Improving Performance of H.264/AVC Transmissions over Vehicular Networks

Ismael Rozas-Ramallal,
Tiago M. Fernández-Caramés,
Adriana Dapena
Facultad de Informática
Universidade da Coruña,
Campus de Elviña,
15071, A Coruña, Spain
Email: ismael.rozas@udc.es

Pedro A. Cuenca-Castillo
Instituto de Investigación en Informática de Albacete,
Universidad de Castilla-La Mancha,
Campus Universitario,
02071, Albacete, Spain
Email: Pedro.Cuenca@uclm.es

***Abstract*—This paper evaluates the performance of H.264/AVC transmissions over IEEE 802.11p –a standard explicitly designed to perform vehicular communications– and proposes two different strategies for improving performance. The first one consists in substituting the convolutional codes used in IEEE 802.11p with Low-Density Parity-Check (LDPC) codes. The second strategy consists in adapting the transmission power by taking into account the picture type (I/P/B picture). Experimental simulation results show that the utilization of these two methods allows us to reduce GOP (Group Of Pictures) losses and improves video quality without increasing computational requirements. The evaluation has been carried out using a testbed that integrates the H.264/AVC JM encoder, an IEEE 802.11p transceiver and an FPGA-based channel emulator that implements vehicular channel models based on a measurement campaign performed in 2006 in Atlanta, Georgia.**

Keywords: H.264/AVC, IEEE 802.11p, vehicular networks, FPGA-based testbed

I. Introduction

Vehicular communications is one of the topics that has recently attracted a lot of attention to the field of ITS (Intelligent Transport Systems) to support services aimed at providing safety and non-safety applications. Such kind of wireless communications may be performed between moving vehicles (Vehicle-To-Vehicle, VTV or V2V) or from vehicles to infrastructure (Roadside-to-Vehicle, V2I or RTV) [1], [2].

IEEE 802.11p [3] is the best positioned standard to act as the reference for the PHYsical (PHY) and Medium Access Control (MAC) layers for safety applications. However, it is not clear which is the most suitable wireless standard for non-safety vehicular applications, being the most cited candidates the WiFi standards IEEE 802.11a/b/g and IEEE 802.16e. Furthermore, they can be narrowed down to only IEEE 802.11a, IEEE 802.11p and IEEE 802.16e, since vehicular communications will take place in the 5 GHz band: both US and European authorities have reserved spectrum for ITS at 5.9 GHz.

There are just a few publications dedicated to evaluate performance of transmissions of H.264/AVC over the vehicular communication standards, being the major contributions related to generic testbeds or theoretical developments. For example, in [4] the authors present results (in terms of packet loss, end-to-end delay and transmission jitter) of H.264/AVC coded video transmissions over mobile area networks (but not in vehicular scenarios) when using IEEE 802.11b interfaces. By using traffic shaping tools they showed that video data requires high Quality of Service (QoS) and that the error resilience and correction mechanisms in the coding standard H.264/AVC were completely ineffective. In [5], the performance of H.264/AVC video streaming was evaluated in vehicular environments using the IEEE 802.11b ad-hoc network protocol. The results obtained in such paper allow concluding that each vehicular scenario presents specific characteristics in terms of average link availability and Signal-to-Noise Ratio (SNR), which can be exploited to develop more efficient applications.

The main contribution of our paper is to evaluate the performance of H.264/AVC over realistic vehicular channels using an FPGA implementation. The FPGA emulator uses the vehicular channel models proposed in [6] and allows evaluating the performance in different vehicular environments without requiring to perform tests on real roads. Our studies are focused on improving the video transmissions that follow the H.264/AVC and IEEE 802.11p specifications, what lead us to implement the proposed schemes in a realistic network. We propose two strategies to improve performance. The first one consists in using Low-Density Parity-Check Codes (LDPCs) instead of the convolutional codes indicated by the IEEE 802.11p standard. The second strategy to improve the performance consists in adapting the power at transmission to the propagation conditions (i.e. the power losses due to the channel) and to the picture type (I/P/B pictures).

II. H.264/AVC Transmission over IEEE 802.11p

Figure 1 shows the layer architecture of the testbed developed to evaluate H.264/AVC transmission over IEEE 802.11p. In this testbed we use LDPC codes [7] and an adaptation of the power at transmission using the information sent by the receiver to the transmitter. Note that we focus on improving the PHY layer but optimization can be also done in other layers by utilizing resilience and concealment techniques.

978-3-901882-50-0

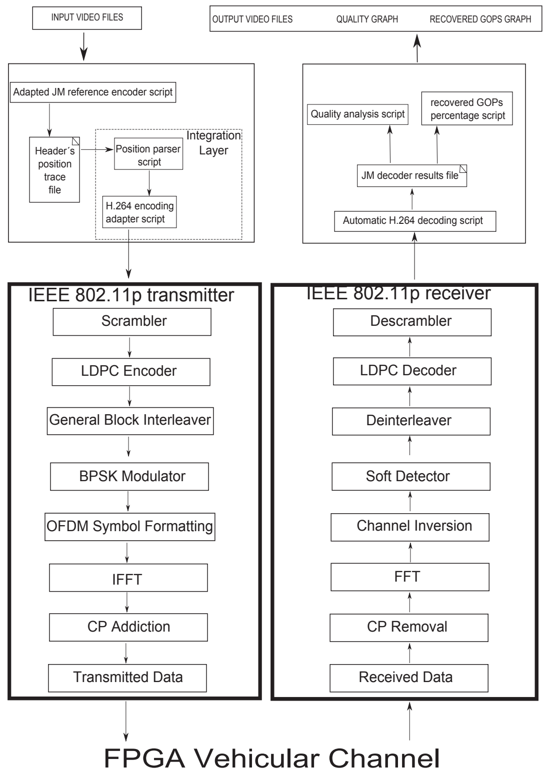

Fig. 1. H.264/AVC Transmission over IEEE 802.11p

A. H.264/AVC Integration Layer

H.264/AVC (or MPEG-4 Part 10) is currently one of the most commonly used formats for recording, compressing and distributing high definition video [8]. Our testbed uses the H.264/AVC JM Reference Software [9] developed in Visual C++, so it is relatively simple to modify the encoder and the decoder parameters and to parse the output file to extract the stream to be transmitted by the IEEE 802.11p transceiver. Among the predefined encoder profiles offered by the H.264/AVC JM Reference Software, we selected the encoder “extended profile”, because it is the specific profile for efficient video streaming. This profile has relatively high compression capability and some extra features for robustness to data losses [8]. The H.264/AVC JM Reference Software was configured to obtain Real-time Transport Protocol (RTP) packets.

It was needed to create a highly parameterized *H.264/AVC Integration Layer* to integrate the H.264/AVC encoder and the vehicular emulator (see [10]). The RTP packets obtained by the H.264/AVC JM encoder are treated by the *IEEE 802.11p Transmitter*.

B. IEEE 802.11p Transmitter

First, non-overlapped windows of 24 bits are passed to an scrambler that uses a 127-bit pseudo-random sequence. The scrambled data are coded by a LDPC encoder to combat the detrimental effects of the channel. The standard specifies an encoding rate of up to $1/2$, that means that the encoder takes as input one information bit and produces at its output two code bits. We have used a matrix $\mathbf{H}$ with dimensions 24×48 with three $1's$ per column.

The output of the LDPC code is the input of an interleaver: 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. After interleaving, the bits are Gray-mapped into Binary Phase Shift Keying Modulation (BPSK) symbols and placed into 48 out of a total of 64 subcarriers by performing the Inverse Fast Fourier Transform (IFFT) and adding a Cyclic Prefix (CP).

C. FPGA-based Vehicular Emulator

The data are then transmitted to an *FPGA-based Vehicular Emulator* [11] that implements channel models obtained after a measurement campaign carried out in the spring of 2006 in Atlanta, Georgia. Such channels were described initially in general terms in [6] and later, in more detail in [12]. In such documents the authors present channel models for six different 5.9 GHz high-speed environments that cover some of the most common situations where VTV and RTV communications may take place. In our experiments, we have used the model corresponding to RTV-Urban Canyon (speed of 120 km/h, Rayleigh distributed coefficients, fading Doppler of 994 Hz).

D. IEEE 802.11p Receiver

Finally, the *IEEE 802.11p Receiver* performs the following steps. First, the CP is removed and the FFT is applied to each OFDM symbol. Next, the channel is estimated in four subcarriers using the pilot symbol which are next used to obtain the channel frequency response for the rest of the subcarriers by linear interpolation. 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.

The LDPC decoder is based on a message-passing algorithm known as sum-product algorithm (SPA) [7] or, sometimes, as Belief Propagation [13]. Given a factor graph, the algorithm calculates the marginal distribution for each unobserved node conditioned by any nodes observed. In our implementation we use a log-domain SPA instead of the probability-domain SPA used in [7] but the use of log-likelihoods instead of probabilities allows us to substitute multiplications with additions, which are computationally more efficient. This algorithm allows us to determine each bit of the codeword based on the joint information of the variables and check nodes. If the word found is correct, the algorithm stops. In other case, it starts again having now as its initial variable node probabilities the values calculated in the last iteration. The number of iterations needed to decode each packet is a very important parameter since it ensures with a very high probability that the codeword is correct but, once reached the correct codeword, it is unnecessary to perform more iterations because the result will be the same. For this reason, our algorithm calculates the Hamming distance between the codeword obtained in two consecutive iterations and then decides whether is necessary to keep on iterating.

In the receiver, there exists an automatic MATLAB® decoding script, that automates the decoding process of the different videos integrating the H.264/AVC JM original decoder.

E. Power Adaptation Scheme

Most of the current wireless communication standards make use of feedback channels (usually limited in terms of throughput), which connect the receiver and the transmitter sides of the communications link to send periodically channel state information from the receiver to the transmitter. This information can be used, for instance, to adapt the transmission power or the modulation type. Since I-pictures are the most important frames for video recovering, it is reasonable to propose a procedure to vary the transmitter parameters to guarantee a good reception of such frames. In particular, we propose to adapt the E_b/N_0 in transmission to guarantee a desired E_b/N_0 at reception when packets corresponding to I-pictures are transmitted.

III. Performance Evaluation

Using the testbed described in Section II, we have carried out several simulation experiments. The evaluation has been performed considering four typical videos in QCIF format (176×144 pixels) [14]: Claire, Coastguard, Foreman and News. For each video, we have obtained a GOP formed by ten frames: one I-picture, three P-pictures and six B-pictures.

From the GOPs, we have obtained the RTP packets that have been transmitted using the IEEE 802.11p transceiver, varying the channel for each PHY packet (8064 bits). The results have been computed by averaging the results obtained in 50 independent experiments. We have performed four different tests. Three of the tests are oriented to determine the influence of the impact of each type of picture (I/P/B). In each individual test, only the packets corresponding to a specific picture type were transmitted through the vehicular channel while the rest of the packets were not perturbed. For the fourth test, all data streams were transmitted through the vehicular channel. All tests allow us to compare the performance of using LDPCs with the obtained using $1/2$-convolutional code included in the definition of IEEE 802.11p.

We have measured two parameters: the *recovered GOP percentage*, which is computed as the number of experiments where all the pictures in a GOP have been recovered respect to the number of experiments, and the *quality* of the recovered videos computed respect to the original videos using the following expression

$$Quality = \frac{4}{6}PSNR(Y) + \frac{1}{6}PSNR(C_b) + \frac{1}{6}PSNR(C_r)$$

where PSNR is the Peak Signal to Noise Rate (in dB) corresponding to the luminance (Y), blue chrominance (C_b) and red chrominance (C_r). The weight parameters correspond to the typical sample scheme 4:2:0. The results are plotted in terms of the E_b/N_0 at reception.

Figure 2 shows the recovered GOPs percentage (top) and the video quality (bottom). Comparing the results obtained using LDPC and the ones obtained using convolutional codes,

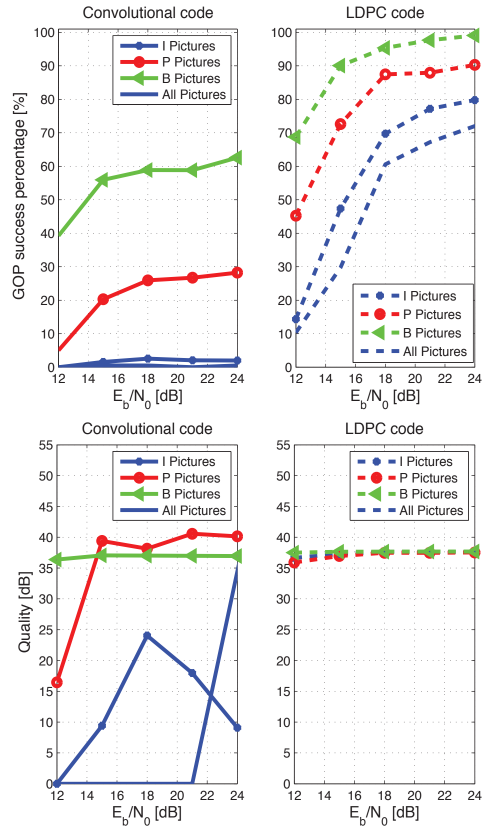

Fig. 2. Performance evaluation of H.264/AVC over IEEE 802.11p.

the advantage of using LDPC is clear for all E_b/N_0 values and tests.

In Figure 2, for LDPC code, the curve corresponding to "all pictures" indicates that the recovered GOP percentage is roughly equal to 50% when the E_b/N_0 at reception is equal to 18 dB. Also note that the picture quality is high for this E_b/N_0. To guarantee an E_b/N_0 value of 18 dB for all packets it is required a considerable increase in transmission power. Observing also Figure 2, we can conclude that the correct reception of the packets corresponding to I-pictures has a high impact in the performance, higher than the one related to P-pictures and B-pictures. As a consequence, we propose to adapt the power at transmission to guarantee that the E_b/N_0 at reception of I-pictures is equal to 18 dB. The advantage of the proposed scheme has been measured considering that the E_b/N_0 at reception can be 12 dB or 15 dB. We have computed the increase of E_b/N_0, recovered GOP percentage and picture quality, using the following expressions

$$\begin{aligned}
\Delta E_b/N_0 &= 1 - \frac{E_b/N_0 \ with \ LDPC}{E_b/N_0 \ with \ adaptive \ power} \\
\Delta GOPs &= 1 - \frac{GOPs \ with \ LDPC}{GOPs \ with \ adaptive \ power} \\
\Delta Quality &= 1 - \frac{Quality \ LDPC}{Quality \ with \ adaptive \ power} \quad (1)
\end{aligned}$$

The E_b/N_0 *with adaptive power* is computed by averaging the E_b/N_0 of all packets in the GOPs. For the four test videos,

Figure 3 plots the performance index increase versus E_b/N_0 increase. We can see that the E_b/N_0 increase varies from 6.03% (when de E_b/N_0 was 15 dB) to 20.09% (when de E_b/N_0 was 12 dB), which is considerably less than the values obtained when all packets are transmitted to obtain 18 dB at reception (16.66% for 15 dB and 33.33% for 12 dB). We also see that the most important difference with respect to the values obtained using LDPC codes (without power adaptation) appears in the recovered GOP percentage, while the quality improvement is relevant only when the E_b/N_0 is 12 dB.

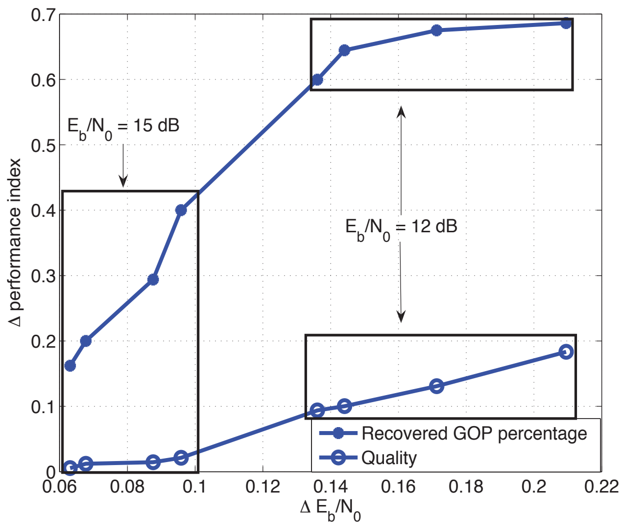

Fig. 3. LDPC with power adaptation: performance increase versus E_b/N_0 increase

A. Computational load

In order to compare the computational load associated with 1/2- convolutional and LDPC codes, we have measured the time required by the encoding and decoding processes to transmit 24 bits of the original video data (see Table I). The encoding and decoding times employed by the convolutional codes are constant, as it is the time required by the LDPC encoder. However, the time required by the LDPC decoder (using SPA) depends on the E_b/N_0. Table I summarizes the required times for the average number of iterations. Note that the time required by the convolutional decoder is similar to the one needed by the LDPC for 10 iterations, but our proposed scheme employs less iterations and, for this reason, the decoder time is considerably low.

	Encoding (msec)	Decoding (msec)
Convolutional	42.6	33.0
LDPC 12 dB (4.39 iterations)	49.1	13.79
LDPC 18 dB (2.54 iterations)	49.1	8.68
LDPC 24 dB (2.09 iterations)	49.1	7.20

TABLE I. MEASURED TIMES FOR THE ENCODING AND DECODING ALGORITHMS.

IV. CONCLUSIONS

Using a testbed based on real vehicular channels we have evaluated the performance of transmitting H.264/AVC coded videos over IEEE 802.11p transceivers. The experiment results show that the utilization of convolutional codes does not guarantee good quality and GOP reception. The experiments also show that the loss of RTP packets corresponding to I-frames produces a considerable degradation in the global performance and, for this reason, it is needed to include additional mechanisms to guarantee a correct transmission of these data. Thus, we have proposed two performance improvement strategies: the substitution of the convolutional codes with LDPCs and the adaptation of the transmission power.

ACKNOWLEDGMENT

This work has been founded by Xunta de Galicia (10TIC105003PR) and Spanish Plan Nacional (TEC2010-19545-C04-01).

REFERENCES

[1] Sichitiu M.L, *Inter-vehicle communication systems: a survey*, IEEE Communications Surveys and Tutorials, vol. 10, issue 2, pp. 88-105, 2008.

[2] Hubaux, J.P., Capkun, S., Jun Luo, *The security and privacy of smart vehicles*, IEEE Security and Privacy, vol. 2, issue 3, pp. 49–55, 2004.

[3] *Standard for Information Technology Telecommunications 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*, 2010

[4] Calafate, C.T. and Malumbres, M.P. and Manzoni, P., *Performance of H.264 compressed video streams over 802.11b based MANETs*, 24th International Conference on Distributed Computing Systems Workshops, 2004, pp. 776–781.

[5] Masala, E. and Kawaguchi, N. and Takeda, K. and De Martin, J. C., *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.

[6] Acosta-Marum G., Ingram M. A., *Six time- and frequency-selective empirical channel models for vehicular wireless LANs*, IEEE Vehicular Technology Magazine, vol. 2, issue 4, pp. 4–11, 2007.

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

[8] Richardson I. U., *The H.264 Advanced Video Compression Standard*, Wiley, 2nd. Edition, 2010.

[9] *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Reference Software to Committee Draft. JVT-F100 JM18.2*, 2010, [http://iphome.hhi.de/suehring/tml/].

[10] Rozas-Ramallal, I., Fernández-Caramés, T. M., Dapena A., García-Naya, J. A., "Evaluation of H.264/AVC over IEEE 802.11p Vehicular Networks", accepted in EURASIP Journal on Advances in Signal Processing.

[11] Fernández-Caramés T. M., González López M., Castedo L., *FPGA-based vehicular channel emulator for real-time performance evaluation of IEEE 802.11p transceivers*, EURASIP Journal on Wireless Communications and Networking, 2010.

[12] Acosta-Marum G., *Measurement, modeling and OFDM synchronization for the wideband mobile-to-mobile channel*, PhD thesis, Georgia Institute of Technology, 2007.

[13] Pearl J., *Reverend Bayes on inference engines: A distributed hierarchical approach*, *AAAI*, 1982.

[14] Reisslein M., Karam L., Seeling P., Madsen T. K., *Video Trace Library*, 2003, [http://trace.eas.asu.edu/yuv/index.html].