Evaluation of Channel Encoders for H.264/AVC Video Transmission over IEEE 802.11p

Ismael Rozas-Ramallal [1], Tiago M. Fernández-Caramés [2], Adriana Dapena [3], José Luís Martínez Martínez [4]

Summary—**This article details different experiments aimed at evaluating the performance obtained when transmitting video files coded with H.264/AVC standard over IEEE 802.11p networks. Specifically, we have evaluated two channel encoding configurations: a 1/2-convolutional code (as defined in the IEEE 802.11p standard) and Low-Density Parity-Check Codes (LDPCs). Our results show that, in the experimental conditions selected, the utilization of LDPCs considerably improves the percentage of videos reconstructed successfully at reception. However, LDPCs lead to decoding delays that are larger than the ones obtained when using convolutional codes. In order to mitigate this limitation, we propose a hybrid scheme that makes uses of an LDPC to code the most relevant headers and frame coefficients, while the rest of the data are coded with convolutional codes.**

Keywords— **H.264/AVC, IEEE 802.11p, Low-Density Parity-Check Codes (LDPC), convolutional codes.**

I. Introduction

Vehicular communications have recently attracted a great deal of attention to the field of intelligent transportation systems due to the demand for solutions aimed at providing safety and non-safety services. Non-safety services offer, for instance, support for infotainment applications or the exchange of multimedia information between vehicles. The standard IEEE 802.11p [1] is probably the best positioned to provide this kind of services since it is an amendment to IEEE 802.11-2007 [2] that addresses the challenges that arise when providing wireless access in vehicular environments. A deep description of IEEE 802.11p is beyond the scope of this paper, but we encourage the interested reader to take a look at the excellent overviews given in [3] and [4].

In recent years it has been also performed significant effort to develop multimedia standards to satisfy the increasing demand of multimedia contents. In particular, the H.264/AVC standard has enhanced compression performance and provided a proper video representation for network transmission, addressing “conversational” (video telephony) and “non-conversational” (storage, broadcast or streaming) applications. Furthermore, H.264/AVC achieves a significant improvement in rate-distortion respect to existing standards [5, 6].

Recently, it has been analyzed the performance of H.264 video streaming in inter-vehicular environments using the IEEE 802.11 ad hoc network protocol [7]. However, we are not aware of any published results regarding the transmission of video coded with H.264/AVC over IEEE 802.11p. As it will be shown in Section IV, the convolutional codes used by default by IEEE 802.11p obtain, in certain experimental conditions, poor results. For such reason, we have also evaluated the performance when using Low-Density Parity-Check Codes (LDPCs) [8] in substitution of the convolutional codes. LDPCs obtain a remarkable performance improvement but they lead to the degradation of two critical aspects: computational load and decoding delay. In order to mitigate these limitations, we propose a hybrid scheme where only the most relevant data are coded using an LDPC, while the rest are coded using a convolutional code.

This work is organized as follows. Section II describes the H.264/AVC layer model, analyzing the different components. Section III describes briefly the IEEE 802.11p simulator developed by the *Grupo de Tecnología Electrónica y de las Comunicaciones* (GTEC) from the University of A Coruña. Section IV evaluates the performance when transmitting H.264/AVC over IEEE 802.11p considering convolutional codes and LDPCs. Finally, Section V presents the hybrid profiles and Section VI contains the conclusions and the future work.

II. H.264/AVC layer model

H.264/AVC provides a simply standardized structure for encapsulating compressed video and its related information (represented in Figure 1).

At the top level, every frame can be divided into one or more slices and transmitted into one or several Network Adaptation Layer Units (NAL Units or NALUs), having its own header (which stores the frame type). During transmission, each NAL corresponds to a Real Time Protocol (RTP) packet. The first NAL contains the Slice Parameter Sets (SPS) and the second one the Picture Parameter Set (PPS), which are key parameters used by the decoder to decode video data and slices fast.

Like previous standards [9], H.264/AVC defines three frame types: Intra frames (I-frames), Predic-

[1]Dpto. de Electrónica e Sistemas, Universidade da Coruña, e-mail: `ismael.rozas@udc.es`
[2]Dpto. de Electrónica e Sistemas, Universidade de A Coruña, e-mail: `tmfernandez@udc.es`
[3]Dpto. de Electrónica e Sistemas, Universidade da Coruña, e-mail: `adriana.dapena@udc.es`
[4]Dpto. de Sistemas Informáticos, Universidad de Castilla-La Mancha, e-mail: `joseluismm@dsi.uclm.es`

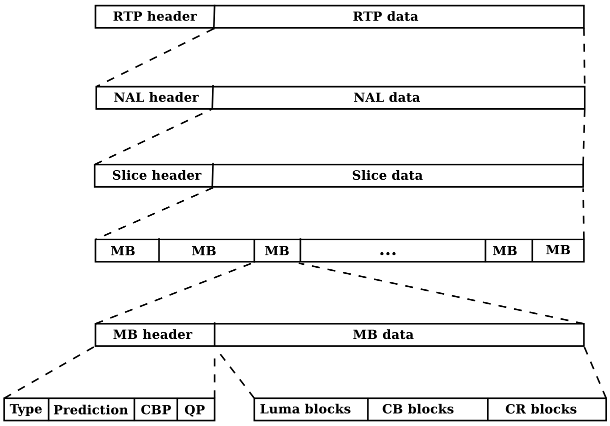

Fig. 1. H.264 layer structure

tive frames (P-frames) and Bidirectional frames (B-frames). Every frame is transmitted inside one data NAL, as a slice, having its own header (which stores the frame type) and the data. In Figure 1, it can be seen that the slice data contain Macro Blocks (MB) information. Each MB describes a particular choice of methods used to code the macroblock, the prediction information (such as coded motion vectors or intra prediction mode information) and coded residual data.

In this layer structure we can see a clear differentiation between headers (RTP header, NAL header, Slice header and MB header) and frame coefficients (MB data). In particular, MB headers are very sensitive to errors in transmission, being the decoder and video player unable to decode and reproduce it due to its structuring nature on video compression.

III. IEEE 802.11p Software Simulator

The experimental results shown in this paper have been obtained using the software simulator developed by the *Grupo de Tecnología Electrónica y de las Comunicaciones* of University of A Coruña [10]. The simulator, developed in MATLAB® and Simulink®, contains the blocks depicted in Figure 2.

The transmission performs the steps shown on the left of Figure 2. First, the data are scrambled, coded and interleaved. The scrambler uses a 127-bit pseudo-random sequence, the encoder is a rate 1/2-convolutional and the interleaver performs a two-step permutation: the first permutation ensures that adjacent coded bits are mapped onto non-adjacent subcarriers, while the second permutation ensures that adjacent coded bits are mapped onto less and more significant bits of the constellation to avoid long runs of low reliability. It is important to highlight the fact of being able to change the 1/2-convolutional channel encoding to other channel encoders like, for instance, LDPC. After interleaving, the bits are Gray-mapped into Quadrature Amplitude Modulation (QAM) symbols and placed into 64 subcarriers. Four subcarriers are dedicated to pilot signals modulated by a pseudo binary sequence. Forty-eight of the rest of the subcarriers are used for placing the data symbols. The subcarrier 0 is reserved for the DC and the remaining subcarriers are for frequency guards. Each group of 64 subcarriers is modulated using Orthogonal Frequency-division Multiplexing (OFDM), what implies that the Inverse Fast Fourier Transform (IFFT) is applied. Finally, a 1/4 cyclic prefix (CP) is added to prevent Inter Symbol Interference (ISI).

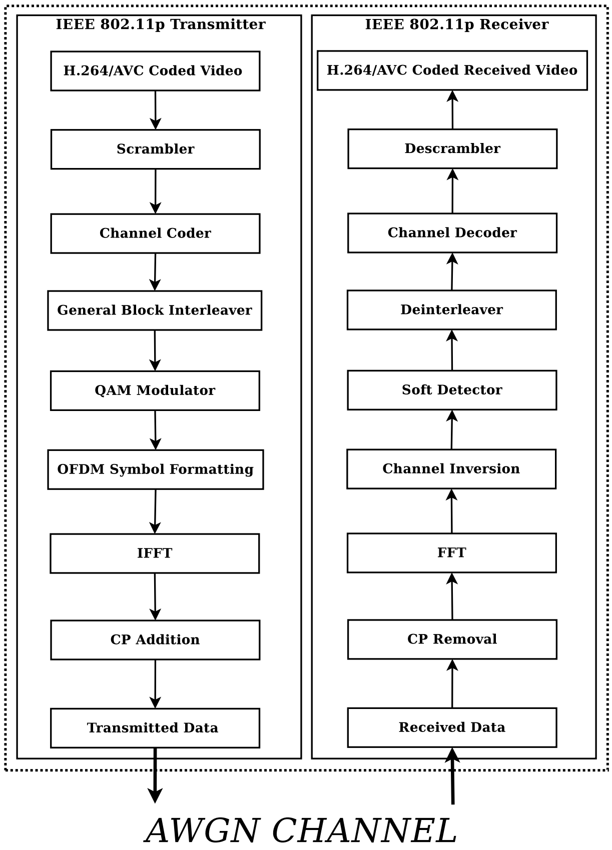

Fig. 2. IEEE 802.11p simulator

The receiver blocks are shown on the right of Figure 2. The first step consists in removing the CP. Then, the FFT is applied to each OFDM symbol. Next, the channel is estimated using the four pilots, obtaining the estimated channel coefficients for the pilot subcarriers. The four channel coefficient estimates are linearly interpolated to obtain the channel frequency response for the rest of the subcarriers. After this, an MMSE (Minimum Mean Square Error) equalizer is employed. Finally, the equalized symbols are sent to a soft detector, whose outputs are deinterleaved, inverting the permutations performed in the transmitter and the decoder carries out decoding.

IV. Experiments: H.264/AVC over IEEE 802.11p

An important aspect to consider in the standard IEEE 802.11p is the performance obtained when the data are protected using a convolutional code. The convolutional code is a type of error-correcting code in which m information bits are transformed into n coded bits [11], being $R = m/n$ the code rate. The convolutional encoders are characterized by the generator polynomials. In particular, IEEE 802.11p defines rates of $R = 1/2$, $R = 2/3$ and $R = 3/4$.

For the case of $R = 1/2$, the system uses the industry-standard generator polynomials $g_0 = 133$ and $g_1 = 171$ [1]. It is also recommended to use the Viterbi algorithm in the decoder.

Besides convolutional codes, in this paper we also evaluate the performance of the system when using an LDPC code [8]. LDPC codes are a class of linear block codes characterized by a parity check matrix $\mathbf{H}$ with d_v ones in each column and d_c ones in each row, where d_v and d_c are chosen as part of the codeword design and are small in relation to the codeword length [11]. Since the fraction of non-zero entries in $\mathbf{H}$ is small, the parity check matrix for the code has a low density. Provided that the codeword length is long, LDPC codes achieve performance close to the Shannon limit. LDPC codes tend to have relatively high encoding complexity (quadratic in block length) but low decoding complexity. In order to compare it directly with convolutional code, it is used a $R = 1/2$ LDPC and a H matrix with dimensions 144×288 and 864 ones, three per column.

In the next subsections, it is compared the resistance of convolutional codes and LDPCs when they are used to protect the transmissions of H.264/AVC-coded content over IEEE 802.11p. As it was mentioned in Section II, H.264/AVC defines tree frames types (I, P and B) and the packets contains four kinds of headers (slice, NAL, macroblock and RTP). The evaluation is performed considering 10 frames of a typical video such as foreman sequence in QCIF format (176×144 pixels) [12]. The Group of Pictures (GOP) used in the simulations is formed by one I-frame, three P-frames and six B-frames. The simulations were carried out coded with the JM implementation of H.264/AVC which is the reference software [13]. The channel introduces Additive White Gaussian Noise (AWGN) in an operative E_b/N_0 from 8 to 11 dB. The results have been obtained by averaging 1000 realizations of AWGN channel.

A. First Experiment: Impact of header transmission

First, we consider the impact of only adding noise to the headers with AWGN. We have carried out three different experiments:

- Adding noise only the headers corresponding to I-frames.
- Adding noise only the headers corresponding to P-frames.
- Adding noise only the headers corresponding to B-frames.

Therefore, for each experiment, we have added the noise only to one type of header and we have evaluated the probability of recovering all frames (10 frames). The concept of *success probability* is defined as the number of times when the video has been completely decoded divided by the total number of realizations. Note that this is a “hard” measure because video parts can be recovered without needing all frames (the only requeriment is to have at least the I-frame).

Figure 3 shows the results obtained for each type of frame and header when the 1/2-convolutional code is used. It can be seen that the impact of adding noise to the macroblock headers leads to a considerable reduction in the success percentage while the effect of the other headers is not so relevants. Moreover, we can see that the most important degradation occurs for the macroblock headers corresponding to P-frames. For instance, the success percentage for a E_b/N_0 of 8 dB is about 35% for P-frames, 55% for I-frames and 59% for B-frames. These results are due to the fact that there exists more P-frames than I-frames, so the effect of error in P-frame headers have more impact than the effect of I-frame headers. Note also that, although there are more B-frames, since P-frames are needed to predict B-frames, the errors on the P-frame headers influence both P and B-frames.

On the other hand, Figure 4 shows the results when substituing the convolutional code with a 1/2-LDPC. It is clear that the quality of this kind of code is considerably higher than the one obtained when using convolutional codes (note that the success probability is close to 100% for all headers).

B. Second Experiment: Impact of coefficient transmission

Our second study focuses on the impact of AWGN on the frame coefficients. We have considered three different situations:

- Only the coefficients corresponding to I-frame are perturbed.
- Only the coefficients corresponding to P-frame are perturbed.
- Only the coefficients corresponding to B-frame are perturbed.

In this case the headers are transmitted without perturbation. Like in the previous experiments, it was evaluated the performance when reconstructing 10 frames.

Figure 5 shows the success probability when the 1/2-convolutional code is used to protect the data. As it can be observed, the noise has more impact on the I-frames, because it is the reference frame for the others. The P-frames are also relevant because they can be used as references for B-frames.

Figure 6 shows the results obtained when using LDPCs. The improvement respect to the utilization of 1/2-convolutional code is remarkable. Note that the performance is nearly 100% for coefficients corresponding of P-frames and B-frames. For the case of I-frames, a similar percentage is obtained for values of E_b/N_0 larger than 9 dB.

V. Hybrid Profiles

The simulation results presented in previous section show that the percentage of reconstructed frames achieves using LDPC is considerably better than the one obtained with convolutional code. However, previous studies [14], show that LDPC imposes

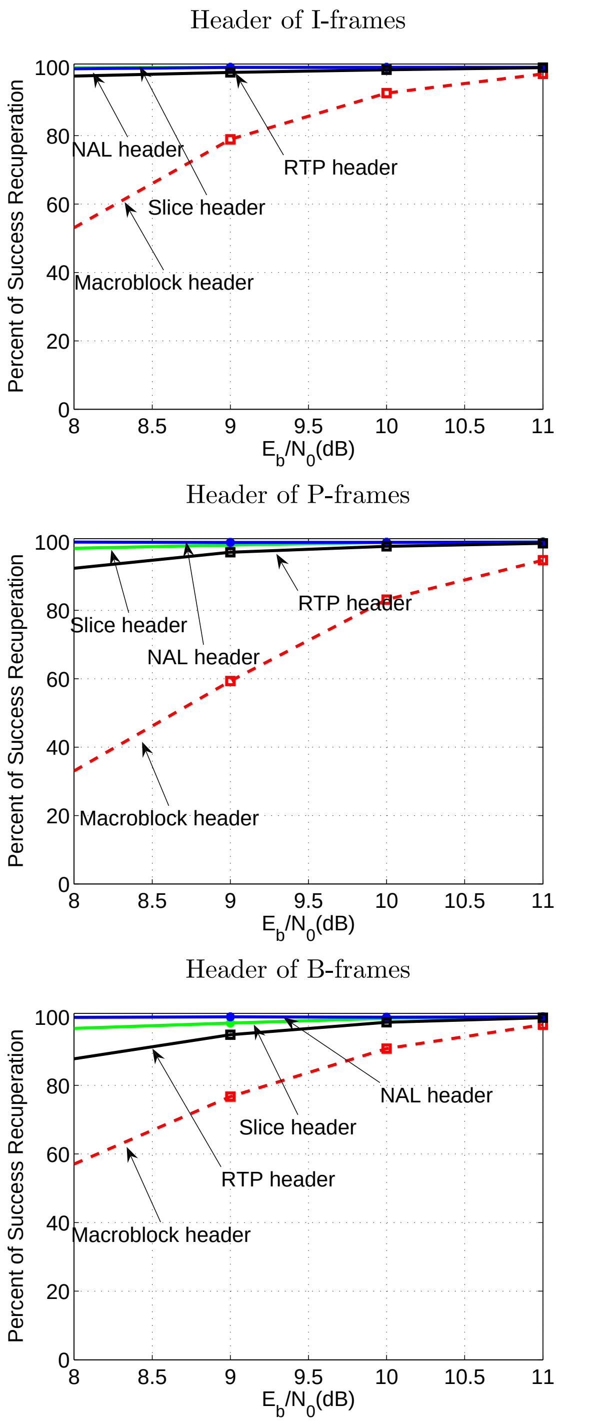

Fig. 3. Convolutional code: percentage of successful reconstruction when individual headers are affected by AWGN

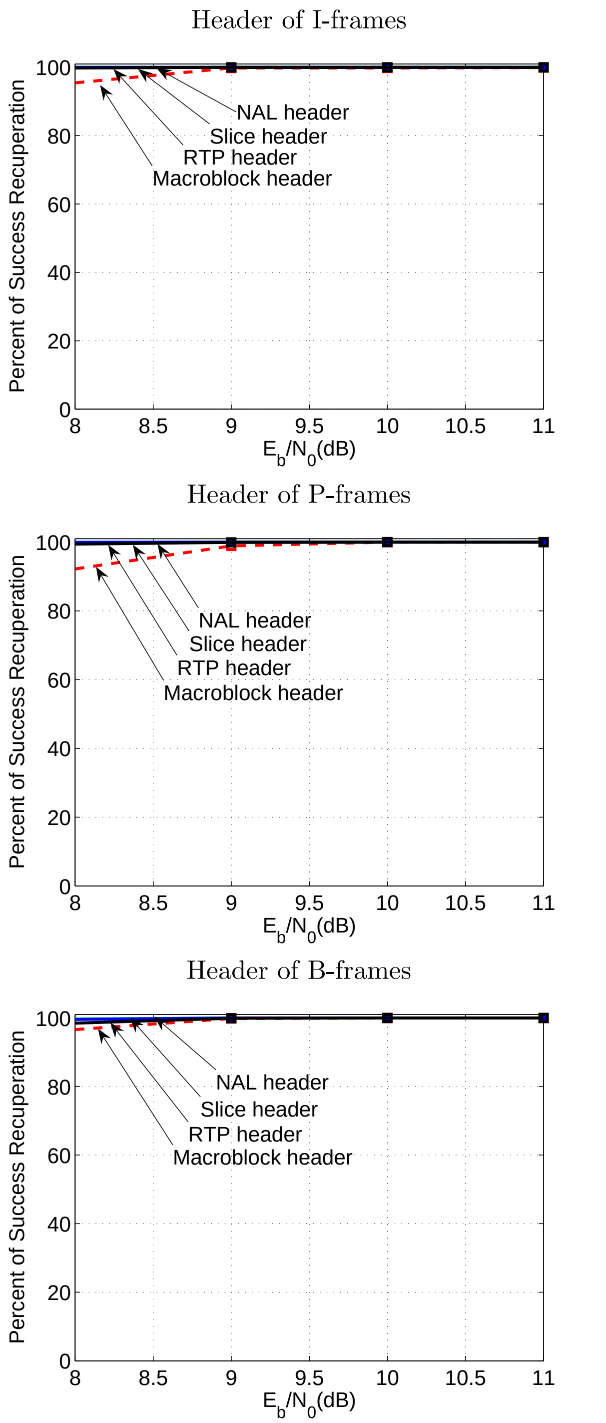

Fig. 4. LDPC: percentage of successful reconstruction when individual headers are affected by AWGN

a larger delay due to the block structure while, in general, convolutional codes are known to show a good performance with a very low latency.

After analyzing the impact of each video part in the H.264/AVC arquitecture, it is clear that it would be useful to define different channel coding profiles in order to adapt the video quality and transmission performance to the E_b/N_0 in transmission. Four profiles are evaluated in this section:

1. *Convolutional code profile*: 1/2-convolutional channel code is applied to all video parts (headers and coefficients).
2. *LDPC I-frames and macroblock headers*: an LDPC is used to code all headers and coefficients of I-frames, and macroblock headers of P and B-frames while the convolutional code is used to code the rest of headers and coefficients of P and B-frames.
3. *LDPC I/P-frames and macroblock headers*: an LDPC is used to code headers and coefficients of both I and P-frames and macroblocks headers of B-frames while the convolutional code is used only to coefficients of B-frames and the rest of headers.
4. *LDPC profile*: an LDPC is used to code all video parts (headers and coefficients).

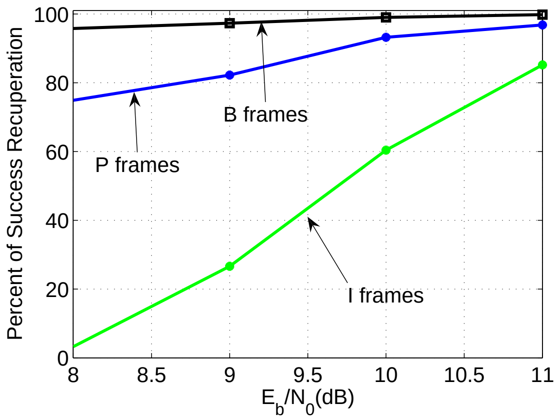

Fig. 5. Convolutional code: percentage of succesful reconstruction when individual frame coefficients are affected by AWGN.

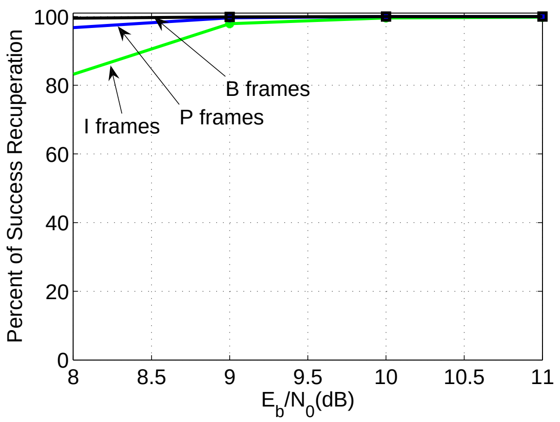

Fig. 6. LDPC: percentage of succesful reconstruction when individual frame coefficients are affected by AWGN.

We have compared these profiles considering the transmission of 10 frames of the same video used in the previous experiments. Figure 7 shows the percentage of successful decodings. It can be seen that the *convolutional profile* provides a very low performance while *LDPC profile* has the best behaviour in this specific channel. Regarding the hybrid profiles, we can see that the profile corresponding to *LDPC I/P-frames and macroblock headers* is able to achieve a percentage of successful recuperation larger than 90% for E_b/N_0 grather than roughly 9.5 dB. Also note that the improvement obtained when we use the profile corresponding to *LDPC I-frames and macroblock headers*, although it is not as important as with the other profiles. Note that the results presented in this figure indicates the perfect recuperation of all frames and, as a consequence, the correct recuperation of the video.

We have also evaluated the frame quality using the following expression

$$Quality = 4/6PSNR(Y) + 1/6(PSNR(C_b) + PSNR(C_r)) \quad (1)$$

where Y, C_b and C_r represent the luminance, blue crominance and red crominance, respectively. The

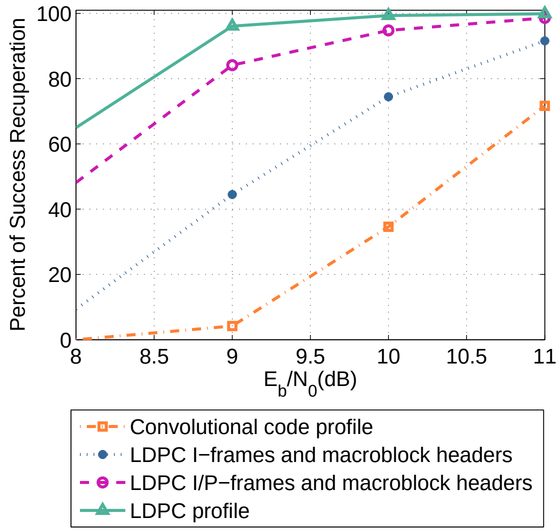

Fig. 7. Profiles comparison: success decoding percentage

factors 4/6 and 1/6 are included to weight each component respect to the sampling pattern (there are 6 sample for each pixel: four Y samples, one C_b sample and one C_r sample). PSNR is the *Peak Signal Noise Rate* between the original component and the recovered one. We can see that the *LDPC I/P-frames and macroblock headers* profile achieves a quality near to the obtained with the *LDPC profile* for all E_b/N_0.

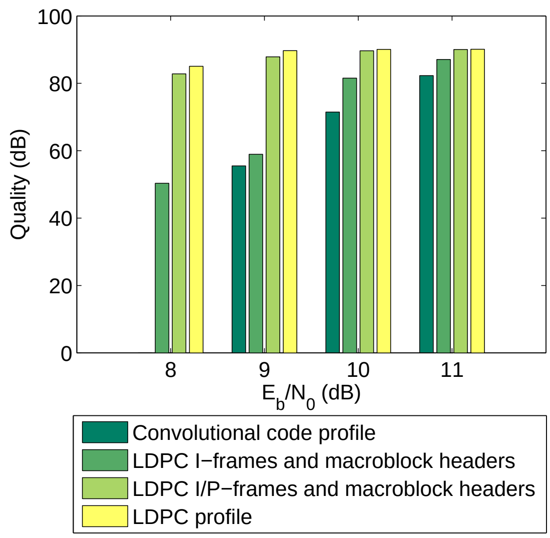

Fig. 8. Fig: Image Quality of decoded images of the studied profiles.

VI. Conclusions and future work

This paper studies the performance of transmitting videos coded with H.264/AVC over IEEE 802.11p with a hybrid channel encoding configuration. The results show that not all parts in video file has the same importance in order to obtain a good quality. In particular, I and P- frames and macroblock headers are essential to obtain an adequate percent of success at the decoding stage. For this reason, we have proposed to protect this critical information by using a powerful coder scheme like a

LDPC, while the the rest of data is protected using a convolutional code.

This is a preliminary study and several questions still open. Future work includes the performance evaluation with vehicular channels like those described in [10], the creation of new hybrid profiles and the utilization of new channel encoders adapted to vehicular environments. Moreover, other transmission schemes will be tested, like Multiple-Input Multiple-Output (MIMO) systems, in order to take advantage from spatial and temporal diversity.

Acknowledgments

This work has been funded by Xunta de Galicia, Ministerio de Ciencia e Innovación of Spain under grants with numbers 10TIC105003PR, TEC2010-19545-C04-01, TEC2010-11776-E.

Referencias

[1] IEEE, *IEEE for - and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications. Amendment 6: Wireless Access in Vehicular Environments*, July 2010.

[2] IEEE, *IEEE for - and exchange between systems- Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 2007.

[3] D. Jiang, L. Delgrossi, IEEE 802.11p: towards an international standard for wireless access in vehicular environments, *in Proceedings of VTC Spring*, Singapore, May 2008.

[4] R. Uzcátegui and G. Acosta-Marum. WAVE: A Tutorial. IEEE Communications Magazine, vol. 47, no. 5, pp. 126-133, 2009.

[5] *ITU-T and ISO/IEC JTC 1: Advanced Video Coding for Generic Audiovisual Services.* ITU-T Rec. H.264/AVC and ISO/IEC 14496, 2003.

[6] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Transaction Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003.

[7] E. Masala, N. Kawaguchi, K. Takeda, J. C. De Martin, "Performance Evaluation of H. 264 Video Streaming over Inter-Vehicular 802.11 Ad Hoc Networks", IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2005), vol. 3, pp. 1936–1940, 2005.

[8] R. G. Gallager, Low Density Parity-Check Codes, *MIT Press*, Cambridge, MA, 1963.

[9] B. Vasudev, K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, Springer, 1997.

[10] Fernández-Caramés T. M., Gonzáez-López M., Escudero C. J., Castedo, L., chapther title "Rapid Prototyping for Evaluating Vehicular Communications", Advanced Applications of Rapid Prototyping Technology in Modern Engineering, edited by Muhammad Enamul Hoque, IN-TECH, September 2011.

[11] A. Goldsmith, *Wireless Communications*, Cambridge University Press, New York, 2005.

[12] M. Reisslein, L. karam, P. Seeling, F. H.P Fitzek, T. K. Madsen, "Video Trace Library", http://trace.eas.asu.edu/yuv/index.html, 2003.

[13] *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, Reference Software to Committee Draft. JVT-F100 JM17.2, 2010. Available on-line at http://iphome.hhi.de/suehring/tml/.

[14] J. Hehn, "LDPC codes and convolutional codes with equal structural delay: a comparison", *IEEE Transactions on Communications*, vol. 57, Issue 6, pp. 1683–1692, 2009.