

RESEARCH

Open Access

# Evaluation of H.264/AVC over IEEE 802.11p vehicular networks

Ismael Rozas-Ramallal, Tiago M Fernández-Caramés, Adriana Dapena\* and José Antonio García-Naya

## Abstract

The capacity of vehicular networks to offer non-safety services, like infotainment applications or the exchange of multimedia information between vehicles, have attracted a great deal of attention to the field of Intelligent Transport Systems (ITS). In particular, in this article we focus our attention on IEEE 802.11p which defines enhancements to IEEE 802.11 required to support ITS applications. We present an FPGA-based testbed developed to evaluate H.264/AVC (Advanced Video Coding) video transmission over vehicular networks. The testbed covers some of the most common situations in vehicle-to-vehicle and roadside-to-vehicle communications and it is highly flexible, allowing the performance evaluation of different vehicular standard configurations. We also show several experimental results to illustrate the quality obtained when H.264/AVC encoded video is transmitted over IEEE 802.11p networks. The quality is measured considering two important parameters: the percentage of recovered group of pictures and the frame quality. In order to improve performance, we propose to substitute the convolutional channel encoder used in IEEE 802.11p for a low-density parity-check code encoder. In addition, we suggest a simple strategy to decide the optimum number of iterations needed to decode each packet received.

## 1 Introduction

Vehicular communications is one of the topics that has recently attracted a lot of attention to the field of Intelligent Transport Systems (ITS). Such kind of wireless communications may be performed between moving vehicles (vehicle-to-vehicle, VTV or V2V) or from vehicles-to-infrastructure (V2I or roadside-to-vehicle, RTV). They basically support services aimed at providing safety and non-safety applications. Vehicular safety applications require a fast exchange of messages in order to obtain a swift reaction from the car or the driver in dangerous situations, like sudden brakings or approaches to blind intersections. Non-safety applications do not suffer from so tight time constraints and include, for instance, the provision of wireless Internet access or infotainment services.

Regarding safety applications, IEEE 802.11p [1] is the best positioned standard to act as the reference for the PHYsical (PHY) and Medium Access Control (MAC) layers of vehicular communications. However, for non-safety vehicular applications, the selection of the most

suitable wireless access standard remains an open issue, being the most cited candidates the WiFi standards IEEE 802.11a/b/g, as well as WiMAX (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.

The IEEE 802.11a/p and IEEE 802.16e PHY layers are interfaces between the MAC layer and the wireless media which are based on the orthogonal frequency-division multiplexing (OFDM) modulation. The IEEE 802.11a/p PHY layer uses 64 subcarriers, while IEEE 802.16e can use up to 2.048. In both cases, the data subcarriers can be modulated with BPSK, QPSK, 16-QAM or 64-QAM depending upon the channel conditions. All three standards employ forward error correction mechanisms that can be implemented through convolutional coding with different coding rates, resulting in different transmission modes with multiple data rates.

Video streaming over vehicular networks is an attractive feature to many applications, such as emergency live video transmission, road-side video advertisement broadcasting, and inter-vehicle video conversation. Unfortunately, the performance of video streaming suffers from the delay

\*Correspondence: [adriana.dapena@udc.es](mailto:adriana.dapena@udc.es)  
Electronics and Systems Department, University of A Coruña, Campus de Elviña s/n, A Coruña, Spain

and packet loss incurred by the vehicular network characteristics. In this article, we evaluate the performance of H.264/AVC over IEEE 802.11p, since it has explicitly been designed to perform vehicular communications. The evaluation has been done using a transceiver developed by Grupo de Tecnología Electrónica y Comunicaciones (GTEC) from University of A Coruña, Spain [2].

One of the most important aspects to evaluate vehicular networks performance via simulation is to use an adequate channel model. The channel models implemented on our system are based on the work described in [3,4], which is mainly based on a measurement campaign carried out in the spring of 2006 in Atlanta, Georgia. From these measurements, the authors obtained six different channel models that cover some of the most common situations where VTV and RTV communications may take place:

- Urban canyons, with dense and tall buildings or slopes, where vehicles move at a speed of roughly 120 km/h. Although the speed may seem really high for an urban environment, note that the term “urban canyon” refers not only to places where there are tall buildings, but any other place where there are slopes or even large metallic billboards at both sides of the road that can cause a great deal of signal reflections.
- Suburban expressways, with moderately dense, low-story buildings, where the speed is approximately 140 km/h.
- Suburban surface streets, with moderately dense, low-story buildings, where the driving speed is 120 km/h.

Although we have obtained results for all scenarios, for the sake of this article length, we will only present the results obtained for two of them: an RTV Urban Canyon and a VTV Suburban Expressway.

The remainder of this article is organized as follows. Section 2 summarizes the state-of-the-art related to video transmission over wireless channels. Section 3 describes the vehicular communications testbed used for testing the performance of video transmission over IEEE 802.11p. Section 4 presents the low-density parity-check codes (LDPCs) we have implemented for replacing the convolutional codes included in regular IEEE 802.11p transceivers. Section 5 evaluates the performance and computational load of the vehicular testbed proposed when using convolutional codes or LDPCs in two vehicular environments. Finally, Section 6 is devoted to conclusions.

## 2 State-of-the-art

The utilization of testbeds to evaluate the performance of IEEE 802.11 technologies to transmit data over vehicles

has been addressed in previous work. In [5], a testbed is created to assess the capabilities of IEEE 802.11 standards in terms of Round Trip Time and packet losses. The results include an evaluation of the performance of VTV and RTV multi-hop communications based on the IEEE 802.11b standard, revealing that distance and line of sight communication are the two factors that mainly affect network communications. In the same way, in [6], it was demonstrated the feasibility of using IEEE 802.11b networks for transfer control protocol and user datagram protocol communications with moving cars. The authors conduct experiments in a friendly environment, with no obstacles or interferences from other radios and vehicles. Therefore, the results can be interpreted as a first approach to vehicular communications. There also exist other interesting testbeds that use IEEE 802.11p, in which this article is focused on. For instance, in [7], it was established a relationship between theoretical measures and the ones obtained with a testbed. According to the authors, their experiments lead to the first public data available from real measurements using NEC Linkbird-MX IEEE 802.11p cards, which showed that the results obtained were closely related to the theoretical figures. These results had a great impact, opening the possibility of evaluating IEEE 802.11p-based testbeds without requiring to perform measurements on the road.

Besides assessing the viability of using different transceivers in vehicular scenarios, the scientific community has studied how this kind of vehicular communications responds to data requirements. However, there are just a few publications dedicated to the standard IEEE 802.11p, being the major contributions related to generic testbeds or theoretical developments. For example, in [8] 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 service and that the error resilience and correction mechanisms in the coding standard H.264/AVC were completely ineffective. In [9], 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 a paper concludes 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. Furthermore, one of the most important contributions is [10], where the authors described the main characteristics of every vehicular channel, in most common parameters like received power or delay spread, extracting a great analysis of transmission reliability in this kind of communications.

Other standards have also been evaluated. For instance, in [11] an IEEE 802.16e (Mobile WiMAX) platform developed in NS-2 [12] was used to transmit video data coded with H.264/AVC, measuring jitter, delay, and peak signal-to-noise ratio (PSNR). The results obtained show that the average delay and jitter grow when the number of interfering nodes increases. Moreover, for high levels of video sample dynamism, the PSNR values obtained when coding with H.264/AVC are higher than the ones obtained by other codecs.

The results of all the articles previously mentioned allow us to conclude that the transmission of H.264/AVC coded videos over vehicular networks needs to be reinforced in order to improve its quality and overall performance. Several papers have addressed this issue in the last years. For example, in [13] different improvements are suggested for a buffering scheme related to forwarding. Two strategies are proposed: forwarding from the sender and forwarding from the receiver. The results show that forwarding from the receiver can improve video quality in most traffic situations. In the same way, a buffer overflow strategy based on deleting video packets with earlier playback deadline was proposed to mitigate this unavoidable fact. Other authors suggest improving the performance of video transmissions by incorporating more efficient transmission mechanisms. An example is given in [14], where it was evaluated the performance (in terms of BER and PSNR) when using LDPC codes. However, it is important to note that some of the LDPC coding rates used in [14] are not supported by IEEE 802.11p.

There also exist some unequal error protection systems that can transmit video data optimally by coding different sub-streams with different rates [15]. In such a paper, it was proposed an optimal bit allocation algorithm in order to allocate different LDPC channel coding rates to different sub-streams in a H.264/AVC coded video to achieve better end-to-end distortion than in an equal error protection (EEP) scheme.

Another approximation for obtaining better results consists in taking into account information about the channel. For instance, in [16], a cross-layer design was built with the objective of adapting the most relevant parameters in the application and physical layers to the transmission conditions. Thus, it was optimized the relationship between the average PSNR and the outage probability in a video broadcast service, obtaining the maximum possible video quality. This technique was studied in several papers [17-19] due to the numerous possibilities of configuration and the conceptualization offered by the cross-layer architecture.

The main contribution of this article is the evaluation of the performance of H.264/AVC over realistic vehicular channels using an FPGA implementation. The FPGA emulator uses the vehicular channel models proposed in

[3] and allows for evaluating the performance in different vehicular environments without requiring to perform tests on real roads. Therefore, this platform is a fast, flexible, and cheap solution to evaluate video performance in empirical scenarios. 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.

### 3 Vehicular communications testbed

The IEEE 802.11p standard is an amendment to the IEEE 802.11-2007 [20] that is technically compatible with the specifications given by ASTM E2213-03 [21], which addresses the challenges arising from providing wireless access in vehicular environments. Its PHY layer is very similar to that of the IEEE 802.11a, but there is an important difference: the 20 MHz bandwidth used by IEEE 802.11a is reduced to only 10 MHz. Such a difference is translated into a data transfer rate loss, but it doubles the OFDM symbol time length, which allows for reducing inter-symbol interference (ISI) and, therefore, supports the large delay spreads usually found in vehicular channels. A deep description of the IEEE 802.11p is beyond the scope of this article, but we encourage the interested reader to take a look at the excellent overviews given in [22,23].

In this section, we present a testbed that is able to transmit and receive H.264/AVC content in different vehicular scenarios. As it is shown in Figure 1, the system consists of three main parts: the H.264/AVC layer module, the IEEE 802.11p transceiver, and the channel emulator. While the transceiver was developed using technical software, the channel emulator is based on an FPGA. They are connected to each other through the PCI bus. This IEEE 802.11p testbed has been developed using as a reference the software transceiver and a hardware channel emulator previously developed by the Group of Electronic Technology and Communications of University of A Coruña [2].

#### 3.1 H.264/AVC layer model

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. H.264/AVC contains a number of new features allowing for compressing video more effectively than previous standards while providing more flexibility in a wide variety of network environments. 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 terms of rate-distortion with respect to existing standards [24,25].

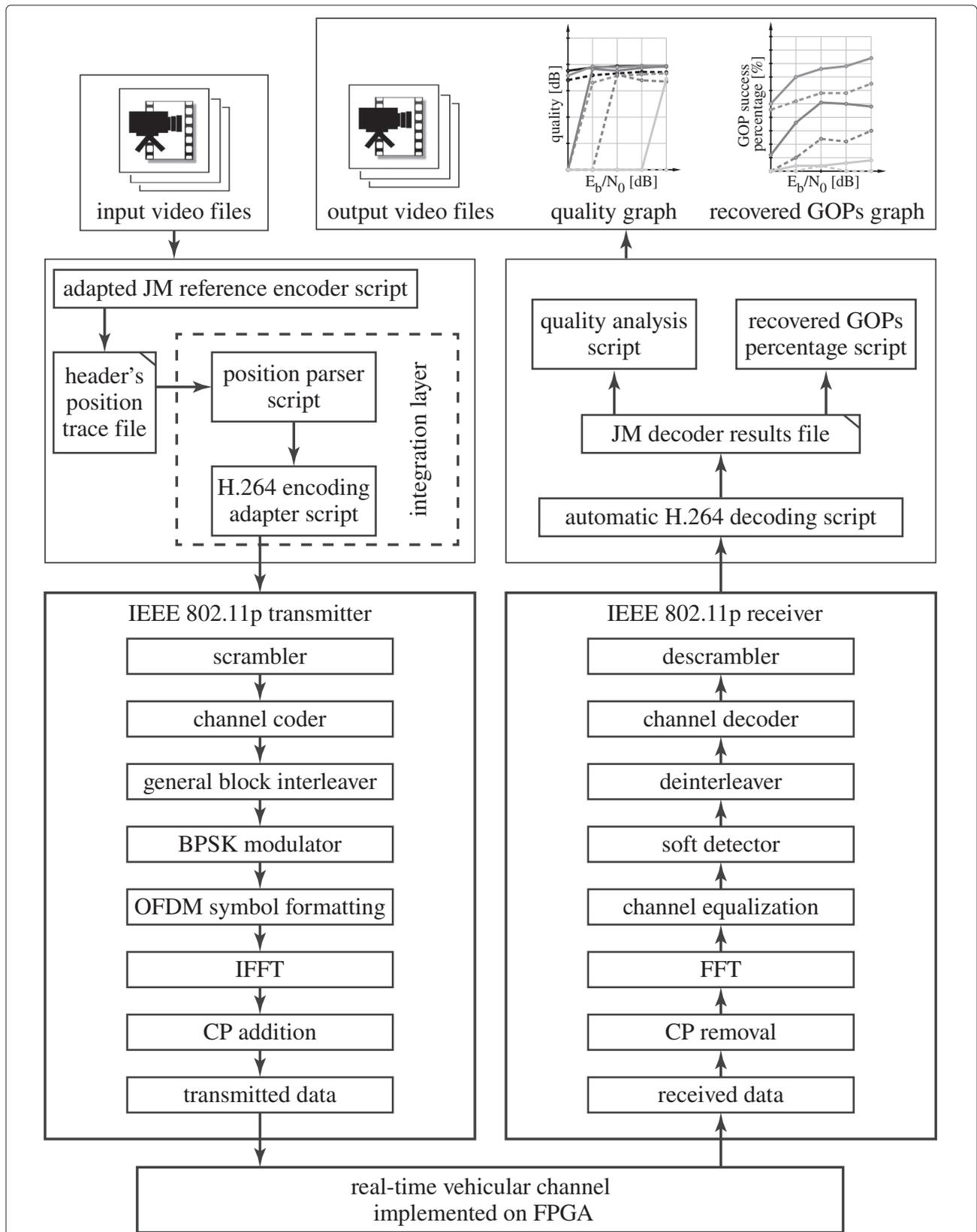


Figure 1 H.264/AVC integration modules.

H.264/AVC offers a simple and standardized structure for encapsulating compressed video and its related information (for more details see [25,26] and references therein). H.264/AVC syntax elements are encapsulated into Raw Byte Sequence Payloads (RBSP) and then into Network Abstraction Layer Units (NALU). The RBSP trailing bits are added in order to create a payload with an integral number of bytes. An RBSP is encapsulated into a NALU by adding a 1-byte header and inserting emulation prevention bytes. It is important to take into account that, like previous standards [27], H.264/AVC defines three frame types: Intra frames (I-frames), predictive frames (P-frames), and bidirectional frames (B-frames). A NALU contains data corresponding only to one type of frame.

H.264/AVC coded videos can be transmitted across networks using transport mechanisms like Real-time Transport Protocol (RTP). RTP defines a packet structure for real-time data transmissions that includes a type identifier and a sequence number used to re-order packets in time while decoding the data. RTP payload formats are defined for various multimedia encoders, including H.264/AVC. Each H.264/AVC NALU can be then inserted as a payload into an RTP packet.

The H.264/AVC encoder and decoder used in this article have been implemented by the Heinrich Hertz Institute in Fraunhofer and are called H.264/AVC JM 18.4 Reference Software [28], being the source code open to all developers. Such a source code was 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. Of all the profiles offered by the H.264/AVC JM Reference Software, we selected the “extended profile” encoder, because it is a profile for efficient video streaming. This profile has relatively high compression capability and some extra features for robustness against data losses [29].

In order to adapt the H.264/AVC packet stream to the requirements of our IEEE 802.11p transceiver, it was necessary to configure the following parameters of the encoder properly in order to extract the video stream:

- Since it is necessary to configure the encoder to obtain an output in RTP format, we selected *OutFileMode* = 1 and *PartitionMode* = 0. Also, we established *SliceMode* = 0 to guarantee that each RTP packet only contains bits from a frame type (I/P/B frames).
- In a previous empirical study, we obtained that the selection of a quantizer equal to 30 provided better video quality at the expense of a small video size increase. For this reason, we used *QPISlice* = 30, *QPPSlice* = 30, and *QPBSlice* = 30. For the remaining parameters, we used the default values: a

frame rate of 30 frames and YUV sampling format 4:2:0 (*FrameRate* = 30 and *YUVFormat* = 1).

Once selected all the parameters, it was necessary to create a highly flexible layer to integrate this optimized coded video and the vehicular emulator (see top layer in Figure 1). Such a layer is able to provide video data to the vehicular emulator in an intelligent way, being possible to choose between the different frames in the GOP or even the different headers or coefficients in a frame.

The first step to build the integration layer consists in modifying the original code of the H.264/AVC JM 18.4 Reference Software encoder in order to obtain the position of each kind of header and coefficient of the video architecture. It must be noted that the extraction of such values was possible after an intense inverse engineering process over the original JM code.

After this process, it was necessary to create an integration module to convert the original video stream to an adapted and personalized bit stream based on the header/coefficient positions. This process requires two scripts. The first one (“Position parser script” in Figure 1) is a MATLAB<sup>®</sup> script able to extract an array with the data to be transmitted. Such an extraction depends on the kind of frame or the type of selected header/coefficient to be transmitted. Then, a second script (“H.264 encoding adapter script” in Figure 1) processes the data array with the objective of generating the data stream file to be transmitted to the vehicular emulator. This second script lets us separate the different images or headers/coefficients in the video depending on the position of the bits in the H.264/AVC architecture. Thanks to this module we can isolate certain data in order to analyze their impact in GOP recovery percentage and video quality. In the same way, this script allows us to transmit different images using different channel encoding procedures.

Once the different videos are transmitted, it is necessary to decode them to obtain different quality of experience (QoE) measurements. In the receiver, there exists an automatic MATLAB<sup>®</sup> decoding script that automates the decoding process of the different videos. Such script invokes the original H.264/AVC JM 18.2 Reference Software decoder, configuring it automatically for the decoding process and making it possible to analyze a remarkable number of videos in a minimum amount of time. The output of the decoder for each video is stored in a different text file, which is parsed by two scripts to obtain the graphs that show the image quality and the percentage of recovered GOPs.

### 3.2 Transmitter

The transmitter performs the steps shown on the left-hand side of Figure 1. First, the data obtained from the H.264/AVC integration layer (RTP packets) are divided in

packets of 4032 bits. Each packet is scrambled, coded, and interleaved. The scrambler uses a 127-bit pseudo-random sequence. The scrambled data is then passed to a convolutional encoder, which introduces, in a controlled manner, certain redundancy into the bit stream. This redundancy allows the receiver to combat the detrimental effects of the channel and, hence, achieve reliable communications in spite of these effects. The standard specifies an encoding rate of up to 1/2, which means that the encoder takes as input a single information bit and produces at its output two coded bits. The convolutional code introduces redundant bits into the data stream through the use of linear shift registers. The generator polynomial is  $g_0 = 133$  and  $g_1 = 171$  in octal mode.

The output of the convolutional code is passed through an interleaver which 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. After interleaving, the bits are Gray-mapped into Binary Phase Shift Keying (BPSK) symbols and placed into 48 out of a total of 64 subcarriers. Four subcarriers are dedicated to pilot signals, DC subcarrier is not used, and the remaining subcarriers are dedicated to frequency guards. Each group of 64 subcarriers is modulated using OFDM, what implies that the Inverse Fast Fourier Transform (IFFT) is applied. Finally, a 1/4 cyclic prefix (CP) is added to prevent ISI.

### 3.3 Receiver

The receiver blocks are shown on the right-hand side of Figure 1. The first step, at the bottom of Figure 1, 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 Minimum Mean Square Error (MMSE) equalizer is employed. Finally, the equalized symbols are sent to a soft detector, whose outputs are deinterleaved, inverting the permutations performed in the transmitter. Finally, the decoding is carried out. For this last step, we use the Viterbi algorithm as implemented by Simulink<sup>®</sup>. Such a component decodes soft decisions, which are integers between 0 and  $2^b - 1$ , where  $b$  is the number of soft decision bits (0 is the most confident decision for logical zero and  $2^b - 1$  the most confident decision for logical one). Thus, the Hamming distance is used to calculate the branch metrics. The decoder works in *Terminated mode* (i.e., each input is treated independently) and the *Traceback depth* (i.e., the number of trellis branches used to construct each traceback path) is set to be equal to the

number of data subcarriers inside each OFDM symbol. Finally, the received video crosses the H.264/AVC integration layer to obtain the data needed for generating the desired graphs, in terms of quality and percentage of GOP recuperation.

### 3.4 FPGA-based channel-emulator

The channel models implemented on the FPGA-based vehicular emulator were obtained after a measurement campaign carried out in the Spring of 2006 in Atlanta, Georgia. Such channels were described initially in general terms in [3] and later, in more detail in [4]. In such documents, the authors present channel models for six different high-speed environments at 5.9 GHz that cover some of the most common situations in which VTV and RTV communications may take place.

The channel models can be grouped into three major scenarios: urban canyons (RTV-Urban Canyon, VTV-Urban Canyon Oncoming), expressways (VTV-Expressway Oncoming, RTV-Expressway, VTV-Expressway Same Direction With Wall), and suburban surface streets (RTV-Suburban Street). Urban canyon and suburban surface streets measurements assume a speed of 120 km/h, whereas the expressway measurements were made consistent with speeds of 140 km/h. Table 1 summarizes the main characteristics of the models.

The vehicular channel emulator has been developed using a BenADDA-IV kit from Nallatech [30] featuring:

- A Virtex-IV FPGA (XC4VSX35-10FF668).
- 4 MB of ZBT-RAM (Zero-Bus Turnaround RAM), two 14-bit Analog-to-Digital Converters (ADCs) able to sample up to 105 MS/s and two 14-bit DACs (digital-to-analog converters) that can run up to 160 MS/s.

**Table 1 Main characteristics of the vehicular channel models**

Vehicular channel	RTV-Urban canyon	VTV-Expressway oncoming
TX-RX distance (m)	100	300 – 400
Speed (km/h)	120	140
Path modulation (number of paths)	Rician (1) / Rayleigh (11)	Rician (1) Rayleigh (10)
Maximum delay spread (ns)	501	302
Rician K (dB)	7.5	-1.6
Overall K factor (dB)	6.7	-3.6
Maximum frequency shift (Hz)	720	1466
Maximum Doppler fading (Hz)	994	858
LOS Doppler (Hz)	654	1452

- It is able to operate either connected to a PC (via the PCI bus) or in stand-alone mode.

Regarding the implementation, it is worth noting that, traditionally, the development on the FPGA was carried out by using low-level description languages such as Very High-Speed Integrated Circuit Hardware Description Language (VHDL), which usually derive into slow development stages. Although in most cases VHDL is able to obtain resource-efficient FPGA designs, programming can become a cumbersome task that may consume a large amount of manpower. To avoid this problem, we have taken advantage of the Xilinx System Generator. It is pretty similar to Matlab's Simulink<sup>®</sup> and permits the use of high-level blocks, enabling complex FPGA designs easier and faster. Moreover, it allows the programmer to interact very easily with systems developed in MATLAB<sup>®</sup> and Simulink<sup>®</sup>, thus simplifying data exchange between a design running on the FPGA and a software implementation that is executed on a PC (in fact, for our tests we have run in MATLAB<sup>®</sup> and Simulink<sup>®</sup> the transceivers while the vehicular channel emulator was running on the FPGA). However, it must be noted that although rapid-prototyping tools like System Generator increase development speed, they usually produce non-optimized large designs that may not fit into the FPGA. Hence, for large designs, optimizations must be performed. A detailed description of the design and the optimizations applied to the vehicular emulator can be found in [2].

#### 4 IEEE 802.11p transceiver with LDPCs

Besides evaluating the convolutional codes included in IEEE 802.11p, in this article we also show the performance of the system when using LDPC codes [31]. LDPCs are a kind 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 [32]. 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 because it carries out an iterative procedure. In order to compare it directly with the convolutional code, we have used a 1/2 LDPC and a matrix  $\mathbf{H}$  with dimensions  $24 \times 48$  and 72 ones, three per column.

The creation of the matrix  $\mathbf{H}$  is straightforward: after fixing the number of ones for each column/row, they are pseudo-randomly spread throughout the matrix. It is interesting to note that it is possible to create improved codes for specific channels by using, for instance, EXtrinsic Information Transfer (EXIT) chart evolution [33].

The LDPC decoder is based on a message-passing algorithm known as Sum-Product Algorithm (SPA) [31] or, sometimes, as Belief Propagation [34]. 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 [31]. Both versions of SPA have similar decoding performance, but the use of log-likelihoods instead of probabilities allows us to substitute multiplications with additions, which are computationally more efficient. This algorithm allows for determining 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 from the values calculated in the last iteration as initial variable node probabilities.

The number of iterations of the LDPC decoder 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 it is necessary to keep on iterating.

#### 5 Experimental results

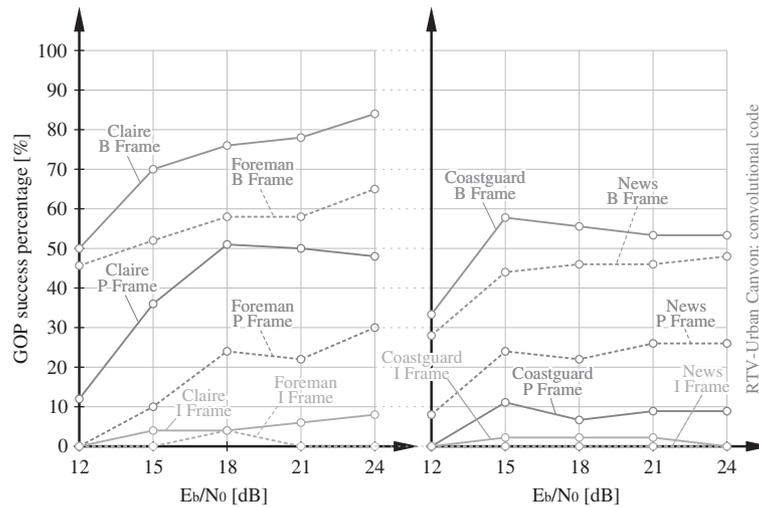
We have carried out several simulation experiments oriented to the evaluation of the performance obtained when videos coded with H.264/AVC are transmitted using our IEEE 802.11p vehicular testbed described in Section 3. The evaluation has been performed considering 10 frames of typical videos in QCIF format ( $176 \times 144$  pixels) [35]: Claire, Coastguard, Foreman, and News. For each video, the Group of Pictures (GOP) used in the simulations is formed by one I-frame, three P-frames, and six B-frames. Table 2 shows the quality of the coded videos obtained using the following expression

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

where PSNR is 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.

**Table 2 Quality of the H.264/AVC coded videos**

Video	Quality (dB)
Claire	39.5336
Coastguard	36.9489
Foreman	37.4593
News	37.1528



**Figure 2** Performance of H.264/AVC over IEEE 802.11p in "RTV-Urban Canyon": convolutional code.

The vehicular channels also added Gaussian noise resulting  $E_b/N_0$  values at reception that ranged between 12 and 24 dB. The results have been obtained by averaging 50 independent GOP transmissions in four different videos, varying the channel coefficients in each transmitted physical packet. The number of physical packets is different for each video: 13 packets in Foreman, 10 in News, 6 in Claire, and 13 in Coastguard.

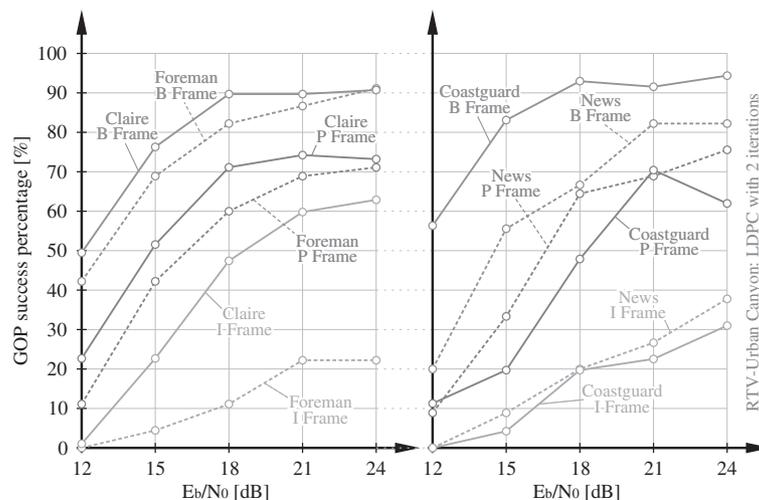
In order to obtain a first measure of the performance, we determined the probability of recovering all frames in a GOP with respect to the total number of GOPs transmitted. Furthermore, since H.264/AVC defines three frame types (as it was mentioned in Section 3.1), we decided to quantify the impact of the channel over the packets associated to each kind of frame. Thus, we have performed

three different experiments in which only the packets corresponding to I/B/P-frames are transmitted through the channel while the rest of the packets are not perturbed, therefore determining the specific impact of each one.

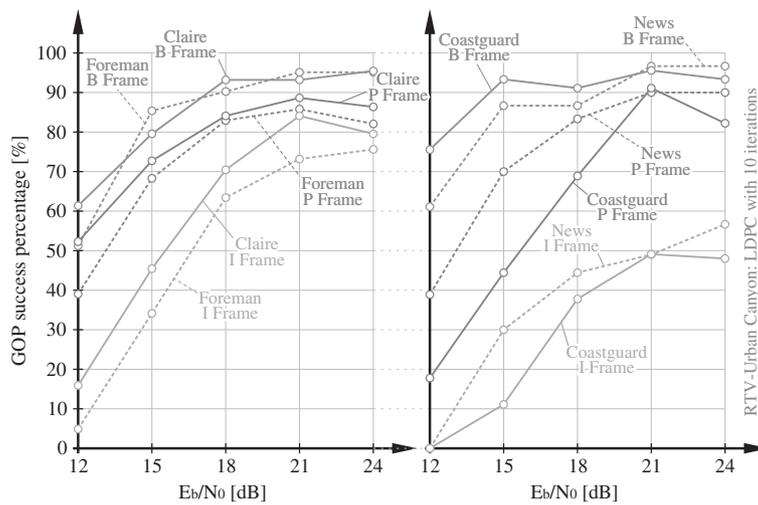
### 5.1 RTV-urban canyon oncoming transmissions

#### 5.1.1 GOP success probability

In this first set of experiments we have evaluated the performance of transmitting the videos over the vehicular channel RTV-Urban Canyon Oncoming. Figure 2 shows the percentage of recovered GOPs for different  $E_b/N_0$  values and for the four images selected when using convolutional codes. The legend indicates which frame type is affected by the channel while the rest remain unaltered.



**Figure 3** Performance of H.264/AVC over IEEE 802.11p in "RTV-Urban Canyon": LDPC with 2 iterations.



**Figure 4** Performance of H.264/AVC over IEEE 802.11p in "RTV-Urban Canyon": LDPC with 10 iterations.

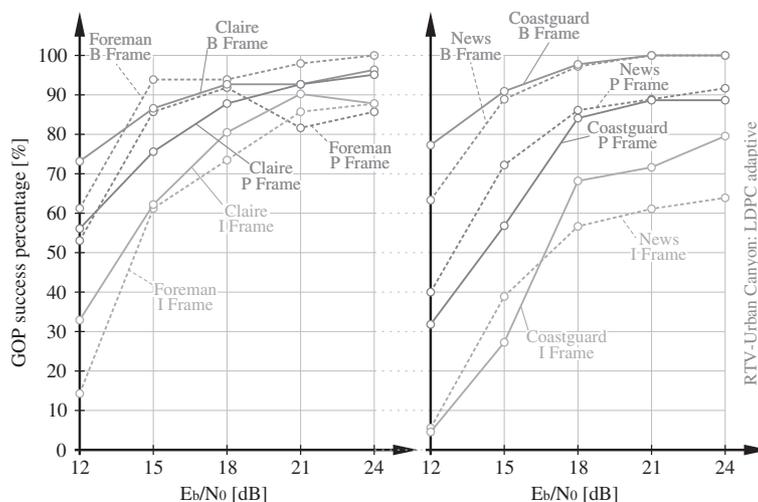
Figure 2 points out several important aspects about the performance of the IEEE 802.11p transceiver:

- In most experiments the convolutional code is not able to recover I-frames, which translates into the lost of almost all frames of the GOP.
- The best performance obtained by the convolutional code occurs when only B-frames are transmitted through the channel. However, note that, in general, the percentage of the recovered GOP is less than 80% for all test videos (it is only obtained more than 80% for Claire with  $E_b/N_0 = 24$  dB).

At the sight of the results it can be stated that the performance of IEEE 802.11p is not acceptable to transmit

videos coded with H.264/AVC in the selected vehicular channel. Due to that reason we have evaluated the performance of the transceiver while using LDPC codes instead of convolutional codes. Figure 3 shows the percentage of recovered GOPs for different  $E_b/N_0$  values when the LDPC decoding algorithm performs two iterations, while Figure 4 exhibits the results obtained for ten iterations. Comparing both figures, it can be concluded that increasing the number of iterations improves the performance, especially for I-frames. Furthermore, it can clearly be observed the improvement obtained with respect to convolutional codes for all kinds of frames.

Figure 5 shows the results obtained when the number of iterations of the decoding algorithm is determined using the decision criterion mentioned at the end of Section 4:



**Figure 5** Performance of H.264/AVC over IEEE 802.11p in "RTV-Urban Canyon": LDPC with a variable number iterations.

the LDPC decoder stops when the distance between two consecutive outputs is zero or when the number of iterations is greater than 30. Note that using this adaptive approach the results are clearly better than those obtained using 10 iterations. For instance, if we look at the  $E_b/N_0$  value where 50% of GOP success probability is achieved for the worst case (I-frames), we see that:

- For the video “Claire”, 15.8 dB are required for an LDPC with 10 iterations, while only 14 dB for a variable number of iterations.
- For “Coastguard”, 21 dB are needed for an LDPC with 10 iterations and 16.4 dB for a variable number of iterations.
- For “Foreman”, it is needed 17 dB for an LDPC with 10 iterations and 14 dB for a variable number of iterations.
- For “News”, it is needed 21 dB for an LDPC with 10 iterations and 16.4 dB for a variable number of iterations.

These results allow us to conclude that, on average, the adaptive approach requires about 3 dB less than using 10 iterations. This fact has an important impact into the transmission power: the variable approach requires only half of the power for reaching 50% of GOP success probability.

Finally, Figure 6 shows the success probability when all the packets are transmitted through the RTV-Urban Canyon Oncoming. We can see that the utilization of LDPCs improves dramatically the performance obtained by the convolutional codes which were unable to recover correctly all the frames of the GOP in the selected scenarios, even for high  $E_b/N_0$  values. Note also that the proposed approach with a variable number of iterations

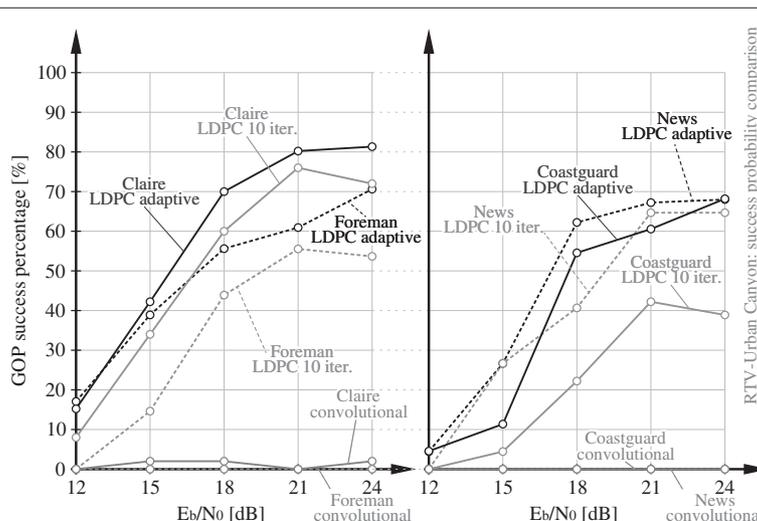
**Table 3 Number of iterations of the LDPC decoder**

Iterations	Occurrences (%)	
	RTV-Urban Canyon	VTV-Expressway Oncoming
1 iteration	0	0
2 iterations	93.9254	94.4907
3 iterations	2.7151	2.7315
4 iterations	0.7363	0.7407
5 iterations	0.4142	0.4167
6 iterations	0.2301	0.2315
7 iterations	0.2301	0.2315
8–29 iterations	less than 0.1	less than 0.1
30 iterations	0.5983	0.6019

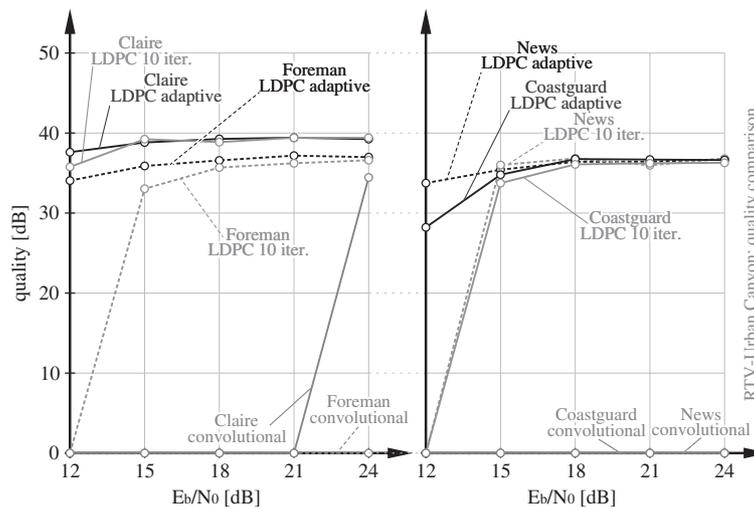
obtains the best performance for any video and  $E_b/N_0$  value.

The smallest difference obtained between the adaptive and the 10-iteration schemes occurs for the video “Claire”. For that video and vehicular scenario, Table 3 shows the number of iterations actually consumed. We can see that, in most cases, only 2 iterations have been required. There exists a residual percentage of GOPs that uses 30 iterations, corresponding to the packets that the LDPC is not able to recover. It is important to remark that the performance difference between the variable and the 10-iteration approach is derived from the small fraction of situations in which more than 10 iterations are required (see Table 3).

Figure 7 compares the performance in terms of frame quality obtained by averaging the quality measure in Equation (1) between of the recovered videos with respect to the original transmitted videos. Such a figure corroborates the results obtained in Section 5.1.1 and exposes the



**Figure 6 Performance of H.264/AVC over IEEE 802.11p in “RTV-Urban Canyon”: Success probability comparison.**



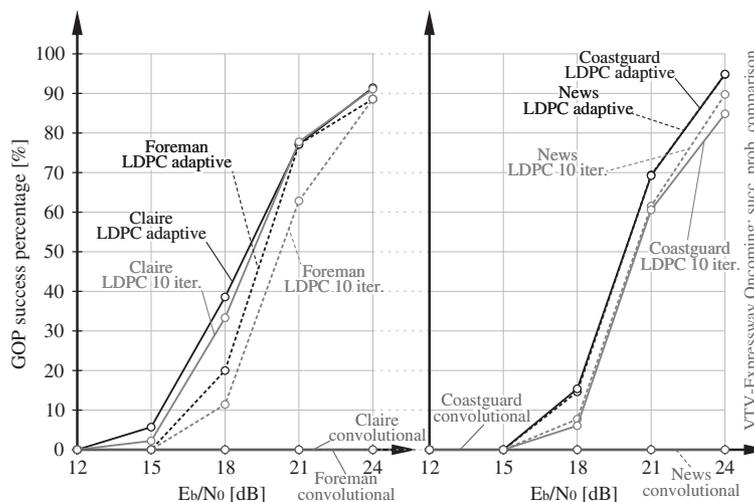
**Figure 7** Performance of H.264/AVC over IEEE 802.11p in "RTV-Urban Canyon": Quality comparison.

remarkable improvement obtained by using LDPCs compared to convolutional codes. Comparing the curves and the maximum values given in Table 2, we conclude that for an adequate  $E_b/N_0$ , it is possible to recover the videos without losing quality. In particular, LDPCs with a variable number of iterations achieve the best performance for almost every  $E_b/N_0$  value and video.

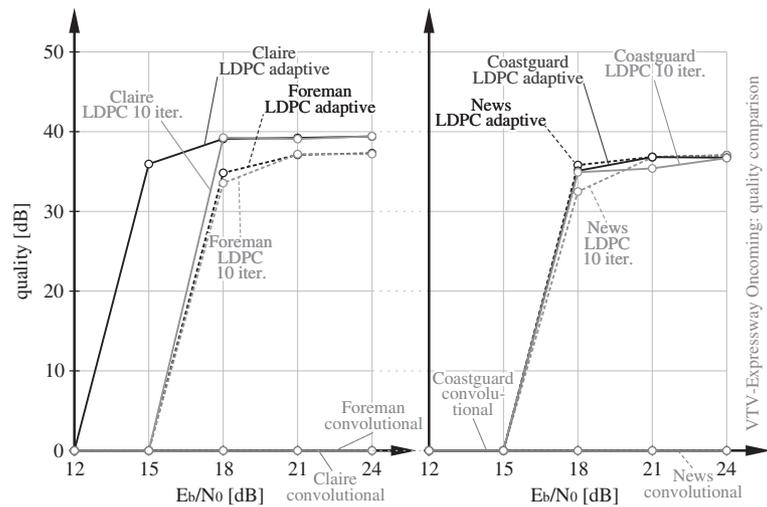
### 5.2 VTV-expressway oncoming

In order to assess the H.264/AVC-based system proposed in another common vehicular scenario, we evaluated also the performance considering now VTV-Expressway Oncoming. Figure 8 shows the results obtained when all the packets are transmitted. We can see again that the

utilization of LDPCs improves the performance obtained by the convolutional codes. Moreover, the utilization of a variable number of iterations allows for obtaining a remarkable improvement for any video and  $E_b/N_0$  value. Comparing these results to the ones obtained for the RTV-Urban Canyon Oncoming transmissions, it can be concluded that the channel in Expressway introduces more degradation. However, the utilization of LDPCs with a variable number of iterations obtains more than 50% of the recovered GOPs. Figure 9 shows that the utilization of this method allows for improving frame quality. The number of iterations presented in Table 3 shows that, like in RTV-Urban Canyon Oncoming, the adaptive LDPC approach performs a very reduced number of iterations.



**Figure 8** Performance of H.264/AVC over IEEE 802.11p in "VTV-Expressway Oncoming": Success probability comparison.



**Figure 9** Performance of H.264/AVC over IEEE 802.11p "VTV-Expressway Oncoming": Quality comparison.

### 5.3 Computational complexity

In order to compare the computational load associated with convolutional and LDPC coding, we have measured the time required by the encoding and decoding processes. Table 4 presents the execution time required by the convolutional and LDPC codes when using 2, 10, and a variable number of iterations. Note that the LDPC decoder carries out two steps: initialization and iteration. The initialization lasts 0.3 ms and each iteration lasts 3.3 ms. In the case of the adaptive LDPC, the decoder time depends on the number of iterations. The averaged number of iterations obtained for 2240 packets of 48 bits has been 2.30 iterations for RTV-Urban Canyon Oncoming, and 2.31 iterations for VTV-Expressway Oncoming. Notice that the number of iterations depends on the channel and the PHY layer, but not on the video (for this reason we only show the results for the video "News").

The results given in Table 4 show that the computational load of the encoders is similar, but they differ greatly in the decoders load: Convolutional decoder requires the same time as the LDPC decoder with 10 iterations. Note that

this time is reduced considerably when the number of iterations is determined for each packet. Considering these results, the GOP success probability and the image quality results shown in previous sections, we can conclude that the LDPC with variable number of iterations is an interesting alternative for providing video services in vehicular environments.

### 6 Conclusions and future work

This article studies the performance of transmitting videos coded with H.264/AVC over IEEE 802.11p. It was developed a testbed that is able to transmit and receive H.264/AVC coded videos over different vehicular empirical scenarios. The system consists of three parts: the H.264/AVC encoder/decoder, the IEEE 802.11p transceiver, and the FPGA channel emulator. Using this testbed, we have evaluated the performance of H.264/AVC video data transmitted over IEEE 802.11p in RTV-Urban Canyon and VTV-Expressway Oncoming channels. The results show a poor performance in both recovered GOP percentage and frame quality. For this reason, we replaced the convolutional coding stage with another one based on LDPC codes, showing a dramatic improvement. Since LDPC decoding uses an iterative algorithm adapted to the relevance of the video data to transmit, we have proposed an LDPC scheme that determines the required number of iterations each time a new packet is received. This new scheme provides very promising results compared to rigid channel encoding schemes. Finally, we have quantified the computational load of each coding stage, concluding that the adaptive LDPC coding variant has better performance and, globally, is much more appropriate for channel encoding in vehicular video transmissions.

**Table 4** Measured times for encoding/decoding algorithms

Algorithm	Encoding time (ms)	Decoding time (ms)
Convolutional	42.6	33.0
LDPC with 2 iterations	49.1	6.9
LDPC with 10 iterations	49.1	33.3
LDPC adaptive (RTV-Urban Canyon)	49.1	7.89
LDPC adaptive (VTV-Expressway Oncoming)	49.1	7.91

Further work will deal with evaluating the performance considering a more complete set of videos and different combinations of coding schemes. In addition, more robust coding schemes will be considered to transmit I-frames. Finally, it is reasonable to think that the performance could be improved by choosing the LDPC matrix depending on the specific vehicular channel.

#### Competing interests

The authors declare that they have no competing interests.

#### Acknowledgements

This study was funded by Xunta de Galicia, Ministerio de Ciencia e Innovación of Spain, and FEDER funds of the European Union under grants with numbers 10TIC105003PR, TEC2010-19545-C04-01, and CSD2008-00010.

Received: 19 December 2012 Accepted: 7 March 2013

Published: 11 April 2013

#### References

- 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)
- TM Fernández-Caramés, M González-López, L Castedo, FPGA-based vehicular channel emulator for real-time performance evaluation of IEEE 802.11p transceivers. *EURASIP J. Wireless Commun. Netw.* **42** (2010)
- G Acosta-Marum, MA Ingram, Six time- and frequency-selective empirical channel models for vehicular wireless LANs. *IEEE Veh. Technol. Mag.* **2**(4), 2134–2138 (2007)
- G Acosta-Marum, Measurement, modelling and OFDM synchronization for the wideband mobile-to-mobile channel. PhD thesis, Georgia Institute of Technology (2007)
- M Jerbi, P Marlier, SM Senouci, in *Proceeding of the IEEE International Conference on Mobile Adhoc and Sensor Systems*. Experimental assessment of V2V and I2V communications, (Pisa (Italy), 2007), pp. 1–6
- R Gass, J Scott, C Diot, in *Workshop on Mobile Computing Systems and Applications (WMCSA)*. Measurements of in-motion 802.11 networking (Washington, USA, 2006), pp. 69–74
- C Cicconetti, F Galeassi, R Mambrini, IEEE 802.11p: laboratory measurements and analysis. *Telecommunication Business Unit - Intecs S.P.A* (2011)
- C Calafate, M Malumbres, P Manzoni, in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCS)*. Performance of H.264 compressed video streams over 802.11b based MANETs (Tokyo, Japan, 2004), pp. 776–781
- E Masala, N Kawaguchi, K Takeda, JC De Martin, in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, vol. 3. Performance evaluation of H.264 video streaming over inter-vehicular 802.11 ad hoc networks (Berlin, Germany, 2005), pp. 1936–1940
- CF Mecklenbrauker, A Molisch, J Karedal, F Tuvfesson, A Paier, L Bernado, T Zemen, O Klemp, N Czink, Vehicular channel characterization and its implications for wireless system design and performance. *Proc. IEEE.* **99**(7), 1189–1212 (2011)
- J Casasepere, P Sanchez, T Villameriel, J Del Ser, in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. Performance evaluation of H.264/MPEG-4 scalable video coding over IEEE 802.16e networks (Bilbao, Spain, 2009), pp. 1–6
- The Network Simulator - NS-2 (2007). [http://www.isi.edu/nsnam/ns/]
- F Xie, KA Hua, W Wang, YH Ho, in *Proceedings of the IEEE 66th Vehicular Technology Conference (VTC)*. Performance study of live video streaming over highway vehicular ad hoc networks (Baltimore, USA, 2007), pp. 2121–2125
- H Li, Y Zhong, in *2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, vol. 1. A novel method for H.264 video transmission using LDPC codes over high BER wireless network (Shenzhen, China, 2009), pp. 464–467
- Y Liu, S Qaisar, H Radha, A Men, in *43rd Annual Conference on Information Sciences and Systems (CISS)*. On unequal error protection with low density parity check codes in scalable video coding (Baltimore, USA, 2009), pp. 793–798
- A Cagdas, MO Sunay, in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. Improving the performance of wireless H.264 video broadcasting through a cross-layer design (Bilbao, Spain, 2009), pp. 1–6
- M Van Der Schaar, D Sai Shankar, Cross-layer wireless multimedia transmission: challenges, principles and new paradigms. *Wireless Commun.* **12**(4), 50–58 (2005)
- I Bajic, Efficient cross-layer error control for wireless video multicast. *IEEE Trans. Broadcasting.* **53**, 276–285 (2007)
- J Qiu, G Zhu, in *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. An adaptive cross-layer video transmission scheme over wireless channels (Hong Kong, China, 2005), pp. 721–724
- 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 (2007)
- Standard specification for telecommunications and information exchange between roadside and vehicle systems—5 GHz band dedicated short range communications (DSRC), medium access control and physical layer specifications (2003)
- D Jiang, L Delgrossi, in *IEEE Proceedings of Vehicular Technology Magazine (VTC)*. IEEE 802.11p: towards an international standard for wireless access in vehicular environments (Marina Bay, Singapore, 2008), pp. 2036–2040
- R Uzcátegui, G Acosta-Marum, WAVE: a tutorial. *IEEE Commun. Mag.* **47**, 126–133 (2009)
- 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)
- T Wiegand, GJ Sullivan, Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003)
- GJ Sullivan, T Wiegand, Video Compression - From Concepts to the H.264/AVC Standard. *Proceedings of the IEEE.* **93**(1), 18–31 (2005)
- B Vasudev, K Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*. (Springer, New York, 1997)
- H.264/AVC JM Software Coordination. [http://iphome.hhi.de/suehring/ttml/]
- IEG Richardson, *The H.264 Advanced Video Compression Standard*. (Wiley, New York, 2003)
- Nallatech. [http://nslam.isi.edu/nslam/index.php/User\_Information]
- RG Gallager, *Low Density Parity-Check Codes*. (MIT Press, Cambridge, 1963)
- A Goldsmith, *Wireless Communications*. (Cambridge University Press, New York, 2005)
- M Gonzalez-Lopez, F Vazquez-Araujo, L Castedo, J Garcia-Frias, in *5th International Symposium on Turbo Codes and Related Topics*. Interleave-division multiple access (IDMA) using low-rate layered LDGM codes (Lausanne, Switzerland, 2008), pp. 315–320
- J Pearl, in *Association for the Advancement of Artificial Intelligence (AAAI)*. Reverend bayes on inference engines: a distributed hierarchical approach, (1982)
- M Reisslein, L Karam, P Seeling, TK Madsen, Video trace library (2003). [http://trace.eas.asu.edu/yuv/index.html]

doi:10.1186/1687-6180-2013-77

Cite this article as: Rozas-Ramallal et al.: Evaluation of H.264/AVC over IEEE 802.11p vehicular networks. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:77.