A MAC centric Cross Layer approach for H.264 video streaming over HSDPA

Salim Benayoune, Nadjib Achir, Khaled Boussetta and Ken Chen
L2TI, Institut Galilée, University of Paris 13
99 Av.J-B Clement, F-93430 Villetaneuse, France.
Email: firstname.lastname@univ-paris13.fr

Abstract— **This paper proposes a MAC centric cross-layer approach for H.264 video streaming over HSDPA wireless links. Our solution is based on minimal interactions between the RLC (Radio Link Control layer) and the application layer (i.e. H.264 encoder) in order to give video packets different importance values according to the semantic of their contents. The performance of our proposed solution is evaluated through NS-2/EURANE simulations. The results show the gains achieved by our solution in terms of perceived video quality.**

Index Terms— **Cross-layer, H.264/AVC, Wireless, UMTS/HSDPA, Content aware ARQ.**

I. Introduction

The introduction of the HSDPA [1] (High Speed Downlink Packet Access) architecture has allowed packet-based video streaming to mobile devices to become a reality, leading video streaming applications to gain more and more interest among users. Unfortunately, the rapid varying behavior of the wireless channels may cause severe degradations in term of perceived video quality. In this regard, the optimization and adaptation of streaming strategies to wireless networks is becoming a challenging task [2].

Various mechanisms have been developed to overcome these problems, among which we can cite feedback control [3], joint source-rate adaptation [4], [5], and error control [2]. Many of these approaches require end systems to be aware of the current state and of the end-to-end feedback. However, a possibly far away video source cannot react quickly enough to the varying conditions of a wireless link. Thus, the need for good video quality implies a careful design of error control techniques as well as a deep understanding of the underlying mechanisms. For this reason, both new video standards, such as H.264 [6], and new wireless networks, such as HSDPA, propose new mechanisms in order to increase video quality under high error rates. Indeed, H.264 video standard has introduced a set of error resilience tools [7] that make the video stream more robust over packet loss. FMO (flexible macro-block ordering) [8], random intra update refreshing and flexible slice sizes [9] are examples of the introduced error resilience tools.

Manuscript received March 1, 2009; revised June 15, 2009; accepted July 3, 2009.

On the other hand, HSPDA introduces an additional functional layer in the protocol stack, namely the MAC-hs layer. The MAC-hs functionality is implemented in the Node-B, which allows a much faster reaction on errors and variations of the channel quality, compared to protocols implemented in the RNC. This allows to achieve higher data rates with lower delay to mobile users by providing several physical layer improvements such as Adaptive Modulation and Coding (AMC), Hybrid ARQ (HARQ) and a substantially shorter TTI (Transmission Time Interval) of 2ms [10]. Node-B is also responsible for the scheduling between different flows in the downlink to mobile terminals [11] and flow control between Node-B and RNC [12], [13]. The efficiency of these introduced features depends essentially upon the interactions of these techniques with the higher and lower layers. In this context, cross layer interactions can have a drastic impact on overall throughput, video quality and capacity of the cell.

In this paper, we focused our efforts on H.264 video standard over HSDPA. We propose a new ARQ scheme for H.264 video streaming over HSDPA networks. This new ARQ uses the information passed from H.264 encoder in order to achieve unequal error protection of video contents at the link layer. Basically, our proposal assigns deadlines to packets according to the individual importance of each video packet and drops those that exceed their deadlines.

The rest of the paper is organized as follows: in section II, video streaming over HSDPA is presented. Section III explains our proposal. Scenarios, performance metrics used for the evaluation of our proposal and results are presented in section IV. Comparison of our proposal with related work is exhibited in V. Finally, we conclude the paper in Section VI.

II. H.264 Video streaming over HSDPA

As defined in [6], the H.264/AVC architecture is composed of two layers: a Video Coding Layer (VCL) and a Network Abstraction Layer (NAL). The later one has been designed in order to provide simple and efficient conveyance for a broad range of network transport protocols, such as RTP (Real Time Protocol), as well as for storage media, like ISO MP4 and MMS. The fundamental processing unit of the network abstraction

doi:10.4304/jcm.4.9.691-699

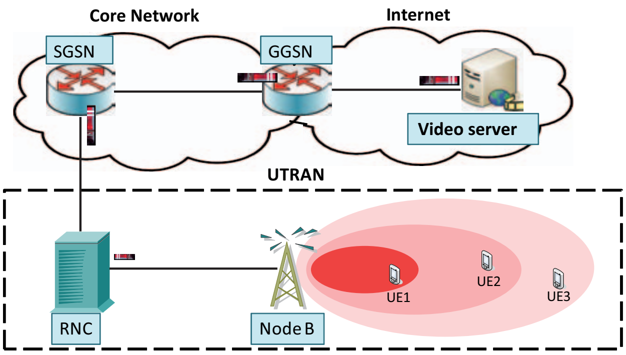

Figure 1. HSPDA video streaming scenario

layer is the NALU (NAL Unit). The NALU could contain video parameters or video data. In the latter case, each NALU will encapsulates a video data unit named *slice*, which is basically a given number of Macro-Blocs associated to a video picture samples. The second layer, Video Coding Layer, is the core compression engine. It enables transform coding with hybrid temporal and spatial predictions. Compared to MPEG-4, H.264/AVC standard can maintain acceptable video quality with up to a 50% reduction in file size, making by this way H.264/AVC ideally suitable for video transmission in a bandwidth limited networks. In particular, when considering wireless networks, such as HSDPA.

Figure 1 depicts a typical H.264 video streaming scenario within a simplified HSDPA architecture. Generally, a mobile User Equipment (UE) requests a streamed video from a H.264 video server located in the Internet. The H.264 video stream is a succession of NALUs (Network Abstraction Layer Units). Thus, each NALU is encapsulated in the RTP/UDP/IP protocol stack. Each video packet is then sent to the UE through the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN) present in the core network. When the RNC (Radio Network Controller) receives this packet, it encapsulates it in one Packet Data Convergence Protocol (PDCP) packet for header compression purposes [14], [15]. The resulting packets are first stored in RNC input buffers with one buffer per data connection. As video packets have variable lengths, the RNC either segments or concatenates incoming data packets into RLC (Radio Link Control) blocks of fixed size named RLC Packet Data Units (RLC-PDUs) [16]. Each PDU is then queued in RLC transmission buffers in order to be transported over a common channel HS-DSCH (High Speed Downlink Shared CHannels).

These RLC blocks are protected by the RLC layer's ARQ mechanism and transmitted to the MAC layer. The MAC layer generates MAC-d (Mac dedicated) PDUs, which are aggregated and sent to the Node-B over the Iub interface in HS-DSCH (High Speed Downlink Shared CHannels) Data Frames. At the Node-B, the MAC-d PDUs are stored in individual Node-B buffers, also known as *HS priority queues*. The Node-B buffers the PDUs until they are scheduled and successfully transmitted over the air interface to a mobile terminal.

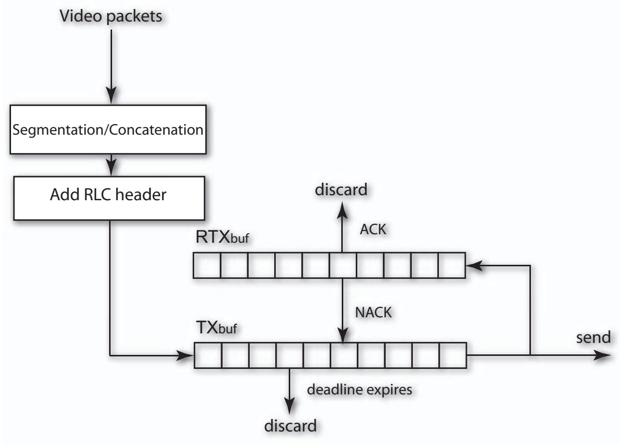

Figure 2. RLC level ARQ

At each Transmission Time Interval (TTI) period, all radio resources are allocated to the flow selected by the scheduling algorithm [11]. The amount of data (PDUs) sent during one TTI depends on the Channel Quality Indicator (CQI), which is sent from UEs to the Node-B at each TTI. Due to time-variant behavior of the wireless channel, different users can experience different channel condition at a given time. Therefore, flow control is performed to regulate the data carrying between RNC buffers and Node-B buffers.

RLC can operate either in *acknowledged mode* (AM) which provides reliable data transfer (i.e. retransmitting erroneous RLC blocks), or in *unacknowledged* (UM) which do not guarantee data delivery. In this paper, we focused on AM, since our proposal enable differentiated bloc retransmission. AM is described in next section.

A. RLC level ARQ: The acknowledged mode

The retransmission strategy adopted by AM (*acknowledged mode*) is the Selective-Repeat ARQ scheme, i.e the only RLC blocks being retransmitted are those that receive a negative acknowledgement [16]. A *status* message is used by the receiver for notifying loss or corruption of an RLC block. The status message is a mixed ACK/NACK flag, in *bitmap* format, for each packet. That is, bit_j indicates whether the j_{th} RLC block has been correctly received or not. The frequency of sending status messages is not specified in the standard. However, several mechanisms are defined, which can trigger a status message. Either the sender or the receiver can trigger the status message. In this paper, we take as assumption that receiver generates a continuous feedback to minimize delay, as proposed in [17].

AM needs a retransmission buffer $ReTX_{buf}$ in addition to the transmission buffer TX_{buf}, as depicted in figure 2. When a packet is sent, it is put in the $ReTX_{buf}$, waiting for its acknowledgement. When a status message is received:

- Packets corresponding to an ACK are discarded from $ReTX_{buf}$ because they have been successfully transmitted.
- The first packet in $ReTX_{buf}$ negatively acknowledged is selected to be sent.
- If there are sufficient credits allocated by the RNC-Node B flow control to the considered video flow, send a new packets to the Node B. Packets in $ReTX_{buf}$ have more priority than those in TX_{buf}.

The key of QoS provided at RLC level is a function named *SDU discard*. The role of this function is to drop from TX_{buf} packets fulfilling one of the two following reasons:

1) The packet is waiting TX_{buf} since a predefined period of time, or
2) The packet has been retransmitted unsuccessfully $MaxDAT$ times.

The second option of the SDU discard function (i.e. after $MaxDAT$ number of transmissions) aims to keep the loss rate constant on the cost of variable delay, which is obviously not suitable for video streaming. The other alternative, which uses a timer-based triggering of SDU discard, strives to keep the connection insensitive to the variations of the wireless channel rate while bounding the maximum delay. The later option is clearly preferable for video streaming. However the key issue, which we address in this paper, is on how to adjust the value of the timer discard function in order to make AM beneficial for a H.264 video streaming?

The simplest and basic approach is to set an identical timer value for all PDUs. This strategy is meaningful when considering an undifferentiated QoS provisioning service. Otherwise, one have to adjust the timer value according to the traffic QoS constraints. In particular, for video streaming a large timer can be chosen assuming a playout buffer at the receiver side. Nevertheless, assigning the same discard timer to all PDUs of a given video streaming flow is not a judicious choice. In fact, video packets have not the same semantic importance and the sender has to consider not only whether a retransmitted packet will arrive in time, but also if that packet is the best choice among all unsent packets.

III. H.264 CONTENT-AWARE ARQ

In this work we focuses on the streaming on the video packets from a base station to a mobile device. We consider that video data is compressed according to the new H.264 standard [18]. This standard provides a set of error resilience tools, which is of great interest, given the need for such mechanisms in the specific context of a wireless link, such as a HSDPA. For example, in H.264 proposes a Flexible Macro-block Ordering (FMO) tool that increments error robustness over losses. In fact, with FMO each frame is divided into *slice groups* and then each slice group comprises one or more slices. For example, each MB in figure 3 belongs either to slice group 0 or 1. Groups 0 and 1 can contain one or more slices. In this way, a slice can contain MBs without respecting raster scan order. The main idea of using FMO here is to increase the number of well received neighbors MBs for predicting a missing ones by using error concealment algorithms [18]. In the following we consider that the H.264 video codec is generating video flows according to the dispersed FMO scheme.

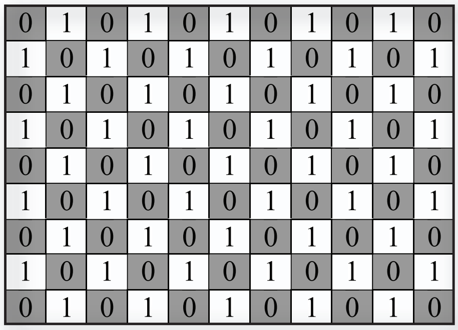

Figure 3. FMO scheme *dispersed*: Grey MBs belong to slice group 0, White MBs belong to slice group 1.

In the following, we present our new ARQ algorithm for video streaming over HSPDA. Our algorithm takes into account the perceptual importance of each packet to determine its deadline.

A. SDU discard function

The SDU discard function discussed in section II-A removes a packet from the buffer TX_{buf} if it does not succeed for a period of time. Let λ be this period of time. Rather than assigning a fixed value to λ for all packets, we propose to assign to each packet p a specific $\lambda(p)$ value, which reflects its importance in the video stream.

Formally, we introduce an *importance function*, noted f, which associates to each packet p a real value reflecting its semantic importance. This value is used to setup the discard timer $\lambda(p)$ of the packet p using the following equation:

$$\lambda(p) = \frac{f(p)}{f_{max}} \times T \tag{1}$$

where f_{max} is the maximum value of the function f and T denotes the initial time before starting the display of video sequences at the mobile terminal. T must be fixed according to the receiver playout buffer size.

Note that the equation (1) is defined such that the most important packets will have T as a deadline. Less important packets will have a less value of λ. Thus, most important packets will be available for transmission at an earlier time, which will increase their transmission opportunities.

The formal definition of the *importance function* (f) is detailed in the following section.

B. Content-aware Importance function

When a video sequence is streamed over a noisy channel, each lost packet causes a reduction of the video quality. The video quality degradation depends on the semantic importance of the lost data. Thus, it is important to introduce a function, which gives for each transmitted packet the semantic importance of its carried data.

H.264 exploits temporal and spatial redundancy to achieve compression of raw video data. While temporal and spatial redundancy increase compression ratio, it also causes the problem of *error propagation*. In fact, loss of a reference video packet will propagate error across multiple frames, which are coded with respect to it.

The goal of our solution is twofold. Firstly, reducing the impact of error propagation due to temporal redundancy by prioritizing frames, which are more referenced by other frames. Secondly, reducing the impact of error propagation due to spatial redundancy by increasing the probability that at least one slice group is correctly received for each frame. For this reason, we define the *importance function* (f) as $f = F + g$, where F is the *temporal importance* of the packet and g the *spatial importance* of the packet.

1) Temporal importance: The basic idea to compute F is to take into consideration that frames which are closer to the beginning of a GOP (Group Of Picture beginning with an I frame) are the most referenced ones and so, are more important than frames in the tail of the GOP. This idea is illustrated in figure 4 where we simulate repeatedly an entire frame loss of foreman video sequence at different GOP positions. We can see that video quality degradation is function of frame position in the GOP. For example, the impact of the lost of frame P_1 is more important than the lost of frame P_5 which is more important than P_9. Thus, we propose to define temporal importance F according to frame positions in GOP. Formally, $F(n)$, the importance of frame n, is computed in respect to its distance from the first frame of its GOP as follows:

$$F(n) = M - n \tag{2}$$

where M is the number of frame in the GoP and n is the position of the n^{th} frame within the GoP.

Let us illustrate this with a simple example. Let $IPPP$ be our GOP structure. Without loosing generality, we suppose in this paper that each frame has two slice groups, thus frames can be noted as $I_{n,i}$, $P_{n,i}$, where n is the frame number and i, $i \in [0, 1]$, the slice group number within the frame. The GOP can be noted as follows:

$$I_{0,0}I_{0,1}P_{1,0}P_{1,1}P_{2,0}P_{2,1}P_{3,0}P_{3,1}$$

After calculation of $F(n)$, we obtain the following result (where bold numbers are the corresponding F values):

$$I_{0,0}(\mathbf{4})I_{0,1}(\mathbf{4})P_{1,0}(\mathbf{3})P_{1,1}(\mathbf{3})P_{2,0}(\mathbf{2})P_{2,1}(\mathbf{2})P_{3,0}(\mathbf{1})P_{3,1}(\mathbf{1})$$

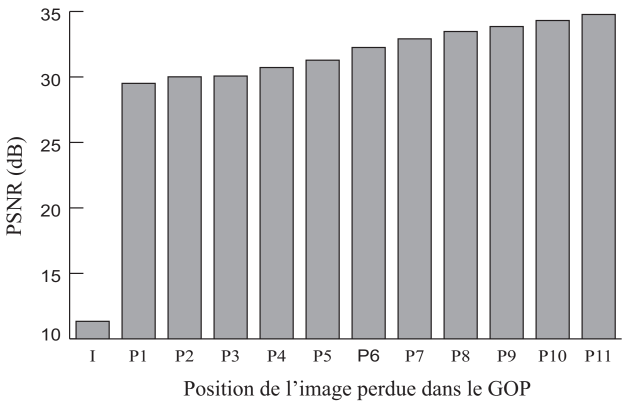

Figure 4. Video quality degradation resulting from the lost of frames in different postions in the GOP.

2) Spatial importance: Slices of frame n are grouped into FMO slice groups denoted $G_{n,i}$, where $i \in [0, 1]$. We assign to these slice groups different importance values in order to ensure that at least one group slice is correctly received in each frame. Since MBs of the correctly received packet are spatially dispersed in the frame, errors resulting of missed packets can be dissimulated perfectly with the decoder.

Formally, the spatial importance value $g(G_{n,i})$ is defined as follows:

$$g(G_{n,i}) = \begin{cases} \mathrm{k} & if \;\; Size(G_{n,i}) > Size(G_{n,1-i}) \\ 0 & if \;\; Size(G_{n,i}) < Size(G_{n,1-i}) \\ i \times \mathrm{k} & if \;\; Size(G_{n,i}) = Size(G_{n,1-i}) \end{cases} \tag{3}$$

where k is a parameter which control the interleaving placement interchange between packets and $Size(x)$ is the size in bytes of group of slices x. The motivation behind this interleaving is that packets belonging to a particular group (e.g. group $G_{n,0}$) in frame n, could be semantically more important than those of another group in frame $n - 1$ (i.e. $G_{n-1,1}$).

In order to illustrate this, suppose that slice group 0 is larger in bytes than group 1 for all frames. In addition, assume that $k = 2$. Therefore, the values provided by the function g, depicted as bold faces numbers, are as follows:

$$I_{0,0}(\mathbf{2})I_{0,1}(\mathbf{0})P_{1,0}(\mathbf{2})P_{1,1}(\mathbf{0})P_{2,0}(\mathbf{2})P_{2,1}(\mathbf{0})P_{3,0}(\mathbf{2})P_{3,1}(\mathbf{0})$$

3) Packet importance: All video packets in one group of slices $G_{n,i}$ take the same importance value which is a sum of its temporal importance and spatial importance. Formally, the importance value of a packet $p_{n,i,j}$ from slice group $G_{n,i}$ is defined as follows:

$$f(p_{n,i,j}) = F(n) + g(G_{n,i}) \tag{4}$$

For the example above, we obtain the result:

$$I_{0,0}(\mathbf{6})I_{0,1}(\mathbf{4})P_{1,0}(\mathbf{5})P_{1,1}(\mathbf{3})P_{2,0}(\mathbf{4})P_{2,1}(\mathbf{2})P_{3,0}(\mathbf{3})P_{3,1}(\mathbf{1})$$

By ordering the slice groups according to their importance values, we obtain an interleaving of slice groups as follows:

$$I_{0,0}(\mathbf{6})P_{1,0}(\mathbf{5})I_{0,1}(\mathbf{4})P_{2,0}(\mathbf{4})P_{1,1}(\mathbf{3})P_{3,0}(\mathbf{3})P_{2,1}(\mathbf{2})P_{3,1}(\mathbf{1})$$

IV. PERFORMANCE EVALUATION

In order to evaluate our proposal, we set up an experimentation platform allowing the emulation of video delivery over HSDPA wireless links. The video sequence is encoded with the H.264/AVC video coding reference software, namely JM [19], in order to obtain a bitstream in conformity with H.264/AVC standard. After that, the encoded stream is parsed and the obtained trace file is used as input to the UMTS/HSDPA simulator. Using the output file of the UMTS/HSDPA simulator, we generate a list of missing IP packets. All the NALUs transported in missing IP packets are removed from the received video stream. The obtained distorted video stream is then decoded and the video quality of the obtained sequence is measured using the PSNR (Peak Signal to Noise Ratio) metric averaged over all frames in the sequence.

In this paper, UMTS/HSDPA network is simulated using the EURANE extension to NS-2 simulator [20]. EURANE models the UTRAN in detail, whereas the nodes SGSN and GGSN are just regular ns-2 nodes used to route IP packets from the Internet to the UTRAN and introduce some delays. In particular, in EURANE all the functions of the RLC (Radio Link Control Protocol) and MAC-hs (MAC in HSDPA) layers are implemented. There are per-flow queues in the RNC and a credit-based algorithm flow control between Node B and RNC. The MAC layer implements the HSDPA scheduler and other functionalities like HARQ. The underlying physical layer is modeled in detail and this model is used to compute a Channel Quality Indicator (CQI) which is feedback from UEs to the base station. Our solution is implemented on the RLC.

Concerning the data flow on UTRAN, each NALU of the encoded video stream is encapsulated in the RTP/UDP/IP packet, with their respective header information. The encoder is instructed to make slices of $100\ bytes$ as maximum size. In all our simulations, we have made the assumption that PDCP layer compresses RTP/UDP/IP header size to 10 Bytes, which is largely feasible using IETF protocols described in RFC 2507 (IP header compression) or RFC 3095 (Robust Header compression ROHC) and considered in [14], [15]. Thus, the maximum size of each video packet is 110 Bytes. The PDU size of feedback channel is the same as for the forward channel and both are equal to $40\ bytes$. In addition, the RLC is configured such that acknowledgement information is repeated in different bitmap PDUs to increase error robustness.

Video sequences are encoded according to the GOP structure $IPPPPPPPPPPP$ with a frame rate of $15\ fps$ (frame per second). The FMO *dispersed* scheme is used as error resilience tool at the encoder side. At the decoder side, an error concealment algorithm is used. Even though we have carried simulations for many video sequences, only results for the well known *foreman* sequence will be shown. The `foreman` video sequence is considered as a complex video scene, since it represents a man speaking to a camera with a charged background and a more pronounced movements. Finally, we fix the initial time before starting the display of video sequences, T, to $1000\ ms$ in all cases except otherwise indicated.

TABLE I.
SIMULATION PARAMETERS

Parameter	Value
TTI	2ms
Maximum UE disatance from node B	800m
HARQ processes	6
Time delay (TTI) in CQI	3
Maximum HARQ retrasmissions	2
$BLER_{target}$	10%
Scheduler at node B	Proportional Fair
Multipath fading environment	Pedestrian A
User Speed	$3km/h$
node B Transmission power	38 dBm
Base station antenna gain	17 dBi
I_{intra} (intra cell interference)	30dBm
I_{inter} (inter cell interference)	-70dBm
Maximum Data Rate	3.6 Mbps
UE type	6
Credit-allocation interval	30ms
RLC mode	AM
RLC PDU size	40 byte
RNC-UE RTT	100ms
Video sequence	foreman
Video format	CIF
Target video rate	300 Kbps
Slice size	100 bytes
GoP size	12
GoP structure	IPPPPPPPPPPP
Resilience tool	FMO
Number of Video Users	4
Initial time (T)	1000ms
Number of simulation runs	10
Simulated time per run	24s
Video quality Metric	PSNR
Frame rate	15 fps

Table 1 summarizes the default values of simulation parameters; these parameters remain unchanged in all scenarios unless otherwise indicated. As indicated in this table, maximum HARQ retransmission is 2. It is well noting that CA-ARQ and HARQ are two mechanisms operating at different levels and different equipments. In fact, CA-ARQ is implemented at the RLC level and in the RNC equipment. However, HARQ operates at MAC-hs layer and is implemented at the Node B. In this way, CA-ARQ and HARQ are complementary rather than competitive.

We consider a single-cell environment, where 4 User Equipments (UEs) connect to the Node B via a High Speed Downlink Shared Channel (HS-DSCH) in the downlink and a dedicated channel (DCH) in the uplink direction. The Node B is connected to the RNC, which itself is connected to the Internet via the 3G-SGSN and 3G-GGSN of the cellular system's core network. The UEs establish a data connection with a host in the Internet. In order to alleviate the impact of scheduling algorithms, all UEs are moving on circles at a distance d from the node B, where d varies from 200 to 800 meters.

In order to obtain valid statistical results, all results shown in this section are the average of 10 simulation runs.

Our mechanism, refereed as CA-ARQ (for Content-

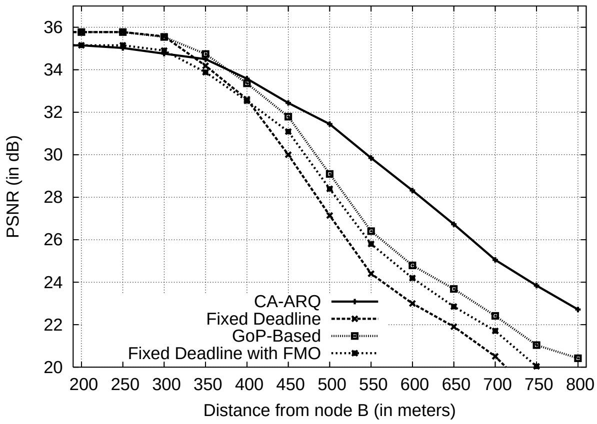

Figure 5. PSNR video quality with $T = 1000$ ms: Quality is considered as poor if PSNR is under 25 dB

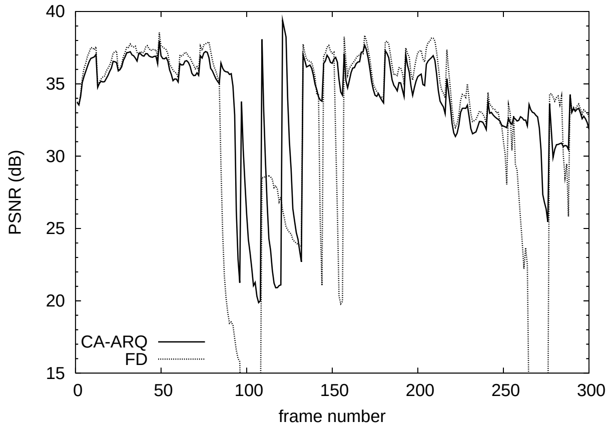

Figure 6. PSNR video quality per frame for UE distance from Node B of 450 meters: peaks in CA-ARQ curve are relating to I frames.

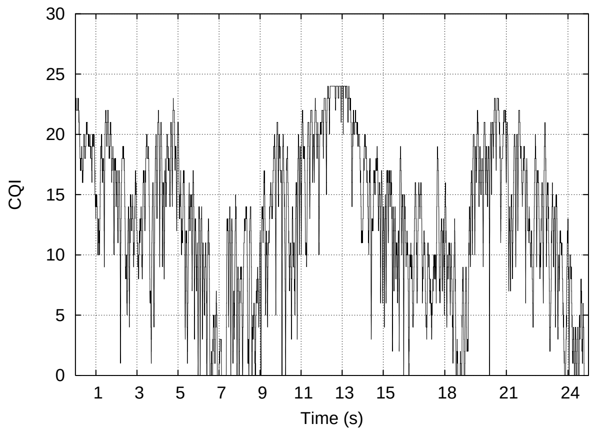

Figure 7. CQI reported by the terminal of figure 6

aware ARQ), is compared to three other mechanisms:

1) *Fixed deadline* mechanism, in which all packets in the video stream are given the same importance and deadline values. In this mechanism, FMO is not used;
2) *Fixed deadline with FMO* mechanism, in which all packets in the video stream are given the same importance and deadline values and the FMO *dispersed* scheme is enabled;
3) *GOP-based deadline* mechanism, in which all packets in the same frame X_n are given the same importance. Moreover, FMO is not used for this mechanism.

Figure 5 shows the PSNR performance according to the distance of the UEs from Node B. For distances from Node B that are less than 300 meters, *Fixed deadline* and *GOP-based deadline* mechanisms outperform CA-ARQ and *Fixed deadline with FMO* mechanisms. This small difference in PSNR is due to the use of FMO in CA-ARQ and *Fixed deadline with FMO* mechanisms. Indeed, the use of FMO adds a small overhead to the encoded sequence. Thus, for a specific target encoding rate, no-FMO methods slightly exceed FMO-based methods like CA-ARQ. However, when the distance from Node B increases, beyond 300 meters, PSNR for *Fixed deadline* mechanism decreases rapidly and the video quality becomes not acceptable at high error rates. The combination of the importance based deadlines and the use of FMO in the CA-ARQ mechanism leads to the best observed video quality. The difference over *GOP-based deadline* mechanism is very clear and reaches $2.5dB$, which is significant for mobile terminals. It is well noting that the use of FMO without adaptive deadlines at RLC does not give acceptable quality. Thus, FMO must be completed with adaptive network mechanisms, such CA-ARQ, to obtain better performance.

In figure 6, peaks in CA-ARQ curve are relating to I frames. Therefore, CA-ARQ brings PSNR up to high values each time a new GOP is to be sent. This results in better average PSNR values. Furthermore, figure 7 illustrates the CQI reported by the UE of figure 6. It is clear that when the CQI is low, the available bandwidth becomes small and the PSNR decreases. However, CA-ARQ is more robust than *Fixed deadline* since the PSNR it brings PSNR up to high values each time a new GOP is to be sent.

The packet loss rate per NALU type is depicted in the figure 8. From that figure, it is clear that I packet loss rate is less than P packet loss rate. Thus, the attribution of deadlines to packets achieves unequal error protection in CA-ARQ. Furthermore, figure 9 depicts the packet loss rate in terms of slice group and position in the GOP. This figure shows clearly that the proposed mechanism permits to prioritize one slice group over the other in each frame. In addition, packet loss increases in terms of the distance of its containing frame from the GOP beginning. This result can be easily explained as follows: larger values of the importance function f in equation (1) leads to larger values of the function λ. This allows packets to become available for transmission at an earlier time, thus increasing their transmission opportunities. This remark is confirmed by the figure 10, where we plot the average number of retransmission in terms of frames position in

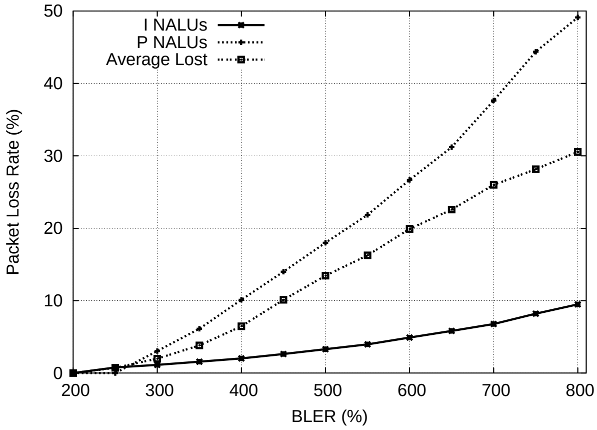

Figure 8. Packet loss ratio per NALU type: unequal error protection is achieved

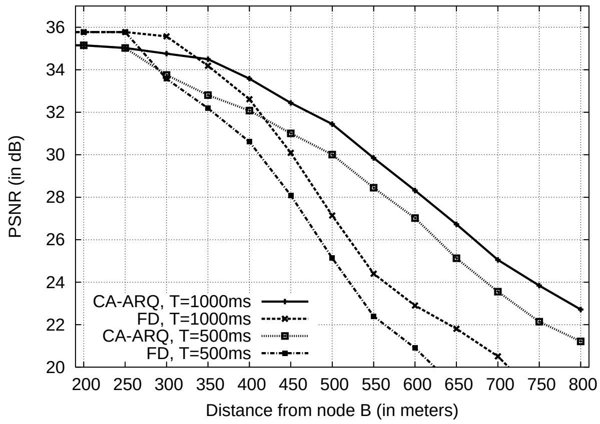

Figure 11. Impact of T values on video quality

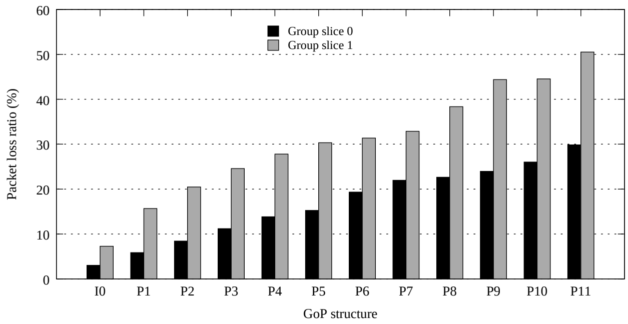

Figure 9. Packet loss ratio per slice group and position in the GoP with distance from Node B of 600 meters

the GOP, given that the distance from the Node B is 600 meters.

The initial time T is an important parameter in video streaming. Usually, it depends on the receiver playout buffer size. Figure 11 depicts the PSNR for T values of 1000 ms and 500 ms. For both CA-ARQ and *Fixed deadline*, PSNR quality is greater for larger values of T. Indeed, for larger values of T in equation (1) we obtain larger values of the function λ of the CA-ARQ mechanism. Here again, the availability of packets for transmission for larger time increases their transmission opportunities. The same remark holds for *Fixed deadline*

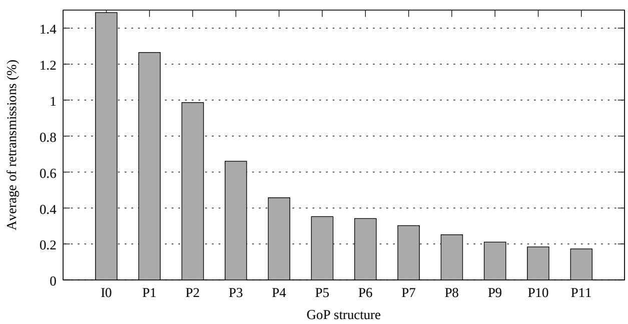

Figure 10. Average number of retransmissions per GOP position with distance from Node B of 600 meters

since the fixed deadline calculated for each packet depends on T.

The PSNR is an objective quality assessment metric. In order to show the subjective enhancement in quality within our mechanism, let us discuss the figure 12. In this figure, we have illustrated two original frames (84 and 324) and their respective decoded frames using the *Fixed deadline* and CA-ARQ mechanisms, respectively. The error pattern used in the two mechanisms is the same. We can see that using the *Fixed deadline*, the impact of error is a large area and that the error concealment mechanism fails to bring up the video quality. However, using CA-ARQ the MBs losses is reduced and visually it is clear that the video quality is better.

V. Related works

Extensive researches have been carried out to enhance error robustness for wireless video transmission. Because of the extreme error sensitivity of hybrid coded video, interaction between the network and application is needed. In [21], an application level proposal for H.264 error robustness over UMTS links is described. The principle of application level proposal is that the application (i.e video codec) shall play a significant role in network communications. In practice, the important end-to-end delay at application level make retransmissions based solutions difficult to adopt. Hong-Bin et *al*. [22] propose an adaptive ACK/NACK mode switching scheme for H.264 video coding. A video proxy server, which interacts quickly with both encoder and decoder, is implemented. Rather than implementing a video proxy, our proposal presented in this paper is a MAC level solution. Since the decision is made at a lower layer, our solution can interacts more quickly than a proxy server.

In the aim to assess importance of video content, proposals were made on the basis of different syntax levels of video stream. We can classify them into Macro-Blocks (MB) based [23], slice-based [24]–[26], and frame-based [27], [28] methods. The retransmission policy of the ARQ scheme proposed in [24] and [25] is slice-based and it is driven by the information about the perceptual and the

(a) frame 84, original (b) frame 84, with FD (PSNR=17.5) (c) frame 84, with CA-ARQ (PSNR=22.2)

(d) frame 324, original (e) frame 324, with FD, (PSNR=20.8) (f) frame 324, with CA-ARQ, (PSNR=26.6)

Figure 12. Example of the impact of packet loss in the frame 84 and in the 324, using FD and CA-ARQ

temporal importance of each packet. Distortion caused by the lost of each packet is taken as the importance value of that packet. This method supposes to have all video content on server before starting streaming. Based on the same packet importance principle, [26] proposes a content-aware retry limit adaptation scheme for video streaming over 802.11 WLANs. Video packets of different importance are unequally protected with different retry limits at the MAC layer. In the same way, a Time-based Adaptive Retry (TAR) mechanism for video streaming over 802.11 WLANs is presented in [27]. TAR determines whether to (re) transmit or discard a packet based on the retransmission deadline attached to that packet. However, importance function is frame-based and thus, all packets of the same frame have the same importance. Another frame-based method is presented in [28], where importance function is done directly on the syntax elements of the compressed video stream (I, B or P). The disadvantage of frame-based methods is that all packets of a frame are given the same importance regardless of distortion caused by each one.

A conditional retransmission MB-based strategy is proposed in [23]. The distortion caused is calculated at MB level. This solution, and MB-based strategies in general, induce high complexity for the video sender.

Our importance function is a slice-group based method. As we have seen in this paper, results obtained for the proposed method is much better than those for GOP-based deadline method, which is a frame-based solution. Furthermore, the use of FMO concept allows us to avoid the complexity of the calculation of distortion value caused by the lost of each individual packet. This complexity is the main drawback of slice-based methods. Finally, since our solution is MAC level based method, the complexity of MB-based solutions is completely avoided.

VI. Conclusion

In this paper we proposed and analyzed a new MAC level content-aware ARQ scheme for video streaming over HSDPA wireless links. The proposed scheme assesses the importance of each packet using the FMO tool, which is a new error robustness tool of H.264 video encoding standard. Temporal values are then given to video packets in respect of their importance. Simulations showed that our content aware ARQ mechanism achieves unequal error protection and outperforms all simulated mechanisms.

References

[1] G. T. 25.308, *High Speed Downlink Packet Access HSDPA; Overall description; Stage 2; Release 8*, v8.5.0, March 2009.

[2] M. van der Schaar and P. A.Chou, *Multimedia Over IP and Wireless Networks: Compression, Networking, and Systems*. ACADEMIC PRESS, March 2007.

[3] B. Girod and N. Farber, “Feedback-based error control for mobile video transmission,” in *Special Issue on Video for Mobile Multimedia* (IEEE, ed.), vol. 87, pp. 1707–1723, October 1999. Invited paper.

[4] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, “Rate/distortion optimized hybrid error control for real-time packetized video transmission,” *IEEE Trans. on Image Processing*, vol. 15, pp. 40 – 53, Jan. 2006.

[5] F. Zhai and A. Katsaggelos, *Joint Source-Channel Video Transmission*. Synthesis Lectures on Image, Video, and Multimedia Processing, Vol. 3, No. 1, Pages 1-136, 2007.

[6] I.-T. R. H.264, *Advanced vido coding for generic audiovisual services*. ITU-T, Mar. 2009.
[7] L. Lin, Y. Xiu-zi, Z. San-yuan, and Z. Yin, "H.264/AVC error resilience tools suitable for 3G mobile video services," *Journal of Zhejiang University SCIENCE*, pp. 41–46, 2005.
[8] P. Lambert, W. De Neve, Y. Dhondt, and R. Van de Walle, "Flexible macroblock ordering in H.264/AVC," *Elsevier J. Visual Commun. Image RepresentationF*, vol. 17, pp. 358 – 375, 1 2006.
[9] S. Wenger, "H.264/AVC over IP," IEEE *Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.
[10] A. Toskala and H. Holma, *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. Wiley, April 2006.
[11] F. Feller and M. Necker, "Comparison of opportunistic scheduling algorithms for HSDPA networks," in *Proceedings of the 12th Open European Summer School and IFIP WG 6.3, WG 6.4, WG 6.6 and WG 6.9 Workshop*, (Stuttgart), 2006.
[12] M. Necker and A. Weber, "Impact of Iub flow control on HSDPA system performance," in *16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications PIMRC*, (Berlin, Germany), 2005.
[13] P. J. Legg, "Optimized Iub flow control for UMTS HSDPA," in *IEEE Vehicular Technology Conference (VTC 2005-Spring)*, June 2005.
[14] 3GPP TS 25.323, *Packet Data Convergence Protocol (PDCP) specification, Release 8.4.0)*, March 2009.
[15] F. Fitzek, S. Hendrata, P. Seeling, and M. Reisslein, "Video quality evaluation for wireless transmission with robust header compression," tech. rep., Acticom mobile networks, 2003.
[16] G. T. 25.322, "Radio link control (RLC) protocol specification (Release 8), v8.4.0, march 2009.".
[17] M. Rossi, L. Scaranari, and M. Zorzi, "On the UMTS RLC parameters setting and their impact on higher layers performance," in *Vehicular Technology Conference, 2003. VTC 2003-Fall*, pp. 1827– 1832, Oct. 2003.
[18] T. Stockhammer, M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 657– 673, July 2003.
[19] H. R. Software, "http://iphome.hhi.de/suehring/tml/.".
[20] E. U. G. R. 1.11), "http://www.ti-wmc.nl/eurane/.".
[21] O. Nemethova, W. Karner, A. Al-Moghrabi, and M. Rupp, "Cross-layer error detection for H.264 video over UMTS," in *International Wireless Summit 2005*, (Aalborg, Denmark), September 2005.
[22] H.-B. Yu, S. Yu, and C. Wang, "A highly efficient, low delay architecture for transporting H.264 video over wireless channel," *Signal Processing: Image Communication*, vol. 19, pp. 369–385, April 2004.
[23] S. Aramvith, C. Lin, S. Roy, and M. Sun, "Wireless video transport using conditional retransmission and low-delay interleaving," 2002.
[24] P. Bucciol, E. Masala, and J. D. Martin, "Perceptual arq for H.264 video streaming over 3G wireless networks," in *Proceedings of IEEE Int. Conf. on Communications ICC*, June 2004.
[25] P. Bucciol, E. Masala, E. Filippi, and J. D. Martin, "A perceptual arq algorithm for H.264 live streaming over congested 802.11e wireless lans," *WSEAS Transactions on Communications*, pp. 99–106, January 2006.
[26] C.-M. Chen, C.-W. Lin, and Y.-C. Chen, "Packet scheduling for video streaming over wireless with content-aware packet retry limit," in *IEEE 8th Workshop on Multimedia Signal Processing*, Oct 2006.
[27] M. Lu, P. Steenkiste, and T. Chen, "Video streaming over 802.11 WLANs with content-aware adaptive retry," in *IEEE International Conference on Multimedia and Expo (ICME)*, July 2005.
[28] Shan and Yufeng, "Cross-layer techniques for adaptive video streaming over wireless networks," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 2, pp. 220–228, 2005.

Salim Benayoune received his Engineer and Master degrees in computer science from Mentouri University (Constantine, Algeria) in 2000 and 2003 respectively. He received his Master in networking from Pierre and Marie Curie University in 2004 and his Ph.D. degree in Networking and information technology from Paris 13 University in 2009. From 2007 to present he has been working as an assistant professor at Paris 13 University. His research focuses on adaptive mechanisms for wireless video transmission.

Nadjib Achir was born in Algiers, Algeria, on September 1976. He graduated from the National Institute of Computer Science, Algiers, Algeria, in 1998, where he received my Engineer Diploma in computer science with honors. Thereafter, he gained an MSC (DEA) diploma with distinction from the University of Versailles Saint-Quentin-en-Yvelines, Versailles, France; in 1999. In 2003, Nadjib Achir received his Ph.D. degree in computer science from the University of Pierre and Marie Curie (Paris 6). He joined Paris 13 University as an Associate Professor in 2004. He is serving on the program committees of many international conferences. His research interests include mobile ad hoc networks, wireless sensor networks, mesh networks and wireless multimedia communications.

Khaled Boussetta is associate professor at University Paris 13, since 2004. He received his M.S. in Computer Science from University Paris 6 in 1996 and his Ph.D. Computer Science from University of Versailles in 2003. In 2001, he obtained an INRIA fellowship and worked for one year as a visiting researcher at the Network Research Lab at UCLA. In 2003 he joined the Networks and Performances Analysis team of LIP6 as a postdoc researcher. His research interests cover design, modeling, and performance evaluation of networks. Currently, his research focuses on QoS support in wireless networks, sensor networks and multimedia applications. He is actually the leader of a French National Research Agency funded project entitled MadGames. He is actively involved in the community as a conference chair, TPC member, and a reviewer for many conferences and journals.

Ken Chen was born in Shanghai (China) on 1960. He received the Engineer Diploma from SUPELEC (Institue on electric and electronic engineering, France) in 1985, and the Doctorate Degree from University Paris 11 (France), in 1988. From 1988 to 1990, he has been a researcher at INRIA (France). From 1990 to 1997, he served as a Maitre de Conferences at ENST (currently known as Telecom ParisTech). Since 1997, he joined the university Paris 13 as a Professor. His current interests are in the area of video communication, optical networks and the FMC issues in ambient networks.