecture

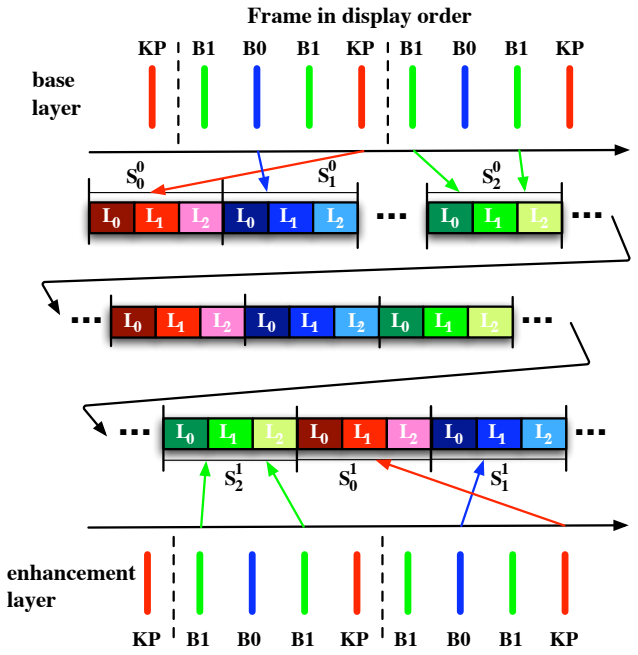

Fig. 2. Bit-Stream Organization

II. Scalable Video Coding Elements for Networking

The video coding system hereafter considered is the STP-Tool [2], a wavelet based scalable codec. It provides good compression performances, with respect to the state of the art in SVC, expecially for High Definition (HD) applications [9]. As it can be seen in Figure 1 the original video sequence is down-sampled in order to generate the desired spatial resolutions.

Temporal scalability is obtained through the use of Motion Compensated Temporal Filtering (MCTF). The input sequence is initially decomposed into Group Of Pictures (GOP) which are independently processed by applying a MCTF, producing a hierarchical temporal decomposition of the original. Spatial scalability is achieved with a closed loop spatial prediction. Starting from the lowest spatial resolution, the quantized temporal subbands $\hat{I}$ are used to predict the temporally filtered signal, generated by the MCTF, at higher resolution (see Fig. 1). All this information, except the motion one, are then lossy coded generating a progressive bitstream, which provides quality scalability. **From the generated compressed stream, is then possible to extract the information required to decode any spatio-temporal and quality resolution (working point), allowed by the used hierarchical decomposition. The lowest decodable working point is usually referred as base layer (BL). All the other decodable video versions, attainable by adding to the base layer the differential information required to scale up along the desired dimensions, are usually referred as enhancement layer (EL)**

Figure 2 shows the details of the bitstream structure considered in this work. It provides two level of spatial resolution, three level of temporal resolution and quality with a GOP size equal to 4. For each spatial resolution, the last picture of every GOP is referred as Key-Picture (KP) and is intra-coded. All the other pictures between two consecutive KPs are compensated using bi-directional motion estimation: the picture $B0$ uses the previous key-picture and the key-picture belonging to the same GOP as reference, while the pictures $B1$ use one key-picture and the $B0$ as reference. Hence all GOPs are represented by six different streams S_i^j, where index $i = \{0, 1, 2\}$ is related to temporal subband (0 for KP, 1 for B0, 2 for B1) and index $j = \{0, 1\}$ is related to spatial resolution (0 for low and 1 for high resolution). As previously mentioned, each sub-bistream S_i^j could be generated with multiple quality layer L_k (three in this setup) where the decoding of a particular quality layer L_K needs of all the previous layers L_k, $k = 1, ..., K-1$.

III. Testbed: Application and Infrastructure

The proposed automatic video content adaptation method has been evaluated considering a HD video-on-demand application, as the Home distribution of HD audio-visual contents. In this context where a given video has to be streamed to several different devices, it will be helpful to have a mechanism to automatically scale the content according to the bandwidth provided by the connection links. This has been realized by using the system depicted in Figure 3 which is composed by three main elements: a Server Repository, a Client and a Bridge, described in the following subsections.

A. Client and Server

The client starts the communication by requesting a given video content to the server, specifying the desired spatial, temporal and quality resolution. Additionally it can specify which scalability dimension should be preferably used during

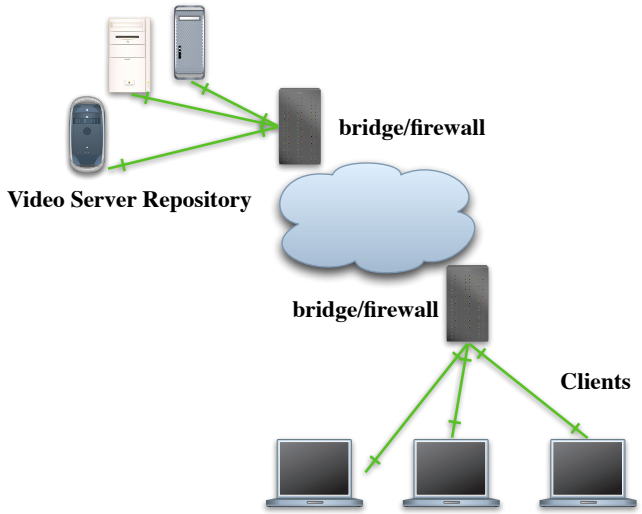

Fig. 3. Network Architecture

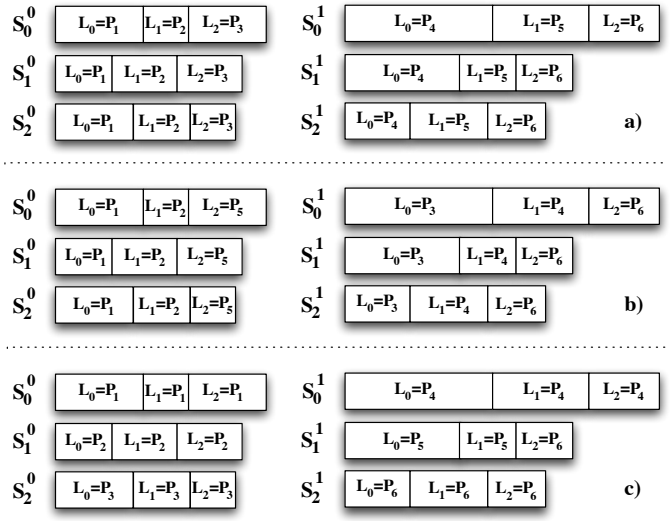

Fig. 4. Priority Assignment: a) configuration C1, b) C2 and c) C3

adaptation. The server will then proceed to the extraction of the requested information from stored scalable bitstream and to packet the data to delivery. During this process, the server will assign different priorities to data corresponding to distinct stream layers. Three possible configurations are supported and each of them can use up to six priority values ($P_1 > P_2 > P_3 > P_4 > P_5 > P_6$), as shown in Figure 4. It is important to note that 6 values of priority are appropriate with respect to the number of scalability layers and settings here considered, but it is easily extendable for supporting a larger number of scalable layers. These operations are very fast because given the bitstream structure shown in Figure 2, the server's task is limited to select the requested data and rearrange it in packets according to the desired video resolution and the used application protocol.

In configuration C1 and C3 high relevance is given to the lowest spatial resolution, with all the sub-bitstreams S_i^0 of base layer at higher priority than the enhancement layer's sub-bitstreams S_i^1. The difference is on how priorities are assigned within the spatial resolution. For example, C1 gives more importance to quality layers, by assigning to the data representing the quality layer 0 a higher priority than quality layers 1 and 2. On the other hand, C3 favours the transmission at low frame-rates because data of lower temporal resolution (S_0^j) has higher priority than "temporal detail" subbands (S_1^j and S_2^j). These settings could be useful for clients with low resolution devices, like mobile devices. C2 favours the quality layers of both spatial resolutions, for example by assigning low priority to the parts of the bitstream related to quality layer 3. Clearly the parts of bitstream for quality layers 1 and 2 of the base layer have high priority because this information is essential for decoding the enhancement layer. This configuration may be adopted when a good tradeoff between quality and spatial resolution is desired.

B. Protocols

The protocols suite, used for the delivery of data, is formed by an application protocol, specifically developed for this application, using UDP over IP. The developed application protocol also transports the information needed by the client to correctly reassemble the subband streams and to arrange the data into UDP packets at server's side. At transport level, UDP has been preferred to TCP because of low-delay and time sensitive features of real-time video content. In case of round trip delay considerably smaller than decoder buffering period, the retransmission of lost data may be effectively adopted because the needed data can reach the decoder side within the display period. In this case the functionalities of TCP protocol could be useful to improve the performance of the system. In this work, the use of decoder buffer and eventually retransmission have been intentionally avoided in order to better test the scalability properties of the compressed stream. Assuming the available bandwidth sufficient to transmit all data needed to correctly decode the base layer, the problem of missing data has been handled by allowing the decoder to opportunely scale the considered visual content.

The QoS is obtained at network level using the Type Of Service (TOS) field of IP protocol. The TOS byte in the IPv4 header has had various purposes over the years, and has been defined in different ways by five different RFCs. A complete review of the historical definitions for the TOS field can be found in RFC 3168 [10]. The RFC 2474 and 2780 replaced the TOS field by Differentiated Services Field (DS), splitting the 8-bit field in a 6-bit Differentiated Services CodePoint (DSCP) and 2-bit Explicit Congestion Notification (ECN), so in the follow we refer as DS field instead of TOS. Successive RFCs, like RFC 2475 for DiffServ [7], suggest a way to use the 6-bit DSCP to differentiate the type of traffic to support QoS.

In the proposed architecture the DSCP field of IP protocol

is used to transport the priority type related to parts of the compressed codestream. As shown in Figure 4, in the particular example considered, the subbands data can have six different types of priority (from P_1 to P_6). This value of the priority is copied in the DSCP field and is then used by the bridge, as will be explained more in details in the next section, to apply the filtering policies, i.e. automatic content scaling. The six bits of DSCP enable 64 different levels of priority and therefore a very flexible representation of different parts of the data stream.

C. Quality of Service

The bridge is the element responsible for the management of QoS by correctly applying the scheduling and filtering policies. It is based on OpenBSD operating system, which is a popular choice for those who demand stability and security from their operating system. This platform embeds Packet Filter (PF), which is well known to be a proven, high-performance and innovative packet filtering tool. The QoS with PF is obtained using Alternate Queueing framework (ALTQ) recently integrated in latest release of OpenBSD. ALTQ provides queueing disciplines to realize QoS and is used to assign packets to queues for the purpose of bandwidth control. The scheduler defines the algorithm used to decide which packets get delayed, dropped or sent out immediately. There are two schedulers currently supported in the OpenBSD implementation of ALTQ: CBQ and PRIQ.

Class Based Queueing (CBQ): split the available bandwidth between the queues in a hierarchical way. A root queue is defined with the total amount of the available bandwidth. From the root queue different children queue are created, and each one take a partition of the bandwidth of the root's one. From each children queue other children queue of lower level could be defined, each one with a partition of the bandwidth of the mother's one, and so on. An useful option is that each queue (except the root's one) can borrow bandwidth from the mother's queue if the mother queue has a temporary unused bandwidth. CBQ can define a priority level for each queue, in order to process as first the queue with high priority in case of congestion.

Priority Queueing (PRIQ): a root queue is defined with the total amount of the available bandwidth, then multiple queues each one with a priority level are defined on network interface. In PRIQ the queues are flat, so it is not possible to define sub-queues. When all the packets of the high priority queue are forwarded, the scheduler processes the packets in the next queue in priority order, an so on. The capability of the scheduler to process the queues depends on the available bandwidth and how the queues are created. In fact the bandwidth define the throughput and so the ability of the scheduler to process packets in time unit. So if the queues with high priority receive a constant flow of packets and the available bandwidth is too low, the scheduler will spend the whole time to process the high priority queues and all the packets of low priority queues will be discarded.

For each scheduler, different algorithms can be selected for queuing and discarding packets. The simplest one is to discard all the packet that should be filled in an already full queue. Other algorithms commonly used are Random Early Detection (RED), Explicit Congestion Notification (ECN) and Random early detection with In/Out (RIO).

In the experiments performed only the PRIQ scheduler has been used, because the native priority structure well adapts to the video application in which different parts of the codestream have different relevance. CBQ scheduler is not suitable for the proposed application since it requires to assign a fixed bandwidth to each queue and this could not be an easy task. For example, supposing to assign a different queue to packets of different temporal subbands an estimation of the rate for each temporal resolution would be required, but this strongly depends on the motion features of the particular video sequence.

As previously described, the filtering rules are based on the inspection of DS field of IP protocol. At scheduler level, 6 different queues q_i are defined each one with a priority value $p(q_i)$, where $p(q_1) > p(q_2) > ... > p(q_6)$. When packets arrive at the bridge they are assigned to different queues inspecting the DS field of IP protocol. That is, if the 6-bit DSCP is equal to P_1, where the correspondence between P_i and parts of bitstream has been explained in section III-A, the packet is assigned to queue q_1, if it is equal to P_2 to queue q_2 and so on. In this way we are sure that packets with high priority will be processed by the bridge even if the available bandwidth is low. The way in which the packets are processed or discarded depends on the priority value $p(q_i)$ assigned to the corresponding queue q_i and the congestion algorithm used.

IV. Experimental results

Two different types of experiment have been performed, with fixed and variable bandwidth. Two resolutions have been considered, the base layer at 960x512 pixels and the enhancement layer at 1920x1024 pixels, each one with two levels of temporal decomposition that enables three frame-rate: 50, 25 and 12.5 Hz. As previously described, the bitstream has been generated with three quality layers at about 39, 35 and 32 dB in PSNR. In line with HD application requirements, all videos have been produced forcing a high and near constant quality, i. e. avoiding flickering. The last constraint requires to use a limited GOP size and to encoded a relevant amount of information to adequately correct the prediction error in B frames. As a consequence, the required transmission rates are quite high if compared with normal video streaming applications. **Nevertheless, the proposed method is applicable also to smaller transmission rates.**

In **fixed bandwidth** experiments, the transmission of the video codestream has been performed with different values of the maximum available bandwidth set in the firewalll. In particular, a value B1 has been set in order to enable the transmission of the full bitstream, B2 is equal to $2/3$ B1, B3 to $1/3$ B1 and B4 to $1/6$ B1. In this experiment the three

bandwidth	50 Hz	25 Hz	12,5 Hz
BL B1	38.6	38.6	38.8
BL B2	38.6	38.6	38.8
BL B3	38.6	38.6	38.8
BL B4	38.6	38.6	38.8
EL B1	38.5	38.6	39.2
EL B2	36.6	36.7	37.2
EL B3	32.2	32.3	32.6
EL B4	30.8	30.7	30.8

TABLE I
PERFORMANCE SUMMARY FOR CONFIGURATION C1

bandwidth	50 Hz	25 Hz	12.5 Hz
BL B1	38.6	38.6	38.8
BL B2	38.6	38.6	38.8
BL B3	34.9	34.8	35.0
BL B4	31.2	31.4	31.5
EL B1	38.5	38.6	39.2
EL B2	36.5	36.6	37
EL B3	33.9	34.2	34.7
EL B4	29.8	29.9	30.1

TABLE II
PERFORMANCE SUMMARY FOR CONFIGURATION C2

configurations described in Section III-A has been tested, in order to show the different decoding performance and the flexibility of Scalable Video Coding. The summary of the experiments is shown in Table I, II and III, where for each configuration the average PSNR value **over the frames** is reported. **The results shown in the tables confirm that the firewall correctly discards packets according to the used priorities setting. From Table I it can be seen as, according to the priority setting $C1$, a near constant quality is achieved for the BL spatial resolution at all the considered frame rates and bandwidths while for the EL the quality depends on the available bandwidth but it is stable for different temporal resolutions. Similarly looking at Table III it can be noticed how the lowest temporal resolution, for both BL and EL, can always be decoded at high quality. For the other temporal resolutions, which in principle should behave in a similar maner, a PSNR degradation is present when the available bandwidth is diminished. Table II describes the behaviour of the system when for a given bandwidth a similar quality is desired at both spatial resolutions (BL and EL). As a consequence, near constant quality is obtained for different frame rates.**

bandwidth	50 Hz	25 Hz	12.5 Hz
BL B1	38.6	38.6	38.8
BL B2	38.6	38.6	38.8
BL B3	35.1	38.6	38.8
BL B4	33.3	35.9	38.8
EL B1	38.5	38.6	39.2
EL B2	36.9	38.2	39.2
EL B3	33.5	35.8	39.2
EL B4	32.3	34.5	39.2

TABLE III
PERFORMANCE SUMMARY FOR CONFIGURATION C3

In **variable bandwidth** experiments, the value of the maximum available bandwidth in the firewall is fixed and equal to the value sufficient to transfer the full codestream (B_F). During the transmission, a disturb traffic is injected into the network for a limited time, where different value of the disturb's bandwidth (B_D) has been tested, in order to overload the traffic in the firewall. For the experiment performed, the priority value $p(q_d)$ set for the disturb queue q_d, satisfies the following condition $p(q_4) < p(q_d) < p(q_3)$. This choice of the priority for the disturb seems to be reasonable, because a higher value could cause higher degradation of the performance, and a lower value is useless because the firewall chose to discard the packets of the disturb traffic in case of congestion. The PSNR over the frames for the base layer and enhancement layer in configurations C1 and C3 is shown in Figure 5, 6, 7, 8. As shown in the figures, the decreasing of the available bandwidth affects only the enhancement layer according to the particular configuration considered. **The quality of the base layer is not influenced by the congestion because of the high priority values assigned to the corresponding portion od the bitstream. This behavior can be better understood looking at Figure 4 and considering the used packet formation and packet filtering methods, described in Section III. The sub-bitstreams S_0^0, S_1^0 and S_2^0 of the base layer have been assigned to the queues q_1, q_2 and q_3 according to the considered configuration ($C1$ or $C3$). These queues have priority $p(q_1)$, $p(q_2)$ and $p(q_3)$ higher than those assigned to the disturb traffic ($p(q_d)$) and consequently packets associated to the base layer are discarded only after the removal of the less important traffic (e.g. disturb and eventually EL information). If the effective bandwidth $B_E = B_F - B_D$ is sufficient to transmit the base layer bitstream is possible to decode it at full resolution also in presence of disturb traffic. Additionally, as shown in Figure 7 the firewall discards the EL packets at high frame-rate (50 Hz) while provides a good quality for the corresponding lower temporal resolution the spatial resolution (see Figure 8).**

V. CONCLUSIONS

In this paper an efficient method for automatically adapting a scalable video stream has ben proposed. Adaptation is performed by opportunely using the functionalities provided by Quality of Service systems. Different configurations have been evaluated, in order to enable flexibility and adaptation with respect to clients preferences concerning the preferred scalability dimension. It has been shown that, thanks to the characteristics of the proposed congestion management method and of the scalable video streams, it is possible to decode a video sequence at a lower spatial resolution or frame rate preserving good quality also in presence of strong bandwidth reduction. The advantage of proposed method compared with other works in literature is the low complexity of the adaptation device, the use of well known mechanisms for providing Quality of Service as Packet Filter, the absence

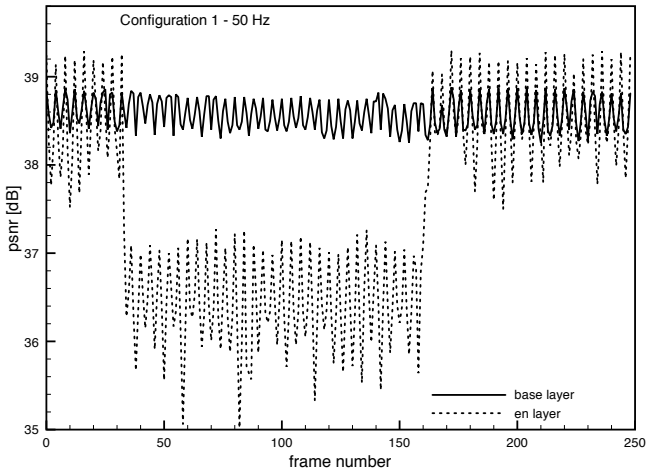

Fig. 5. Configuration C1 - 50 Hz

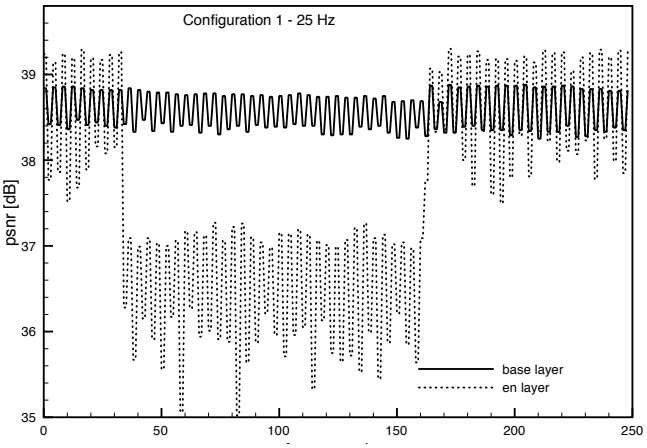

Fig. 6. Configuration C1 - 25 Hz

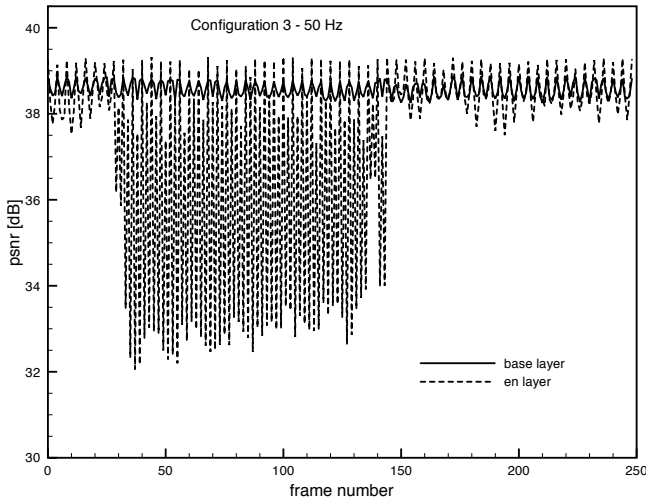

Fig. 7. Configuration C3 - 50 Hz

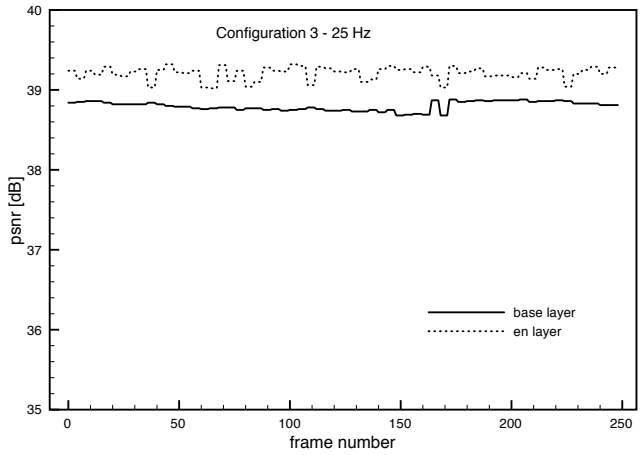

Fig. 8. Configuration C3 - 25 Hz

of feedback channel and no needs of bandwidth estimation algorithms. Future works could address the robustness against the errors on communication channel **and consider the retransmission of lost packets** which are important issues in real video streaming applications.

ACKNOWLEDGMENT

The authors would like to thank Prof. L. Salgarelli for his precious support in the design of the proposed network testbed and application protocols.

REFERENCES

[1] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audio-visual services," Tech. Rep., ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 8 (including SVC extension): Consented in July 2007.

[2] N. Adami, A. Signoroni, and R. Leonardi, "State-of-the-art and trends in scalable video compression with wavelet-based approaches," *CSVT, IEEE Transactions on*, vol. 17, no. 9, pp. 1238–1255, Sept. 2007.

[3] P. B. Ssesanga and M. K. Mandal, "Efficient congestion control for streaming scalable video over unreliable IP networks," in *Proc. of IEEE ICIP 2004*, October 2004.

[4] Dieu Nguyen and Joern Ostermann, "Streaming and congestion control using scalable video coding based on H.264/AVC," *Journal of Zhejiang University - Science A*, vol. 7, no. 5, pp. 749–754, May 2006.

[5] Odd Inge Hillestad, Andrew Perkis, Vasken Genc, Sean Murphy, and John Murphy, "Adaptive H.264/MPEG-4 SVC video over IEEE 802.16 broadband wireless networks," *Packet Video 2007*, pp. 26–35, 12-13 Nov. 2007.

[6] T. Pliakas and G. Kormentzas, "Scalable video streaming traffic delivery in IP/UMTS networking enviroments," *Journal of Multimedia*, vol. 2, no. 9, pp. 37–46, April 2007.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. weiss, "An architecture for differentiated services," Tech. Rep., RFC 2475, December 1998.

[8] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-time system for adaptive video streaming based on SVC," *CSVT, IEEE Transactions on*, vol. 17, no. 9, pp. 1227–1237, Sept. 2007.

[9] L. Lima, F. Manerba, N. Adami, R. Leonardi, and A. Signoroni, "Wavelet based encoding for HD appplications," in *Proc. of the IEEE ICME*, July 2007.

[10] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," Tech. Rep., RFC 3168, September 2001.