Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 218046, 11 pages
doi:10.1155/2008/218046

Research Article

Scalable and Media Aware Adaptive Video Streaming over Wireless Networks

Nicolas Tizon[1, 2] and Béatrice Pesquet-Popescu[1]

[1] *Signal and Image processing Department, TELECOM ParisTech, 46 Rue Barrault, 75634 Paris, France*
[2] *R&D Department, Société Française du Radiotéléphone (SFR), 1 Place Carpeaux, Tour Séquoia, 92915 La Défense, France*

Correspondence should be addressed to Béatrice Pesquet-Popescu, beatrice.pesquet@telecom-paristech.fr

Received 29 September 2007; Accepted 6 May 2008

Recommended by David Bull

This paper proposes an advanced video streaming system based on scalable video coding in order to optimize resource utilization in wireless networks with retransmission mechanisms at radio protocol level. The key component of this system is a packet scheduling algorithm which operates on the different substreams of a main scalable video stream and which is implemented in a so-called media aware network element. The concerned type of transport channel is a dedicated channel subject to parameters (bitrate, loss rate) variations on the long run. Moreover, we propose a combined scalability approach in which common temporal and SNR scalability features can be used jointly with a partitioning of the image into regions of interest. Simulation results show that our approach provides substantial quality gain compared to classical packet transmission methods and they demonstrate how ROI coding combined with SNR scalability allows to improve again the visual quality.

1. INTRODUCTION

Streaming video applications are involved in an increasing number of communication services. The need of interoperability between networks is crucial and media adaptation at the entrance of bottleneck links (e.g., wireless networks) is a key issue. In the last releases of 3G networks [1], jointly with a high speed transport channel, the high speed downlink packet access (HSDPA) technology provides enhanced channel coding features. On the one hand, packet scheduling functionalities of the shared channel located close to the air interface allow to use radio resources more efficiently. On the other hand, error correction mechanisms like hybrid automatic repeat request (HARQ) or forward error correction (FEC) contribute to build an error resilient system. However, these enhancements are designed to be operational through a large collection of services without considering subsequent optimizations. In the best case, a QoS framework would be implemented with network differentiated operating modes to provide a class of services [2]. To guarantee continuous video playout, streaming services are constrained by strictly delay bounds. Usually, guaranteed bitrates (GBR) are negotiated to maintain required bandwidth in case of congestion. Moreover, to guarantee on-time delivery, the retransmission of lost packets must be limited, leading to an over allocation of resources to face the worst cases. The main drawback of a QoS-oriented network is that it requires a guaranteed bitrate per user and thus it does not allow to take advantage of rate variability of encoded videos. In [3], a streaming system is proposed with QoS differentiation in order to optimize experienced quality at client side in the case of degraded channel quality. Assuming that the bandwidth allocated to the user is not large enough with respect to negotiated GBR, this study shows that prioritization of packets following the regions of interest (ROI) can achieve a substantial gain on perceived video quality.

In the scope of packetized media streaming over best-effort networks and more precisely channel adaptive video streaming, [4] proposes a review of recent advances. The closest approach from our works is the well-known rate-distortion optimized packet scheduling method. However, in this technical review, scalable-based solutions are considered as inefficient due to the fact that poor compression performances and wireless networks are not really studied with their most important specificities at radio link layer like radio frame retransmissions. In [5], Chou and Miao have addressed the problem of rate-distortion optimized packet scheduling conducted as an error-cost optimization problem. In their approach, encoded data partitioned into dependent data units, which can be a scalable stream, are represented as a directed acyclic graph. This representation is used with channel error rate measurements as input parameters of a Lagrangian minimization algorithm. This general framework can be adapted in terms of channel model and transmission protocol between the server and the client. For example in [6], the error process of a wireless fading channel is approximated by a first-order Markov process. Then, in order to choose the optimal scheduling policy, the server uses this model combined with video frame-based acknowledgment (ACK/NACK) from the client to compute the expected distortion reduction to be maximized. In [7], a similar approach is proposed considering a measure of congestion instead of the previous distortion. Besides, packet scheduling algorithms can switch between different versions of the streamed video, encoded with different qualities, instead of pruning the previous set of dependent data units. Then, These methods based on rate (congestion)-distortion optimized packet scheduling are in theory likely to provide an optimal solution to media aware scheduling problem. However, without simplification, the Lagrangian optimization is computationally intensive and the channel estimation (delay, capacity) may be more difficult when packets are segmented and retransmitted below application layer (e.g., ARQ at radio link control (RLC) layer). Moreover, in a wireless system, packet scheduling on the shared resource occurs at MAC or RLC layers independently of the application content.

In [7], media bitrate adaptation problem is set as a tradeoff between the current stream pruning and stream switching among a set of videos with different qualities. In order to provide more flexible schemes, the scalable extension of H.264/AVC, namely, scalable video coding (SVC), [8] allows to encode in the same bitstream a wide range of spatiotemporal and quality layers. In [9], a generic wireless multiuser video streaming system uses SVC coding in order to adapt the input stream at the radio link layer as a function of the available bandwidth. Thanks to a media-aware network element (MANE) that assigns priority labels to video packets, in the proposed approach, a drop priority-based (DPB) radio link buffer management strategy [10] is used to keep a finite queue before the bottleneck link. The main drawback of this method is that the efficiency of source bitrate adaptation depends on buffer dimensioning and with this approach, video packets are transmitted without considering their reception deadlines.

In this paper, our approach is to exploit the SVC coding in order to provide a subset of hierarchically organized substreams at the RLC layer entry point and we propose an algorithm to select scalable substreams to be transmitted to RCL layer depending on the channel transmission conditions. The general idea is to perform a fair scheduling between scalable substreams until the deadline of the oldest unsent data units with higher priorities is approaching. When this deadline is expected to be violated, fairness is no longer maintained and packets with lower priorities are delayed in a first time and later dropped if necessary. In order to do this, we propose an algorithm located in a so-called media aware network element (MANE) which performs a bitstream adaptation between RTP and RLC layers based on an estimation of transport channel conditions. This adaptation is made possible thanks to the splitting of the main scalable stream into different substreams. Each of these substreams conveys a specific combination of SNR and/or temporal layers which corresponds to a specific combination of high-level syntax elements. In addition, SVC coding is tuned, leading to a generalized scalability scheme including regions of interest. ROI coding combined with SNR and temporal scalability provides a wide range of possible bitstream partitions that can be judiciously selected in order to improve psychovisual perception.

The paper is organized as follows: in the next section we describe the scalable video coding context and the related standardized tools. In Section 3, we address the problem of ROI definition and propose an efficient way to transmit partitioning information requiring only a slight modification of the compressed bitstream syntax. Then, in Section 4, we present our developed algorithm to perform bitstream adaptation and packet scheduling at the entrance of RLC layer. Finally, simulation results are presented in Section 5 and we conclude in Section 6.

2. SCALABLE VIDEO CODING CONTEXT

2.1. SVC main concepts

To serve different needs of users with different displays connected through different network links by using a single bitstream, a single coded version of the video should provide spatial, temporal, and quality scalability. As a distinctive feature, SVC allows a generation of an H.264/MPEG-4 AVC compliant, that is, backwards-compatible, base layer and one, or several, enhancement layer(s). Each enhancement layer can be turned into an AVC-compliant standalone (and not anymore scalable) bitstream, using built-in SVC tools. The base-layer bitstream corresponds to a minimum quality, frame rate, and resolution (e.g., QCIF video), and the enhancement-layer bitstreams represent the same video at gradually increased quality and/or increased resolution (e.g., CIF) and/or increased frame rate. A mechanism of prediction between the various enhancement layers allows the reuse of textures and motion-vector fields obtained in preceding layers. This layered approach is able to provide spatial scalability but also a coarse-grain SNR scalability. In a CGS bitstream, all layers have the same spatial resolution but lower layers coefficients are encoded with a coarser quantization steps. In order to achieve a finer granularity of quality, a so-called medium grain scalability (MGS), identical in principle to CGS, allows to partition the transform coefficients of a layer into up to 16 MGS layers. This increases the number of packets and the number of extraction points with different bitrates. Coding efficiency of SVC depends on the application requirements but the goal is to achieve a rate-distortion performance that is comparable to nonscalable H.264/MPEG-4 AVC. The design of the scalable

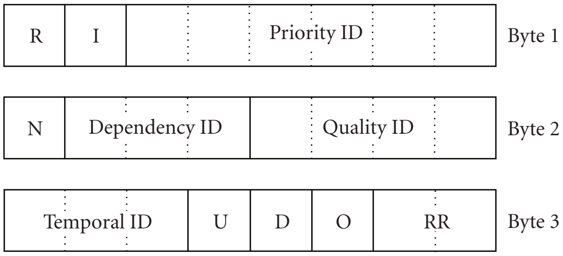


Figure 1: Additional bytes in SVC NAL unit header.

H.264/MPEG4-AVC extension and promising application areas are pointed out in [8].

2.2. Bitstream adaptation

An important feature of the SVC design is that scalability is provided at the bitstream level. Bitstreams for a reduced spatial and/or temporal resolution can be simply obtained by discarding NAL units (or network packets) from a global SVC bitstream that are not required for decoding the target resolution. NAL units of progressive refinement slices can additionally be dropped or truncated in order to further reduce the bitrate and the associated reconstruction quality. In order to assist an MANE (e.g., a network gateway) in bitstream manipulations, the one-byte NAL unit header of H.264/MPEG4-AVC was extended by 3 bytes for SVC NAL units [11]. These additional bytes signalize whether the NAL unit is required for decoding a specific spatiotemporal resolution and quality (or bitrate) as illustrated in Figure 1. The simple priority ID "PRID" indicator is used to infer the global priority identifier of the current NAL unit. A lower value of PRID indicates a higher priority. In oder to provide a finer discrimination between SVC NAL units and to facilitate bitstream parsing, the NALU header allows to assign different priorities inside each scalable domain thanks to the values of temporal id, dependency id, and quality id fields. The reserved bit "R" can be ignored and flag "I" specifies whether the current frame is an instantaneous decoding refresh (IDR) frame. The interlayer prediction flag "N" indicates whether another layer (base layer) may be used for decoding the current layer and "U" bit specifies the reference base pictures utility (used or not) during the interprediction process. Then, discardable flag "D" signals that the content of the information in current NAL units is not used as a reference for the higher level of dependency id. At last, "O" gets involved with the decoded picture output process and "RR" are reserved bits for future extension.

2.3. Flexible macroblock ordering (FMO)

H.264/AVC provides a syntactical tool: FMO, which allows partitioning video frames into slice groups. Seven different modes, corresponding to seven different ordering methods, exist, allowing to group macroblocks inside slice groups. For each frame of a video sequence, it is possible to transmit a set of information called picture parameter set (PPS), in which the parameter slice_group_map_type specifies the FMO mode of the corresponding frame. According to this parameter, it is also possible to transmit additional information to define the mapping between macroblocks and slice groups. Each slice group corresponds to a network abstraction layer (NAL) unit that will be further used as RTP payload. This mapping will assign each macroblock to a slice group which gives a partitioning (up to eight partitions) of the image. There exist six mapping methods for an H.264 bitstream. In this study, we use the mode 6, called *explicit MB*, to slice group mapping, where each macroblock is associated to a slice group index in the range [0..7]. The relation of macroblock to slice group map amounts to finding a relevant partitioning of an image. Evaluation of partitioning relevance strongly depends on the application and often leads to subjective metrics.

3. ROI EXTRACTION AND CODING

3.1. ROI definition

In image processing, detection of ROIs is often conducted as a segmentation problem if no other assumptions are formulated about the application context and postprocessing operations that will be applied on the signal.

Concerning the application context of our study, we formulate the basic assumption that in the majority of cases, a video signal represents moving objects in front of almost static background. In other words, we make the assumption that the camera is fixed or that it is moving slower than the objects inside the scene. With this model, moving objects represent the ROI and FMO is restricted to 2 slice groups. According to this definition, motion estimation (ME) that occurs during the encoding process delivers relevant information through motion vector values to detect ROIs. In H.264, the finest spatial granularity to perform ME is a 4×4 block of pixels while FMO acts at macroblock level. In our simulations, to detect ROIs we compute the median value of motion vectors in a macroblock. Each vector is weighted by the size of the block it applies to. Next, the macroblock is mapped to ROI if this median value is higher than a threshold value, as depicted in Figure 2.

3.2. Mapping information coding

The H.264/AVC standard defines a macroblock coding mode applied when no additional motion and residual information need to be transmitted in the bitstream. This mode, called SKIP mode, occurs when the macroblock can be decoded using information from neighbor macroblocks (in the current frame and in the previous frame). In this case, no information concerning the macroblock will be carried by the bitstream. A syntax element, mb_skip_run, specifies the number of consecutive skipped macroblocks before reaching a nonskipped macroblock.

In our macroblock to slice group assignment method, a skipped macroblock belongs to slice group 2 (lowest priority). In fact, this assignment is not really effective because no data will be transmitted for this macroblock. The set of skipped macroblocks in a frame can be seen as

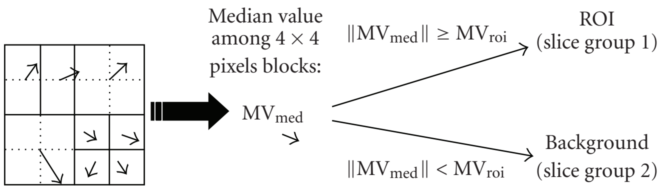

Median value
among 4 × 4
pixels blocks:
MV_{med}
$\|\text{MV}_{\text{med}}\| \geq \text{MV}_{\text{roi}}$
ROI
(slice group 1)
Background
(slice group 2)
$\|\text{MV}_{\text{med}}\| < \text{MV}_{\text{roi}}$

Figure 2: Macroblock classification according to the motion vector value.

a third slice group (with null size). In a general manner, mb_skip_run syntax element can be considered as a signaling element to indicate a set of macroblocks belonging to a slice group (index incremented by one) as depicted in Figure 3. If slice groups with higher indices are lost, the decoding process will still be maintained with lower indexed slice groups. This method generalizes the use of mb_skip_run syntax element and allows to code macroblock to slice group mapping without sending explicit mapping with the frame header, picture parameter set (PPS). Indeed, mb_skip_run is included into the H.264 bitstream syntax, coded with an efficient entropy coding method. This coding method does not introduce new syntax elements but as the meaning of mb_skip_run is modified (in the case of more than one slice group), the provided bitstream is no longer semantically compliant with regard to the H.264 reference decoder. At the client side, each slice group is received independently through a specific RTP packet. To be able to perform bitrate adaptation, the MANE needs to know the relative importance of each slice group without parsing the scalable bitstream. In the next section, we propose a method using SVC high-level syntax to label each slice group with the appropriate priority.

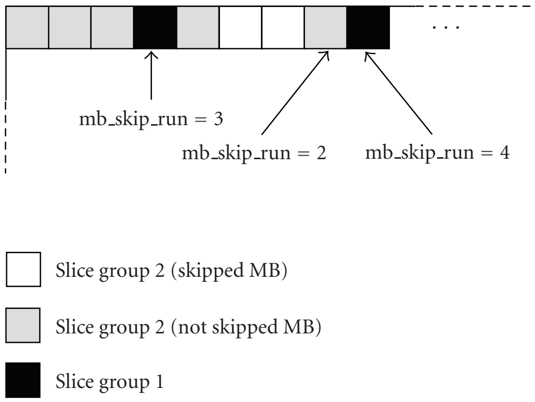


Figure 3: An example of macroblock to slice group map coded via mb_skip_run syntax.

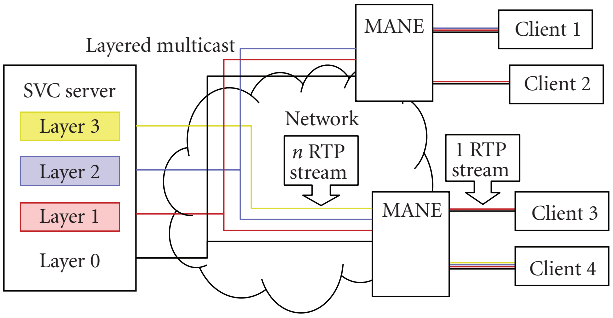


Figure 4: Scalable bitstream adaptation in the MANE based on users conditions.

4. ADAPTATION AND PACKET SCHEDULING

In the sequel, we will restrict scalability abilities of SVC to the temporal layering with the well-known hierarchical B pictures structure, and to SNR scalability with MGS slices coding. In fact, we assume that spatial scalability-based adaption has already occurred when reaching the bottleneck link. Thanks to the additional bytes in SVC NAL unit headers, the network is able to select a subset of layers from the main scalable bitstream. Moreover, in the previous section, we described a coding method in order to provide a data differentiation at image content or ROI level. In this section, we propose a packetization method that combines SVC native scalability modes and the underlying scalability provided by ROI partitioning with FMO.

4.1. Packetization and stream-based priority assignment

In this study, we adopt an adaptation framework in which the streaming server sends scalable layers as multiple RTP substreams that are combined into a single RTP stream, adapted to each client transmission condition in the MANE [11] as described in Figure 4. With SVC extended NAL unit header, 6 bits indicate simple priority ID. Then, we use this field to specify the importance of a slice group (SG)determined upon ROI definition in Section 3, and the third byte specifies NAL unit assignment to temporal and quality levels. The higher the importance of the SG, the lower the value of the priority ID. Inside a scalability domain (temporal or SNR), packet prioritization derivation is straightforward according to the appropriate level ID in the third byte of the NAL unit header. For example, temporal level 0 corresponds to the highest priority among temporal level IDs. In the case of combined scalability, priority labeling is more complicated and usually dependent on the application. For example, watching a scene with high motion activities may require high temporal resolution rather than high-quality definition because human vision

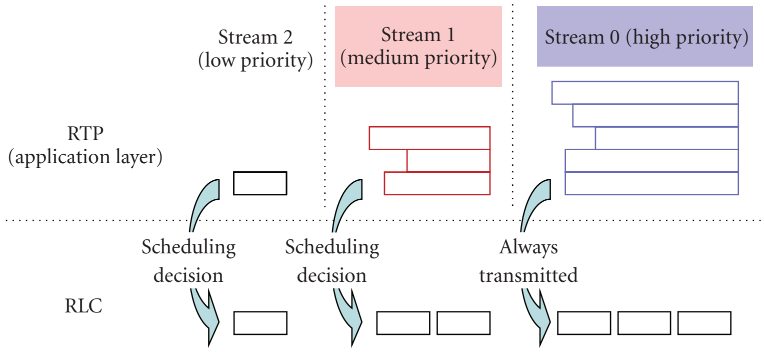


Figure 5: Scalable scheduling principle with three substreams.

does not have time to focus on moving objects details but privileges display fluidity. Then in this example, if the receiver undergoes bandwidth restrictions, it would be more judicious for the MANE to transmit packets with highest-temporal level and lowest-quality level before packets with lowest-temporal level and highest-quality level. On the contrary, with static video contents, the MANE will favor quality rather than temporal resolution. Finally, adding ROI scalability makes possible to deliver different combinations of quality and temporal scalabilities between regions of the same video frame. In Section 5.2, from simulation results, we discuss how to find the best combination of scalable streams to optimize perceived video quality in function of the considered application and media content. Next, we assume that MANE input data is composed of N substreams indexed from higher to lower importance or priority. Each stream can be a simple scalable layer with a given temporal or quality level or a more sophisticated combination of layers as explained before.

4.2. Packet scheduling for SVC bitstream

In the remaining of this study, we consider that the MANE sees RLC layer as the bottleneck link and performs packet scheduling from IP layer to RLC layer. In the case of a 3G network, the MANE is most probably between the radio network controller (RNC) and the gateway GPRS support node (GGSN) and we neglect transmission delay variations between the server and the MANE. Then, each RTP packet whose payload is an NAL unit is received by the MANE at $t = \mathrm{TS} + t_0$, where TS is the sampling instant of the data and t_0 the constant delay between the MANE and the server. Next, to simplify this we put $t_0 = 0$ knowing that this time only impacts the initial playout delay. Moreover, inside each scalable stream, packets are received in their decoding order which can be different from the sampling order due to the hierarchical B pictures structure. Hence, the head-of-line (HOL) data unit of a stream queue is different from the minimum sampling instant of queued packets: $\mathrm{TS}_{\min}$.

Input RTP streams are processed successively. When scheduling RTP packet, the algorithm evaluates the transmission queues of the most important streams and, according to network state, the current packet will be delayed or sent to RLC layer. All streams are next transmitted over the same wireless transport channel and when an RTP packet reaches RLC layer, all necessary time slots are used to send the whole packet. Therefore, the general principle of the algorithm is to allow sending a packet only if packet queues with higher priorities are not congested and if expectable bandwidth is sufficient to transmit the packet before its deadline.

In order to detail the algorithm, we are considering that the bitstream is transmitted through a set of L streams and the scheduler is up to send the HOL packets of the kth stream at time t. Let us denote $\mathrm{TS}_k(t)$ as the sampling instant of this packet, $S_k(t)$ as its size, $d_k(t)$ as its transmission time, and $D_{\max}$ as the maximum end-to-end delay for all packets of the streaming session. Scheduling opportunities for this packet will be inspected only if its reception deadline is not past and if a significant ratio ϵ of the maximum end-to-end delay is still available before reaching this deadline as follows:

$$t - d_k(t) < (1 - \epsilon)D_{\max}. \tag{1}$$

If this condition is not verified, the packet is discarded. Otherwise, to perform the transfer of the packet to the RLC layer (see Figure 5), that is to send or to delay the packet, packet queue of the lth stream, where $l = k + 1, \dots, L$, is considered as a single packet with time stamp $\mathrm{TS}_{\min\, l}(t)$. Then, we define $D_l(t)$, the transmission time for this aggregated packet and we fix $t' = t + d_k(t)$. The second condition which must be verified before sending the packet is

$$t' - \mathrm{TS}_{\min}(t') < (1 - \epsilon)D_{\max} - D_l(t'). \tag{2}$$

With this condition, the algorithm assures that the network is able to send the packet without causing future packets loss from streams with higher priorities. If this condition is not verified, the packet is put on the top of the kth queue and the algorithm examines the $(k + 1)$th stream.

Moreover, packet dependency can occur between packets from the same stream, in the case of a combined scalability-based stream definition, or between packets from different streams. Therefore, in order to provide an efficient transmission of scalable layers, the algorithm delays packet delivering until all packets from lower layers which are necessary to decode the current packet are transmitted.

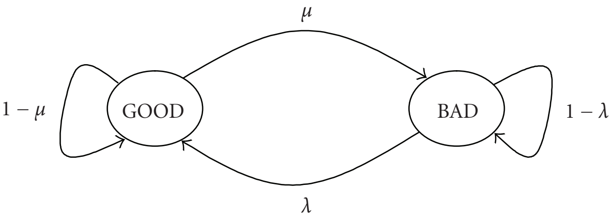


Figure 6: 2-state Markov channel model.

Given these two conditions, the main difficulty is to evaluate the 5 variables that are defined as a function of time and need to be calculated in the future. Firstly, let us note that the RTP streams are processed sequentially and thus between t' and t instants, the sizes of the others packet queues ($l \neq k$) will increase and their oldest time stamp will remain unchanged. So, we can write $\mathrm{TS}_{\min l}(t') = \mathrm{TS}_{\min l}(t)$. Next, we calculate the $d_k(t)$ value which amounts to perform a channel delay estimation. In order to do this, we are considering that the channel state is governed by a 2-state Markov chain. Therefore, thanks to this model, the network is simply considered to be in "GOOD" or "BAD" state as depicted in Figure 6. The transition probabilities, λ and μ, are considered as function of time variables in order to take into account possible channel state evolutions. In order to complete the network model, we define tti and rfs as the variables that represent the transmission time interval (TTI) and the radio frame size (RFS) constant values. A radio frame is actually an RLC protocol data unit (RLC-PDU). Before reaching the RLC layer, an RTP packet is segmented into radio frames and an RLC-PDU is sent every TTI. In fact, if tti and rfs are constant, we implicitly assume that we are dealing with a dedicated channel with constant bitrate. Nevertheless, in our simulations tti value can be modified in order to simulate a radio resource management-based decision of the network which can perform bandwidth allocation on the long run. Additionally, channel state transitions occur every TTI, so we can write the current time as a discrete variable: $t = n \times \mathrm{tti}$. Finally, the transition probabilities, λ and μ are dynamically calculated every TTI performing a state transition count over a sliding time window $T = N \times \mathrm{tti}$.

Let us define the random process $\mathrm{TT}(t)$ (transmission time) which represents the time spent by the network (including RLC retransmissions) to send a radio frame whose first sending instant is t. Actually, TT is a discrete time process and we have $\mathrm{TT}(t) = \mathrm{TT}(n \times \mathrm{tti}) = \mathrm{TT}(n)$. As rfs is constant, $I = \lceil S_k(t)/\mathrm{rfs} \rceil$ is the number of RLC-PDUs involved in the transmission of the current HOL RTP packet of the kth stream. With these notations, let us denote $\mathrm{tti} \times \{n_0, n_1, \dots, n_I\}$ with $n_0 = n$, the sequence of sending instants corresponding to the first transmission of the related RLC-PDUs. So, we can express the overall transmission time of the RTP packet as follows:

$$d_k(t) = \sum_{i=n}^{I} \mathrm{TT}(n_i). \tag{3}$$

In order to evaluate $\mathrm{TT}(n)$, we use past observations thanks to radio link control acknowledged mode (RLC AM) error feedback information sent by the receiver. This information is received by the transmitter after a certain feedback delay, $r \times \mathrm{tti}$, and r is a fixed integer value which depends on RLC configuration. Moreover, we estimate the average value of TT over the RTP packet transmission duration by the average value of $\mathrm{TT}(n-r)$. In other words, we consider that the average channel state is constant through RTP packet transmission duration. So, we have the following estimated parameter:

$$\widehat{d_k}(t) = E\{\mathrm{TT}(n-r)\} \times \left\lceil \frac{S_k(t)}{\mathrm{rfs}} \right\rceil. \tag{4}$$

When the channel is in "GOOD" state, $\mathrm{TT}(n) = \mathrm{tti}$ and when the channel state is "BAD," we approximate $\mathrm{TT}(n)$ by the average TT value of previously retransmitted RLC-PDU (one time at least) over the previously defined time window T. Let us denote $\mathrm{tt}_{\mathrm{bad}}$ by this average value. We have

$$\mathrm{tt}_{\mathrm{bad}}(n) = \frac{\sum_{i=n-N,\mathrm{TT}(i)>\mathrm{tti}}^{n} \mathrm{TT}(i)}{\sum_{i=n-N,\mathrm{TT}(i)>\mathrm{tti}}^{n} i}. \tag{5}$$

Then, the mean value of $\mathrm{TT}(n)$ can be expressed as

$$\begin{aligned} E\{\mathrm{TT}(n)\} &= \mathrm{tt}_{\mathrm{bad}}(n) \times P(\mathrm{TT}(n) = \mathrm{tt}_{\mathrm{bad}}(n) \mid \mathrm{TT}(n-1)) \\ &\quad + \mathrm{tti} \times P(\mathrm{TT}(n) = \mathrm{tti} \mid \mathrm{TT}(n-1)). \end{aligned} \tag{6}$$

In order to provide the estimation of $D_l(t')$ involved in the scheduling condition defined by (2), we define $S_l(t')$ as the size of the aggregated RTP packets of the lth stream. In addition, let us define $r_l(t)$ as the source bitrate of this lth stream calculated over the previously defined time window T. Thus, in the sequel, we will use the following approximation:

$$S_l(t') = S_l(t) + r_l(t) \times d_k(t). \tag{7}$$

Next, we estimate the transmission time of this aggregated packet assuming that the previous network estimation (6) will be usable over the time interval $[t, D_l(t')]$. Therefore, similar to (4), we can write

$$\widehat{D_l}(t') = E\{\mathrm{TT}(n-r)\} \times \left\lceil \frac{S_l(t')}{\mathrm{rfs}} \right\rceil. \tag{8}$$

5. EXPERIMENTAL RESULTS

5.1. Simulation tools

To evaluate the efficiency of the proposed approach, some experiments have been conducted using a network simulator provided by the 3GPP video ad hoc group [12].

This software is an offline simulator for an RTP streaming session over 3GPP networks (GPRS, EDGE, and UMTS). Packet errors are simulated using error masks generated from link-level simulations at various bearer rates and block error rate (BLER) values. Moreover, this simulator offers

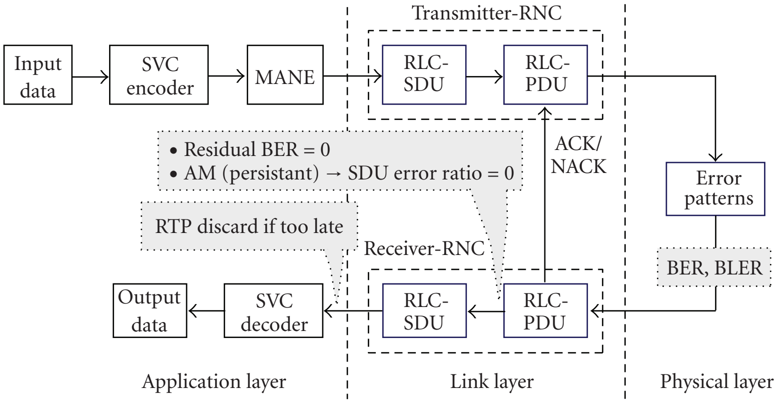


Figure 7: Simulation model.

the possibility to simulate time events (delays) using the time stamp field of the RTP header. The provided network parameters are nearly constant throughout the session. For simulating radio channel conditions two possible input interfaces are provided: bit-error patterns in binary format, as well as RLC-PDU losses in ASCII format. Error masks are used to inject errors at the physical layer. If the RLC-PDU is corrupted or lost, it is discarded (i.e., not given to the receiver/video decoder) or retransmitted if the RLC protocol is in acknowledged mode (AM). The available bit-error patterns determine the bitrates and error ratios that can be simulated. Two bit-error patterns with binary format are used in the experiment. These patterns are characterized by a relatively high BER (BER $= 9.3e-3$ and BER $= 2.9e-3$) and are suited to be used in streaming applications, where RLC layer retransmissions can correct many of the frame losses. All bearers are configured with persistent mode for RLC retransmissions and their bitrates are adjusted using the RLC block size and the TTI parameters provided by the simulator. An erroneous RLC packet is retransmitted until it is correctly received. If the maximum transfer delay due to retransmission is reached, the corresponding RTP packet is discarded. Therefore, the residual BER is always null, only missing RTP packets may occur, as depicted in Figure 7. In order to validate a strategy, results must be provided over a large set of simulations varying the error mask statistics. Therefore, for a simulation, the error pattern is read with an offset varying from 0 at the first run and incremented by 1 for each run and finally the results are evaluated over a set of 64 runs, as recommended in [13].

In addition, the RTP packetization modality is single network abstraction layer (NAL) unit mode (one NAL unit/RTP payload), the division of original stream into many RTP substreams leads to an increase of the number of RTP headers. To limit the multiplications of header information, the interleaved RTP packetization mode allows multitime aggregation packets (NAL units with different time stamps) in the same RTP payload. In our case, we make the assumption that RoHC mechanisms provide RTP/UDP/IP header compression from 40 to 4 bytes in average, which is negligible compared to RTP packet sizes, and we still packetize one NAL unit per RTP payload.

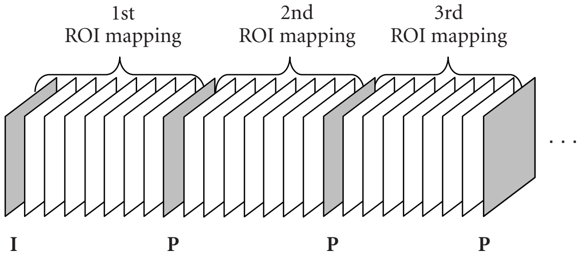


Figure 8: Prediction mode structure and ROI coding scheme.

5.2. *Simulation results*

To evaluate the proposed approach, we present simulation results obtained with the following three test sequences.

(i) *Mother and daughter (*15 *fps, QCIF,* 450 *frames)*: fixed background with slow moving objects.

(ii) *Paris (*15 *fps, QCIF,* 533 *frames)*: fixed background with fairly bustling objects.

(iii) *Stefan (*15 *fps, QCIF,* 450 *frames)*: moving background with bustling objects (this sequence is actually a concatenation of 3 sequences of 150 frames in order to obtain a significant simulation duration).

The prediction mode scheme for frame sequencing is the classical IPPP... pattern in order to evaluate the robustness of the proposed approach and its capacity to limit distortion due to error propagation. The ROI is periodically redefined after each P frame, as illustrated in Figure 8. Concerning the common scalability features, SVC bitstreams are encoded with a group of pictures (GOP) size of 8 (4 temporal levels) and one MGS refinement layer which corresponds to a quantization factor difference of 6 from the base to the refinement quality layer. Then, each RTP packet can be either the quality base layer of a slice group or its enhanced quality layer at a given temporal level. The constants defined in Section 4.2 are used with the following values: $D_{\max} = 1.5$ s, rfs $= 80$ bytes, tti $= 10$ ms by default, and $r = 2$. Finally, ϵ is fixed to 25% after a progressive decrease (65% at the beginning) during the first seconds of the transmission.

Table 1: Performance comparison between H.264 (one RTP stream) and SVC (2 RTP streams: base layer and SNR refinement).

	Mother and daughter	Paris	Stefan
H.264/AVC	27.58 dB	26.43 dB	18.6 dB
SVC	34.2 dB	29.74 dB	27.73 dB

In fact, at the beginning of the transmission each RTP queue is empty and the scheduling algorithm could cause network congestion as it would transmit all the refinement layers without discarding before reaching the stationary state. Thus, the progressive decrease of ϵ allows us to limit this undesirable behaviour during the transitional period.

5.2.1. Adaptation capabilities

Table 1 presents simulation results obtained by configuring each channel with a BLER of 10.8% (BER $= 9.3e - 3$). For "Paris" and "mother and daughter" sequences, the bitrate provided at RLC layer is 64 Kbps and then by removing 4 bytes/packet of RLC header information, the maximum bitrate available at application level (above RTP layer) is approximately 60.8 Kbps. Moreover, for these two sequences, in the case of H.264 coding, a bitrate constrained algorithm at source coding was used in order to match an average target bitrate of 60 Kbps. Concerning "Stefan" sequence, the motion activity is much more significant and to obtain an acceptable quality, we encode the video with an average target bitrate of 120 Kbps. Thus, the corresponding channel used to transmit this sequence is configured with a TTI of 5 ms, leading to a maximum available bitrate of 121.6 Kbps. In the case of SVC coding, the video is encoded without bitrate control algorithm and streamed through two RTP streams. The first one corresponds to the quality base layer transmitted with the highest priority and the second corresponds to the enhanced quality layer transmitted with lower priority. For this first set of simulations, no other scalability features, temporal or SNR, are used to differentiate the RTP streams. PSNR values are measured over the whole sequence and the proposed method allows to gain from 3.3 dB to 9.13 dB. The capacity of our method to better face error bursts is particularly visible in Figure 9. At the beginning of the session, up to $t = 150$ ms, the two coding methods provide a good quality. With SVC coding, the quality is a little bit lower, but more constant, due to the progressive decrease of ϵ previously described. At the end of this starting period, an error burst occurs and the quality with the nonscalable coding dramatically decreases. However, as the content of the sequence does not vary a lot from one image to another, the decoder is able to maintain an acceptable quality. Next, at around $t = 350$ ms, another error burst occurs and also the content of the video is quite more animated. Then, with H.264 coding, the decoder is no longer able to provide an acceptable quality, whereas with SVC we observe only a limited quality decrease. So, our proposed method better faces error bursts, adapting the transmitted bitrate given the estimated capacity of the transport channel.

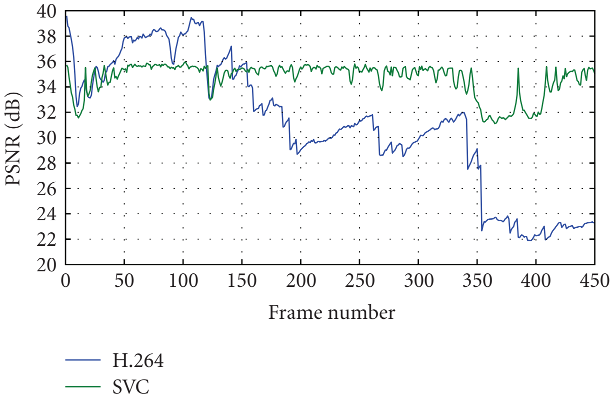


Figure 9: Frame PSNR evolution for "mother and daughter" test sequence (BLER = 3.3%, tti = 10 milliseconds).

Moreover, our algorithm provides an adaptation mechanism that avoids fatal packet congestion when the source bitrate increases. This second aspect is particularly interesting in the case of video which represents bustling objects with a lot of camera effects (zoom, traveling, etc.) like "Stefan" sequence. In this sequence, as illustrated in Figure 10, the bitrate (at MANE input) hugely fluctuates due to the high motion activity. On the one hand, our algorithm allows bitrate variations and achieves a good quality when the available channel bitrate is large enough. On the other hand, when the required bitrate overcomes the channel capacity, the quality refinement layer is discarded, leading to a limited quality decrease ($t = 8$ s). Next, during a short period, even if the source bitrate decreases under the channel capacity, this enhanced quality layer is still discarded. This localized congestion phenomenon is due to the response time of the algorithm. After this transitory period, the full quality is achieved again.

5.2.2. Adaptation capabilities and bandwidth allocation

In this section, the simulations are conducted in order to study the combined effects of channel errors and bandwidth decrease. Indeed, the implementation of a dedicated channel with a purely constant bitrate is not really efficient in terms of radio resource utilization between all users. Then, a more advanced resource allocation strategy would decrease the available bandwidth of the user when his conditions become too bad, in order to better serve other users with better experienced conditions. This allocation strategy, which aims at maximizing the overall network throughput or the sum of the data rates that are delivered to all users in the network,

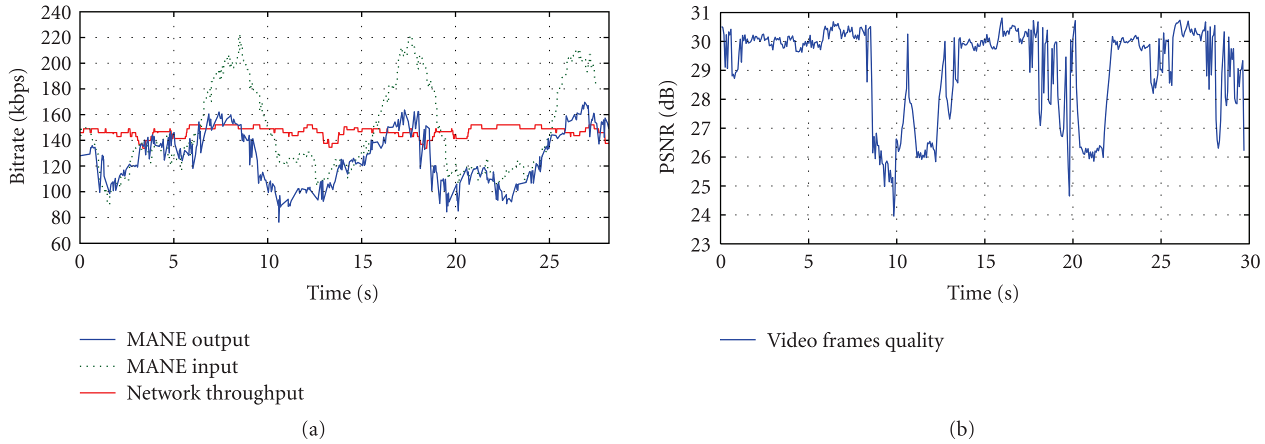


Figure 10: Bitrate adaptation with highly variable source bitrate (Stefan, BLER = 3.3%, tti = 4 milliseconds).

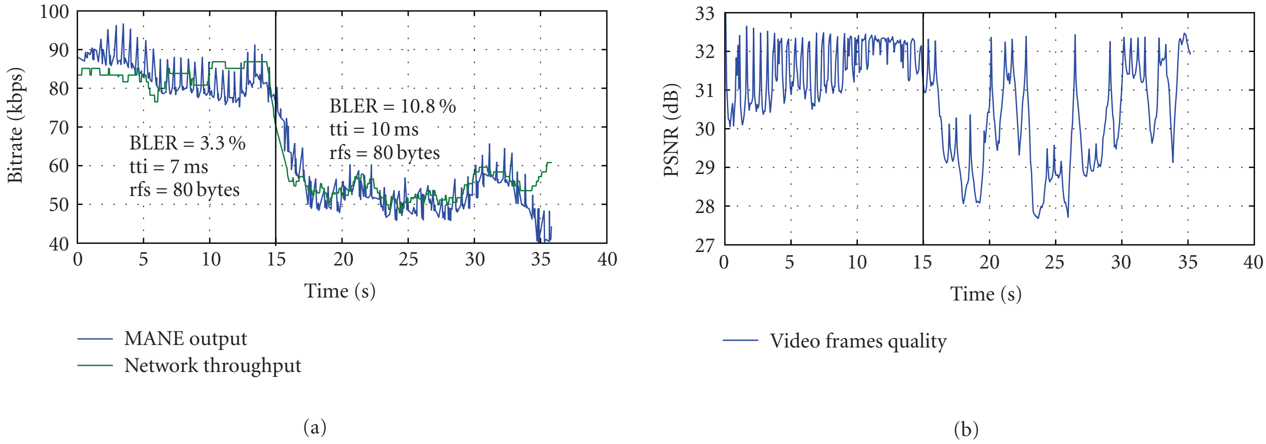


Figure 11: Bitrate adaptation with two RTP streams: quality base layer and SNR refinement layer (Paris).

corresponds to an ideal functioning mode of the system but it is not really compatible with a QoS-based approach.

Actually, with a classical video streaming system, it is not really conceivable to adjust the initially allocated channel bitrate without sending feedbacks to the application server, which is generally the only entity able to adapt the streamed bitrate. Moreover, when these feedbacks are implemented, adaptation capabilities of the server are often quite limited in the case of a nonscalable codec: transcoding, bitstream switching, and so forth. Then in our proposed framework, with the MANE located close to the wireless interface, it is possible to limit the bitrate at the entrance of the RLC layer if a resource management decision (e.g., bandwidth decrease) has been reported. In this case, as illustrated in Figure 11, our adaptive packet transmission method allows to maintain a good level of quality while facing a high error rate and a channel capacity decrease. In the presented simulation results, after 15 ms a quality decrease of 1.7 dB in average and 4 dB in the worst case is measured, whereas the available user bitrate is reduced by more than 30% because of the combined effects of allocated bandwidth decrease (30%) and BLER increase.

5.2.3. *Scalability and ROI combined approach*

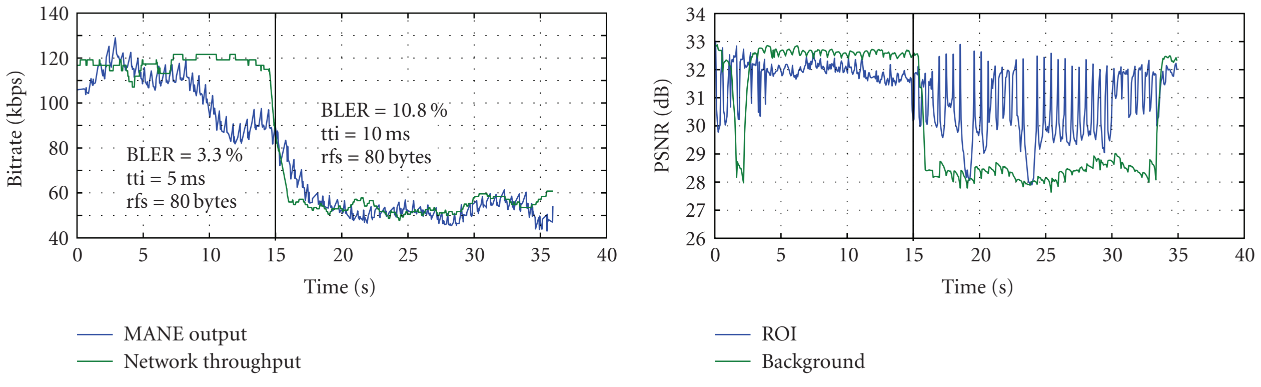
In this section, we evaluate the contribution, in terms of psychovisual perception of the ROI-based differentiation combined with SVC intrinsic scalability features. In order to do this, the simulator is configurated like in the previous section with a bandwidth decrease at the 15th second. At the source coding, an ROI partitioning is performed as described in Section 3 and a quality refinement layer is used, leading to a subset of three RTP streams:

(i) the quality base layer of the whole image (high priority),

(ii) the refinement layer of the ROI slice group (medium priority),

(iii) the refinement layer of the background (low priority).

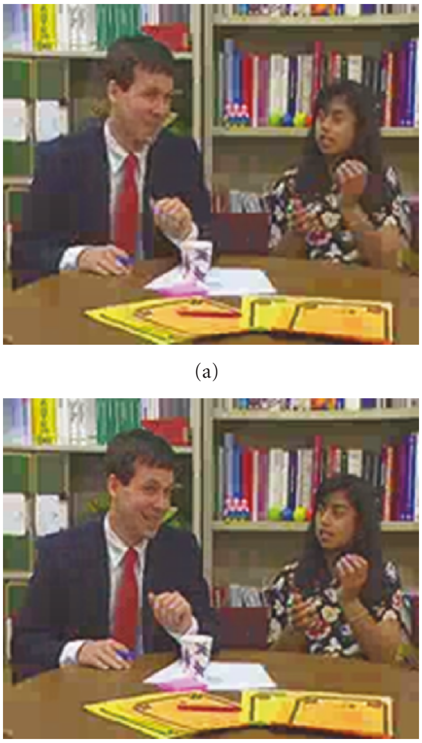
In Figure 12, we can observe the quality variation per image region through the session. So, at the beginning, when channel conditions are favorable, the two regions are transmitted with quite similar quality levels and we reach the



(a)

(b)

Figure 12: Bitrate adaptation with 3 RTP streams: quality base layer, SNR refinement for ROI, and SNR refinement for background (“Paris” sequence).



(a)

(b)

Figure 13: Visual comparison at $t = 17.5$ seconds (Paris, BLER = 10.8%, tti = 10 milliseconds). (a) No ROI differentiation, (b) ROI and SNR combined scalability (“Paris” sequence).

maximum achievable quality between $t = 8$ s and $t = 15$ s. Next, when the channel error rate increases, the available bandwidth is reduced by 50% and we clearly observe two distinct behaviors, following the concerned image region. The quality of the background deeply falls (4 dB in average) and remains almost constant. On the contrary, the quality of the ROI becomes more variable but the PSNR decrease is contained (less than 2 dB in average).

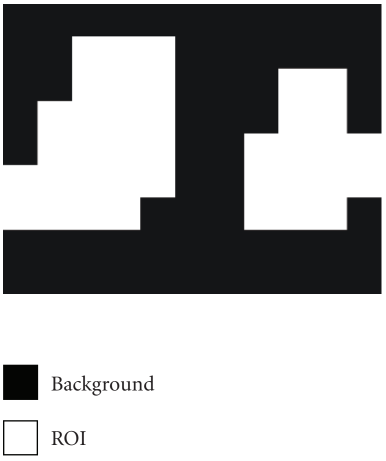


Figure 14: Slice group mapping (“Paris” sequence, t=17.5 seconds).

In order to illustrate these PSNR variations, a visual comparison is provided in Figure 13. In fact, the main interest of this method is that quality variations of the background are not really perceptible. So, in order to better illustrate the gain of this method in terms of visual perception, we compared the displayed image in two cases: with and without ROI differentiation, with the channel conditions evolution of the previous simulation. Moreover, Figure 14 represents the slice group partitioning between ROI and background for the concerned video frame. Thus, we can observe that figures and human expressions of the personages are provided with better quality when the ROI-based differentiation is applied. Moreover, some coding artefacts are less perceptible around the arm of the woman.

In addition, our proposed algorithm is designed in order to allow more complex layers combinations with temporal scalability. In our simulations, the utilization of the temporal scalability did not provide a substantial additional perceived quality gain. In theory, it would be possible to perform more sophisticated differentiation between images regions. For example, we can imagine a configuration where the stream with the highest priority contains the following layers:

(i) quality base layer of the ROI with the full temporal resolution,

(ii) SNR refinement layer of the ROI with a reduced temporal resolution,

(iii) quality base layer of the background with a reduced temporal resolution.

In fact, the bitrate of a quality base layer, and more particularly for the background, is often low. Hence, the bitrate saved by removing from the temporal resolution of the background is not high enough to compensate for the additional SNR refinement layer of the ROI. Therefore, the global bitrate of this RTP stream would be high and it would not be surely transmitted, leading to degraded performances.

6. CONCLUSION

This study proposes a complete framework for scalable and media aware adaptive video streaming over wireless networks. At the source coding, we developed an efficient coding method to detect ROIs and transmit ROI mapping information. Next, using the SVC high-level syntax, we proposed to combine ROI partitioning with common scalability features. In order to multiplex scalable layers, we adopted the MANE approach. In our system, the MANE is close to the wireless interface and it manages RTP packets transmission to the RLC layer following priority rules. In order to do this, a bitrate adaptation algorithm performs packet scheduling based on a channel state estimation. This algorithm considers the delay at RLC layer and packet deadlines in order to maximize the video quality avoiding network congestion. Our simulations show that the proposed method outperforms classical nonscalable streaming approaches and the adaptation capabilities can be used to optimize the resource utilization. Finally, the ROI approach combined with SNR scalability allows to improve again the visual quality. Future work will aim at generalizing this study in the case of a shared wireless transport channel.

REFERENCES

[1] 3GPP, “High Speed Downlink Packet Access (HSDPA),” *3GPP TS 25.308 V7.3.0*, June 2007.

[2] M. Etoh and T. Yoshimura, “Advances in wireless video delivery,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 111–122, 2005.

[3] N. Tizon and B. Pesquet, “Content based QoS differentiation for video streaming in a wireless environment,” in *Proceedings of 15th European Signal Processing Conference (EUSIPCO ’07)*, Poznan, Poland, September 2007.

[4] B. Girod, M. Kalman, Y. J. Liang, and R. Zhang, “Advances in channel-adaptive video streaming,” *Wireless Communications and Mobile Computing*, vol. 2, no. 6, pp. 573–584, 2002.

[5] P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390–404, 2006.

[6] D. Tian, X. Li, G. Al-Regib, Y. Altunbasak, and J. R. Jackson, “Optimal packet scheduling for wireless video streaming with error-prone feedback,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’04)*, vol. 2, pp. 1287–1292, Atlanta, Ga, USA, March 2004.

[7] E. Setton, Z. Xiaoqing, and B. Girod, “Congestion-optimized scheduling of video over wireless ad hoc networks,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ’05)*, vol. 4, pp. 3531–3534, Kobe, Japan, May 2005.

[8] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable H.264/MPEG4-AVC extension,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP ’06)*, pp. 161–164, Atlanta, Ga, USA, October 2006.

[9] G. Liebl, T. Schierl, T. Wiegand, and T. Stockhammer, “Advanced wireless multiuser video streaming using the scalable video coding extensions of H.264/MPEG4-AVC,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ’06)*, pp. 625–628, Toronto, Canada, July 2006.

[10] G. Liebl, H. Jenkac, T. Stockhammer, and C. Buchner, “Radio link buffer management and scheduling for wireless video streaming,” *Telecommunication Systems*, vol. 30, no. 1–3, pp. 255–277, 2005.

[11] S. Wenger, Y.-K. Wang, and T. Schierl, “RTP payload format for SVC video,” *draft, Internet Engineering Task Force (IETF)*, February 2008.

[12] 3GPP and Siemens, “Software simulator for MBMS streaming over UTRAN and GERAN,” *document for proposal, TSG System Aspects Working Group4#36, Tdoc S4-050560*, September 2005.

[13] 3GPP and BenQmobile, “Coponents for TR on video minimum performance requirements,” *document for decision, TSG System Aspects Working Group4#39, Tdoc S4-060265*, May 2006.