RESEARCH

Open Access

CrossMark

A protection scheme for multimedia packet streams in bursty packet loss networks based on small block low-density parity-check codes

Filippo Casu[*], Julián Cabrera, Fernando Jaureguizar and Narciso García

Abstract

This paper proposes an enhanced forward error correction (FEC) scheme based on small block low-density parity-check (LDPC) codes to protect real-time packetized multimedia streams in bursty channels. The use of LDPC codes is typically addressed for channels where losses are uniformly distributed (memoryless channels) and for large information blocks. This work suggests the use of this type of FEC codes at the application layer, in bursty channels (e.g., Internet protocol (IP)-based networks) and for real-time scenarios that require low transmission latency. To fulfil these constraints, the appropriate configuration parameters of an LDPC scheme have been determined using small blocks of information and adapting the FEC code to be capable of recovering packet losses in bursty environments. This purpose is achieved in two steps. The first step is performed by an algorithm that estimates the recovery capability of a given LDPC code in a burst packet loss network. The second step is the optimization of the code: an algorithm optimizes the parity matrix structure in terms of recovery capability against the specific behavior of the channel with memory. Experimental results have been obtained in a simulated transmission channel to show that the optimized LDPC matrices generate a more robust protection scheme against bursty packet losses for small information blocks.

Keywords: Multimedia communications; Time-sensitive services; Real time; Forward error correction; LDPC; LDGM; Bursty channels; Small block codes

1 Introduction

Multimedia communications are moving toward IP networks, since they are more efficient than circuit-switching channels. The most popular real-time applications that deliver services using Internet protocol (IP) networks are represented by voice over IP (VOIP), television over IP, and video conference [1, 2].

Nevertheless, IP networks have some weak points (e.g., unreliable packets delivery and delay control problems) that might damage the media content. For this reason, research is centring its efforts on developing efficient and robust systems against delays and channel errors.

Specifically, errors and losses can occur at different levels: data link layer and higher layers [3]. In this sense, those at the data link layer are originated by low values of signal-to-noise ratio (bits arrive at the receiver either correctly

or erroneously). In this layer, error detection and error correction are possible with the use of forward error correction (FEC) codes. On the other hand, at higher layers, losses and errors also occur due to sporadic impulse noise or congestion of the network routers, leading to bursts of packet losses (erasure channels). In this case, the use of application layer FEC (AL-FEC) [4, 5] can prevent these errors.

In this paper, we focus on packet delivery problems at the application layer, regardless where the losses are originated and assuming that either a packet arrives perfectly or is lost.

IP is usually employed with the transport level protocols transmission control protocol (TCP) and user datagram protocol (UDP). TCP provides reliable and ordered packet delivery and is employed by many popular internet applications, such as World Wide Web (WWW), E-Mail, etc. UDP is commonly used in time-sensitive applications, since it allows lower delays, thanks to a simpler transmission model. Nevertheless, UDP provides unreliable

*Correspondence: fic@gti.ssr.upm.es
Grupo de Tratamiento de Imágenes, E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid, Madrid, Spain

Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 2 of 14

communication: neither the reception of packets nor the order of the received packets are guaranteed. Hence, it is necessary to include additional mechanisms to give reliability to the communication. In this sense, a typical solution is partly represented by the real-time transport protocol (RTP), an application layer protocol developed by the Video Audio Transport Working Group of the Internet Engineering Task Force (IETF) (RFC 3550 [6]) that is generally used over UDP. Although the use of RTP over TCP is not excluded, in this work we focus on the widely used UDP/RTP combination.

RTP defines a packet format for delivering data with real-time characteristics, such as interactive audio and video. Moreover, RTP includes payload type identification, sequence numbering, and timestamping that are intended to ease the management of the transmission. However, RTP does not include protection mechanisms. Therefore, in order to avoid losses at a packet level, it is necessary to add complementary recovery modules.

Protection techniques are usually divided in two large families: automatic repeat request (ARQ) schemes and FEC schemes. Whereas ARQ techniques are based on the retransmission of lost information, which is requested by the receiver, FEC techniques consist in generating redundancy data that is sent jointly with media data and can be used to recover lost information at the receiver [7]. Combination of both FEC codes and ARQ techniques has also been proposed as in the hybrid automatic repeat request (HARQ) [8].

In time-sensitive communication, FEC-based schemes are usually preferred to ARQ schemes, since no extra delay is added due to the retransmission of lost information [7]. Nevertheless, FEC techniques also add some latency derived from the encoding and decoding operations, that has to be considered in this scenario.

In addition, regarding scenarios where multicast communications are used, (e.g., IPTV and multi-client video conference) with a considerable number of clients, the adoption of either FEC or ARQ schemes is a key decision that may influence the success of the protection scheme used. Since each receiver might suffer from different losses, in ARQ schemes, each receiver has to send different requests to the transmitter, generating an additional traffic that may further contribute to the congestion of the network. This situation is different for FEC-based schemes, where the redundancy produced at the transmitter is always sent to all the receivers, so each one can use it if necessary. Therefore, although FEC-based schemes introduce additional band occupation, due to the transmission of redundant recovery data, they seem more suitable than ARQ techniques when it comes to time-sensitive multi-client communications.

In this sense, the FECFRAME Working Group has elaborated the RFC 6363 [9] in order to provide a FEC-based protection scheme to RTP multimedia transmissions. RFC 6363 recommends creating a parallel RTP-FEC packetized stream, composed of recovery packets, from a packetized media stream. This parallel stream is originated applying FEC codes across the RTP media packets (RFC 3550). Each RTP-FEC packet contains an RTP header of its own, data that protect the RTP source headers, and the FEC payload.

Different possibilities of FEC codes employed in this area are exclusive OR (XOR)-based codes, basic and interleaved, low-density parity-check (LDPC) codes, Reed-Solomon codes, fountain codes, etc. [4, 5].

Reed-Solomon codes [10, 11] offer the best recovery capability for block codes, since they belong to the class of maximum distance separable (MDS) codes. Nevertheless, they require a considerable computational cost that can limit their use in real-time communications at the application layer [12].

XOR-based codes are largely used in AL-FEC schemes, in particular the interleaved ones proposed by the Pro-MPEG forum in its COP 3 [13], where media packets are organized in matrices, and recovery packets are generated applying XOR operations by columns (in some cases by rows as well). The recovery capability depends on the interleaving matrix dimensions. This type of scheme is particularly useful in the case of bursty losses, since recovery packets have not been generated from consecutive data packets. The main limitation of these FEC schemes is the rigid structure of the interleaving matrix.

This paper only considers those FEC codes which are open source and deliberately leaves out patented codes (e.g., fountain codes). More specifically, we focus on LDPC codes [14], since they can recover multiple (bursty or single) losses, as the data symbols used to generate each redundant symbol are not constrained in a fixed structure, as is the case of interleaved codes. On the other hand, comparing LDPC codes with Reed-Solomon ones, although LDPC codes are sub-optimal in terms of recovery capabilities [12], they require a lower computational cost. Indeed, they need linear or quasi-linear encoding/decoding complexity [15]. This characteristic is very useful in multicast transmission scenarios with different types of receivers, including those with limited computational capability.

LDPC codes are suited for channels where losses occur following a uniform distribution of probability, and each loss is independent from all the other ones, i.e., memoryless channels [16]. Moreover, LDPC codes are defined as large block codes [17], since their robustness increases with the number of information packets involved. Nevertheless, using a large number of information packets implies an increase in the latency, which may be inconvenient for real-time applications.

Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 3 of 14

In our work, we are interested in applying this type of codes to channels with memory, that is, where losses occur in bursts, since IP networks are characterized by this type of behavior. In addition, we consider codes that involve a small number of information packets in an attempt to reduce the required latency. Hence, our first contribution in this paper is an analysis to a priori evaluate the recovery capability of an LPDC code under these conditions. The structure of an LDPC code is defined by a randomly generated parity matrix that identifies which packets are involved in the generation of each FEC packet. As this is a stochastic process, these matrices might not be uniformly robust, hence our approach is capable of assessing those parts of the parity matrix that are weaker against bursty losses and those parts that are stronger. Once the different parts of the parity matrix are outlined, we propose a second algorithm to modify the weakest parts with the aim of improving overall recovery capabilities for channels with memory. Moreover, although our approach is based on giving a new structure to a randomly generated LDPC code, we preserve the intrinsic characteristics of the original LDPC code in order to keep the same decoding algorithm at the receiver.

In the literature, we can find several approaches that use this family of codes in scenarios with channels with memory. In this sense, the authors in [18] propose a technique that allows to construct an LDPC matrix intended for burst erasure correction, using a superposition scheme that consists in replacing the entries of a base matrix H_{base} with binary matrices called superposition matrices [19]. Each 0 entry is replaced by a 0's superposition matrix, while each 1 entry is replaced by a circulant or permutation matrix. The recovery capability depends on the characteristics of the selected permutation matrices. As a simpler alternative, we propose to create an H matrix using classical pseudo-random algorithms, easy to find in the literature, and afterwards modify H in order to make it more robust against bursts of lost packets. This is done by means of a novel characterization of its recovery capability in bursty channels. Another approach for creating the parity matrix is presented in [20]. They propose two algebraic methods for the systematic construction of quasi-cyclic LDPC codes. However, they evaluate their recovery performances for very large values of blocks, while in our approach, we consider a significantly smaller value.

In other works [21, 22], it is also proposed to modify the parity matrix using a column permutation algorithm. The authors propose different metrics to estimate the recovery capabilities of the parity matrix. Again, these works consider a larger number of information packets than that of our approach.

On the other hand, the authors in [17, 23, 24] propose to randomize the order of the transmission of the packets of the stream. This technique allows to decorrelate the

packets affected by bursts and reaches good results for a high number of blocks. On the contrary, for low values of this parameter, our results show that our modified matrices perform better in bursty channels.

Other works address the application of LDPC to bursty channels, focusing on alternative more complex decoding algorithms, like those in [25, 26], or encoding algorithms, like the one in [27], as a mean to recover from error bursts.

The paper is organized as follows: in Section 2, we describe the main characteristics of LDPC codes. In Section 3, LDPC codes for packetized stream are presented. In Section 4, we assess the performance of the proposed LDPC scheme through experimental results. Finally, in Section 5, we present our conclusions.

2 LDPC codes
2.1 An introduction to LDPC codes
LDPC codes were introduced by Gallager [14]. However, they remained unused for more than 30 years, until Mackay and Neal rediscovered them [28]. They belong to the class of linear block codes, which are defined by parameters k, the number of symbols of a data vector $\mathbf{u}$, and n, the number of symbols of a code vector $\mathbf{c}$. Thus, the number of redundancy (or parity) bits is $n - k$, which, in the case of a systematic code, they are added to the k bits of a data vector. Another important parameter is represented by the rate code, $R = k/n$, that is inversely proportional to the added redundancy.

As other linear block codes, an LDPC code is defined by its parity-check matrix H of dimensions $n - k \times n$, whose entries are exclusively 1's and 0's.

The parity-check matrix is so named because it provides $n - k$ parity check equations that generate constraints between data bits and parity bits. Moreover, an LDPC code is defined as a linear block code for which the parity-check matrix H is very sparse, which means a low density (LD) of 1's. In the literature, several algorithms to construct H are presented [15, 23, 28].

As usual, in all linear block codes, in order to generate code vectors $\mathbf{c}$ from data vectors $\mathbf{u}$, we need to define a matrix G, the so called generator matrix, which holds:

$$\mathbf{c} = \mathbf{u}G; \tag{1}$$

The main algorithms that create G from H consist in arranging H in an appropriate form that allows to develop G and construct it in a systematic form. Thus, H is randomly generated and then it is organized as:

$$H = \left[P^T | I_{n-k} \right]; \tag{2}$$

where I_{n-k} is the identity matrix of dimensions $(n - k) \times (n - k)$ and P is a sparse matrix of dimensions $k \times (n - k)$. So, the corresponding G matrix is:

$$G = [I_k | P]; \tag{3}$$

This approach is based on the use of the Gauss-Jordan elimination [29, 30].

On the other hand, matrix H can alternatively be represented as a bipartite graph (also known as Tanner Graph). The graph is formed by code nodes and check nodes and it associates a parity-check equation to each check-node. The number of edges that come out from y_i and c_j defines the degree of the node.

Other important parameters of LDPC codes are the number of 1's in each column, w_c, and the number of 1's in each row, w_r. The first one, w_c, indicates the degree of the code nodes, and the second one, w_r, corresponds to the degree of the check nodes. Taking into account these parameters, an H matrix is defined as low density when $w_c << n - k$ and $w_r << k$.

For a regular LDPC code, the relation between the two parameters holds:

$$n \cdot w_c = (n - k) \cdot w_r; \tag{4}$$

Otherwise, the LDPC code is called irregular [31]. In this paper, we consider only the regular version of these codes, since we are interested in having a fixed number of data packets that generate each FEC packet.

3 Small block LDPC codes in low-latency memory channels

3.1 Protection scheme

In a typical video transmission model based on IP/UDP/RTP protocols, the encoded video stream is encapsulated in RTP media packets that, in turn, are encapsulated in UDP. Since the typical maximum transmission unit (MTU) is 1500 bytes, we assume this as the fixed size of data packet. Since this is a design system parameter, the latency introduced by our algorithm only depends on the parameters k and n.

Our protection system works at the application layer, and it follows the architecture presented in the FEC framework [9]. This means that two RTP instances are generated: one for the source packets and another one for the repair packets (RTP-FEC packets). Thus, an RTP-FEC packet contains an RTP header of its own, and the redundancy data for the RTP source header and its payload. The information protecting header and protecting payload media data is generated by applying the FEC code across the RTP source packets. As an example, in [32], the structure of the FEC packet employed for the Reed-Solomon-based approach is very similar to the structure of our FEC packets.

In our work, each block of k data packets is encoded using an LDPC code. The selected LDPC codes are the so-called low-density generator matrix (LDGM) codes, which are a simplified version of LDPC codes. The main characteristic of this type of LDPC codes is that matrix G and matrix H coincide [17]. Moreover, in [17], the use

of this type of codes is addressed to a very large value of k, whereas in our work, it is significantly lower. The utilization of these codes is very useful to understand how the modifying algorithm works for low values of k, since a modification of the H matrix has a direct impact on the G matrix. We remind that, in the generic LDPC codes, the creation of the G matrix from the H matrix is not trivial and any modifications in H are not reflected in G (see Section 2.1). In the LDGM codes, H is an $(n - k) \times k$ matrix, whereas in the case of classical LDPC codes, it is $(n - k) \times n$ (see Section 2.1). The bipartite graph of LDGM codes is shown in Fig. 1.

A regular LDGM code has a fixed number of 1's per column and a fixed number of 1's per row. The equation that expresses this condition is:

$$k \cdot w_c = (n - k) \cdot w_r; \tag{5}$$

where the edges that link parity nodes to check nodes are not considered in w_c and w_r.

In LDGM codes, a recovery packet is generated for each row of H, as the result of applying the XOR operation to the data packets corresponding to the entries equal to 1 in H (Fig. 2).

The three main steps of our protection scheme are illustrated in Fig. 3: (i) division of data packets into blocks of low k, (ii) encoding of each block of k data packets, and (iii) the outcome of the encoding for each block of k data packets forms a set of $n - k$ packets.

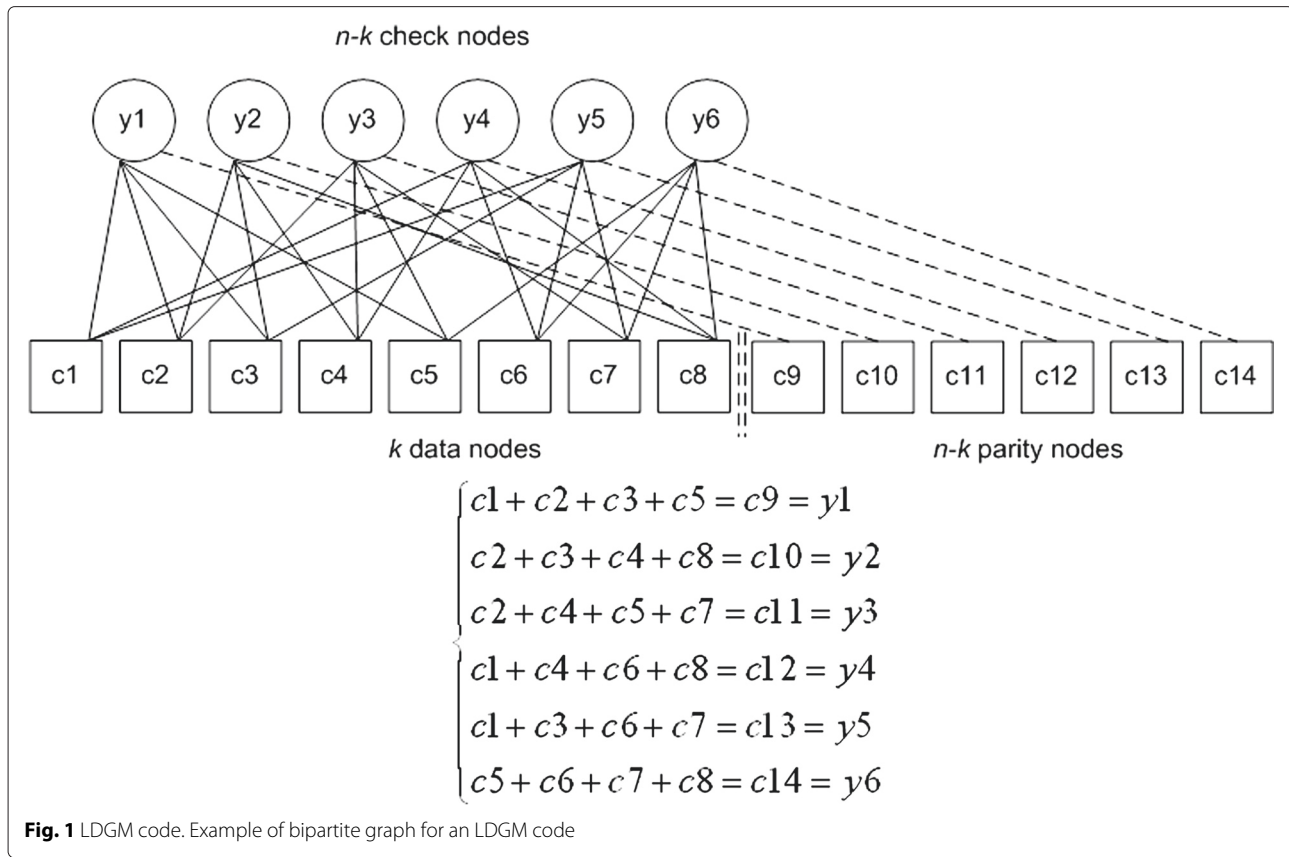### 3.2 Low-density burst-oriented generator matrix

As said before, it is not possible to exactly compute the a priori recovery capability of this family of codes from the value of the parameters k and n. For this reason, we have designed an algorithm that evaluates the global recovery capability of a randomly generated LDGM code for a bursty channel.

This algorithm firstly estimates the contribution of each column of the matrix H to the global recovery capability of the LDGM code (see Section 3.2.1). With that measurement, we are able to identify those parts of the parity matrix that are more sensitive to bursty losses and those parts that are stronger. Moreover, aggregating the values of all the columns of matrix H, we obtain a global measurement of the recovery capability of the matrix.

Finally, taking into account sensitive parts and strong parts, in a second step, our algorithm modifies them with the aim of improving the general recovery capability of the LDGM code keeping its intrinsic parameters, k, n, w_c, and w_r.

3.2.1 *Column-based recovery measurement for matrix H*

We propose the use of a simplified model to estimate this measurement. We assume that only a single burst can occur per block of k packets. Note that, for low values of
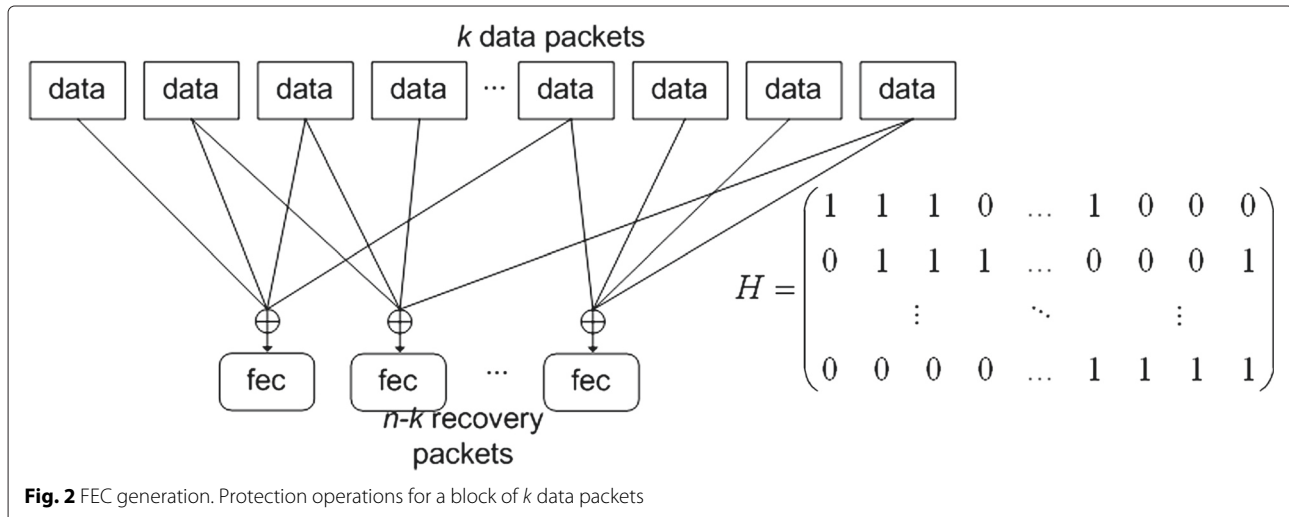
Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 5 of 14

Fig. 1 LDGM code. Example of bipartite graph for an LDGM code

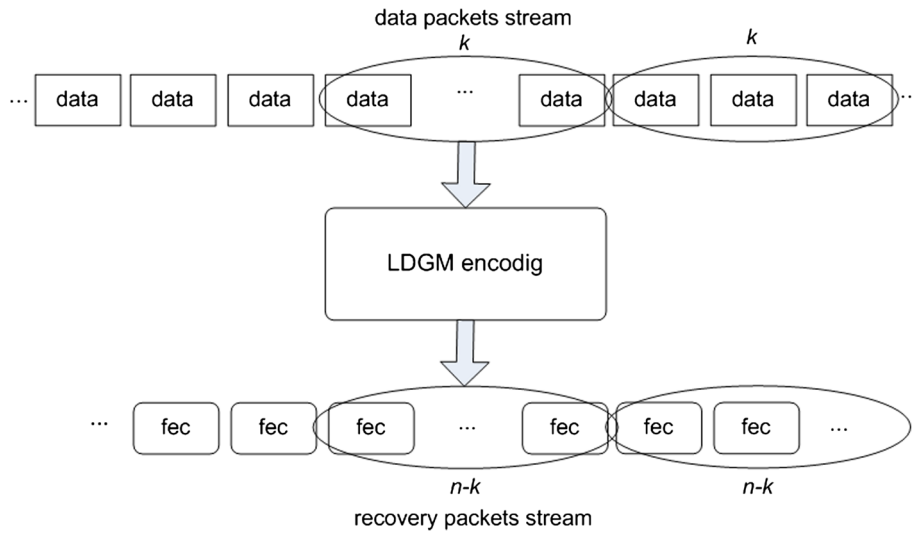packet error rate (PER), this assumption is reasonable for small values of k, as shown in [33].

Let us consider a block of k packets to be protected following the rules indicated by matrix H (Fig. 4). For each column j, we consider all possible bursts that start at the packet in position j and whose lengths range from 2 packets to $n - k$ packets. For each burst, we evaluate whether the lost packets can be recovered or not. We estimate the recovery capability of column j (column-based recovery measurement, CRM) as the number of recovered bursts (the higher this value is, the more recovery capability column j has). Formally:

$$\text{CRM}(j) = \sum_{l=2}^{\max(n-k,n-j)} \text{FEC}\left(B_j^l\right); \qquad (6)$$

where B_j^l is a burst of length l, $l \in (2, n-k)$, starting in packet j, $j \in (0, k-1)$, and FEC($\cdot$) indicates whether

Fig. 2 FEC generation. Protection operations for a block of k data packets

Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 6 of 14

Fig. 3 Protection scheme. Example of the packetized protection scheme based on LDGM codes

the lost packets have been recovered by the LDGM code (FEC($\cdot$) = 1) or not (FEC($\cdot$) = 0).

Hence, our algorithm is able to characterize the recovery capability of H in a column-based manner. We tag as the weakest parts of H matrix those columns that have a low value of CRM, whereas the strongest parts are represented by the columns that achieve a high value of CRM.

Figure 5 shows an example of CRM(j). In this example, it is easy to see the parts where local minima and local maxima of CRM are located.

3.2.2 Global recovery measurement
We give a global measurement of recovery capability: we compute the total amount of bursts that can be recovered
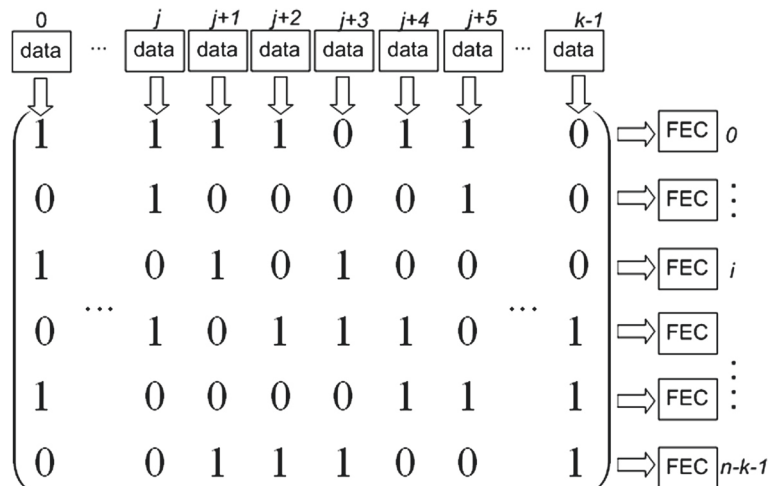
for a given matrix H by the global recovery measurement (GRM), defined as:
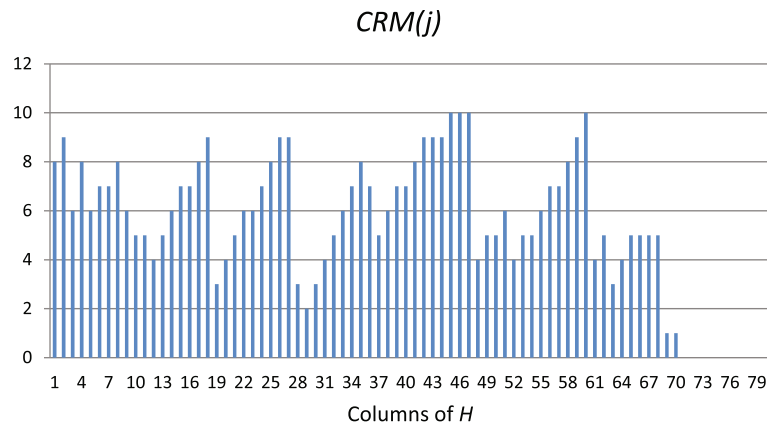
$$\mathrm{GRM}(H) = \sum_{j=0}^{k-1} \mathrm{CRM}(j) \tag{7}$$

The higher GRM(H) is, the higher recovery capability a given matrix H has.

3.2.3 Burst-oriented modifying algorithm
Once the weakest and the strongest parts of the matrix H are identified, the following step of our algorithm consists in identifying the global minimum of CRM(j) of H

Fig. 4 Correspondence column packets

Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 7 of 14

Fig. 5 CRM(j). Number of recovered bursts of lost packets for each position inside a block of k packets, which correspond to the columns of H, that is CRM(j)

(CRM$_{\min}$(j)), that is a minimum of recovery capability in the matrix.

The objective is to increase the value of CRM$_{\min}$(j) with the aim of improving the global recovery capability of the matrix, that is, increasing the value of GRM(H). This operation is performed by a local analysis around column j. For that purpose, a burst of lost packets that begins in j and has an arbitrary length l is considered in order to test how it affects the recovery capability of matrix H. This means that the algorithm considers a sub-matrix within H, that starts in column j and terminates in column ($j + l - 1$), and checks which rows have more than one packet affected by the burst. Those rows are useless to recover lost packets, since each row is able to recover just one lost packet. Therefore, the algorithm aims at modifying the equations defined by H in order to be able to recover the packets affected by the artificial burst.

Let us consider the example in Fig. 6, where the case of a four lost-packet burst (packet j to packet $j + 3$ are lost) is shown. The LDGM code employed in the example cannot recover this burst. However, although Eq. 4 is able to recover packet $j + 3$, which can be employed in Eqs. 0 and 3, Eqs. 0, 2, 3, and 5 have more than one lost packet, so they are useless to recover packets j, $j + 1$, and $j + 3$. If we involve the right packet in those equations whose packets are not involved in the burst, we would be able to recover more lost packets using the remaining equations.

In detail, the algorithm moves the second 1 entry of Eq. 2 to Eq. 1 (generically, from row A to row B), in order to employ the packet $j + 2$ in the generation of the second FEC packet, as shown in Fig. 6. In this new case, the updated LDGM code is able to solve Eqs. 1, 2, and 4 during the first iteration of the decoding algorithm. This means that packets $j + 2$, j, and $j + 3$ can be recovered, since there is only a lost packet per row. In the second iteration, it

is possible to solve Eq. 0 (packets j and $j + 3$ are already known) recovering packet $j + 1$. The algorithm operates this type of changes in the sub-matrix of H that starts in column j, where CRM$_{\min}$ is situated, until column $j + l$, with l defined by the characteristics of the bursty channel.

Finally, in order to keep constant the value of w_r for every row of H, we have to move a 1 entry from row B to row A, in a different part of the matrix, as shown in Fig. 7, which refers to the example presented in Fig. 6. Since this change should not penalize the recovery capability of this part of the matrix, the best option to move the 1 entry from is the sub-matrix of H that starts at column h, where CRM$_{\max}$ is, to $h + l$. Typically, this part is more resilient against modifications than other parts, considering that a larger value of CRM entails a higher robustness against bursts. That means that packets are better organized within the equation.

This operation (modifying the part of the matrix where CRM$_{\min}$ is placed, in order to improve its value) changes GRM(H). The algorithm iterates while GRM(H) keeps on increasing: when this global parameter decreases, the algorithm stops modifying the H matrix. The low-density burst-oriented generator matrix (LDBOGM) is the H matrix that achieves the best GRM(H) value that our algorithm is able to reach. Formally, let GRM$_t$(H) be GRM(H) before the iteration t, and GRM$_{t+1}$(H) its updated value. Then, we continue iterating the algorithm till:

$$\text{GRM}_{t+1}(H) < \text{GRM}_t(H) \qquad (8)$$

The main steps of our algorithm are outlined in Table 1, and Fig. 8 shows all the process of our burst-oriented modifying algorithm by a block diagram.

Finally, it is meaningful to underline that the decoding operations are based on a simple iterative algorithm: given a set of linear equations, if one of them has only
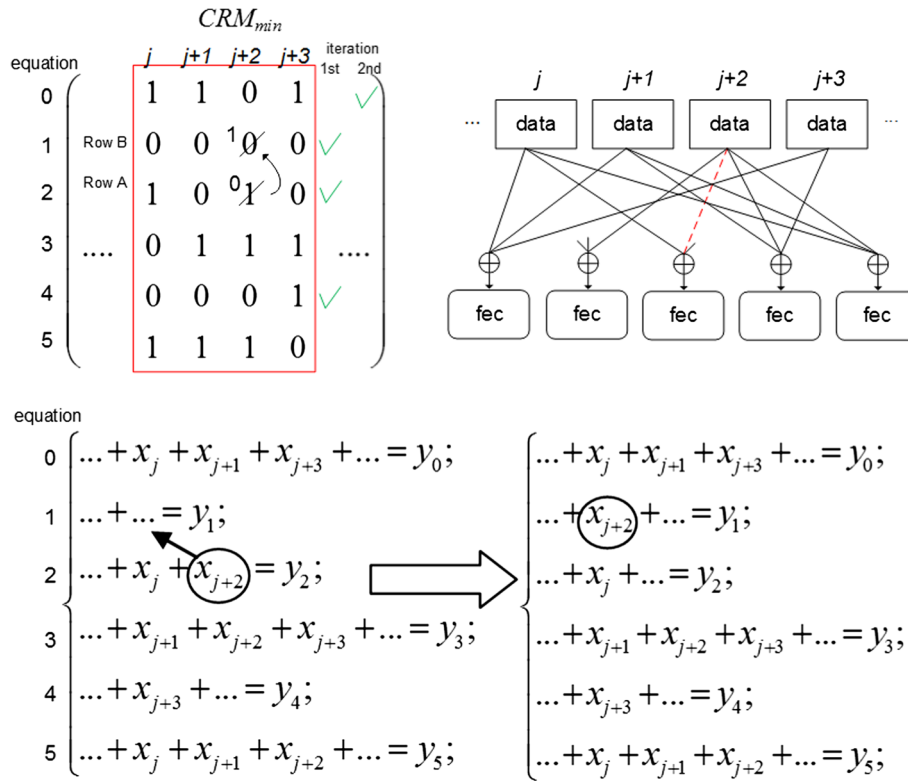
Fig. 6 *H* modification. Example of how to modify an *H* matrix in order to achieve a best recovery capability

one unknown variable, then its value is that of the constant term. This variable is replaced in all remaining linear equations, and the algorithm reiterates. Hence, several unknown variables can be found by the recursive algorithm. Our modifying algorithm does not modify the original iterative decoding algorithm, since the modified generator matrix is known both at encoder and decoder.

4 Simulations and results
4.1 Design parameters

A crucial aspect in the configuration of any channel code is the appropriate selection of parameters k and n according to the characteristics of the communications channel used. In our approach, the choice of these parameters is determined, on the one hand, by statistical characteristics
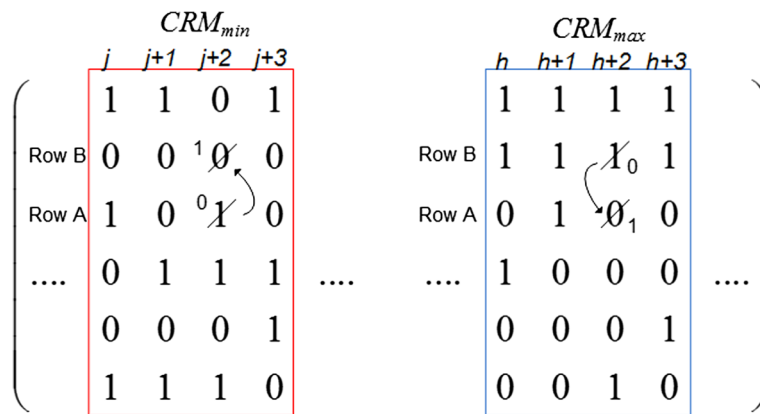
Fig. 7 Fixed number of 1 entries per row. Example of how to keep the fixed number of 1 entries per row (w_r) without modifying the recovery capability of *H*

Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 9 of 14

Table 1 Main steps of burst-oriented modifying algorithm

1. Find the global minimum of CRM(j), CRM$_{min}$(j);

2. Define a window within H that has as first column the corresponding column of CRM$_{min}$(j), the same number of rows of H, and a predefined number of columns l (a window of columns);

3. Solve the equations system H without the packets (variables) used in the sub-matrix of the global minimum;

4. Point out which equations (rows) cannot be solved;

5. Move the 1 entry belonging to an equation unsolved (row A) in the sub-matrix of the global minimum to another equation (row B). Row B is a row that does not loose its recovery capability with this change. The objective is to solve the equations system (as in the example in Fig. 6 where row A is the third one, and row B is the second one);

6. Find the global maximum of CRM(j), CRM$_{max}$(i);

7. Define a new sub-matrix that starts in the column of the global maximum and has the same dimensions of the previous matrix;

8. Move 1 entry from row B to row A, in order to preserve w_r for all the columns of H.

of the channel, and, on the other hand, by the imposed minimum code rate. The aim of this choice is to guarantee the best achievable recovery capability with the lower added redundancy.

The selection of k is particularly relevant because its value directly affects the latency generated at the receiver, becoming a problem in the case of time-sensitive services, such as videoconference. Thus, it is very useful to employ low values for this parameter. In that sense, the approach in [33] employs an interleaving protection scheme with $k = 80$ and is able to recover bursts of up to 20 packets. These values depend also on the rate of the provided services. A typical value proposed for bitrates of about 10 Mbps and for XOR-based interleaved systems is about $k = 100$ [13].

Regarding n, this parameter determinates the amount of added redundancy and the recovery capability. Typical reasonable values are those that achieve a redundancy rate about 15 and 20 % of the total information [33]. Moreover, it is important to underline that in this work, we focus on very low values of k (and n). Whereas in literature, very high values of k are mostly proposed, in our scheme $n = 100$ and $k = 80$.

4.2 Transmission channel

The main statistical characteristics of the considered channels are represented by two values: the PER and the average burst length (L_m).

For this work, we are considering a transmission channel based on a simplified Gilbert-Elliot model according to [34]. The two-state model that defines the simplified Gilbert-Elliot model is shown in Fig. 9, where B indicates the bad state, i.e., the drop of a packet, and G represents the good state, a successful
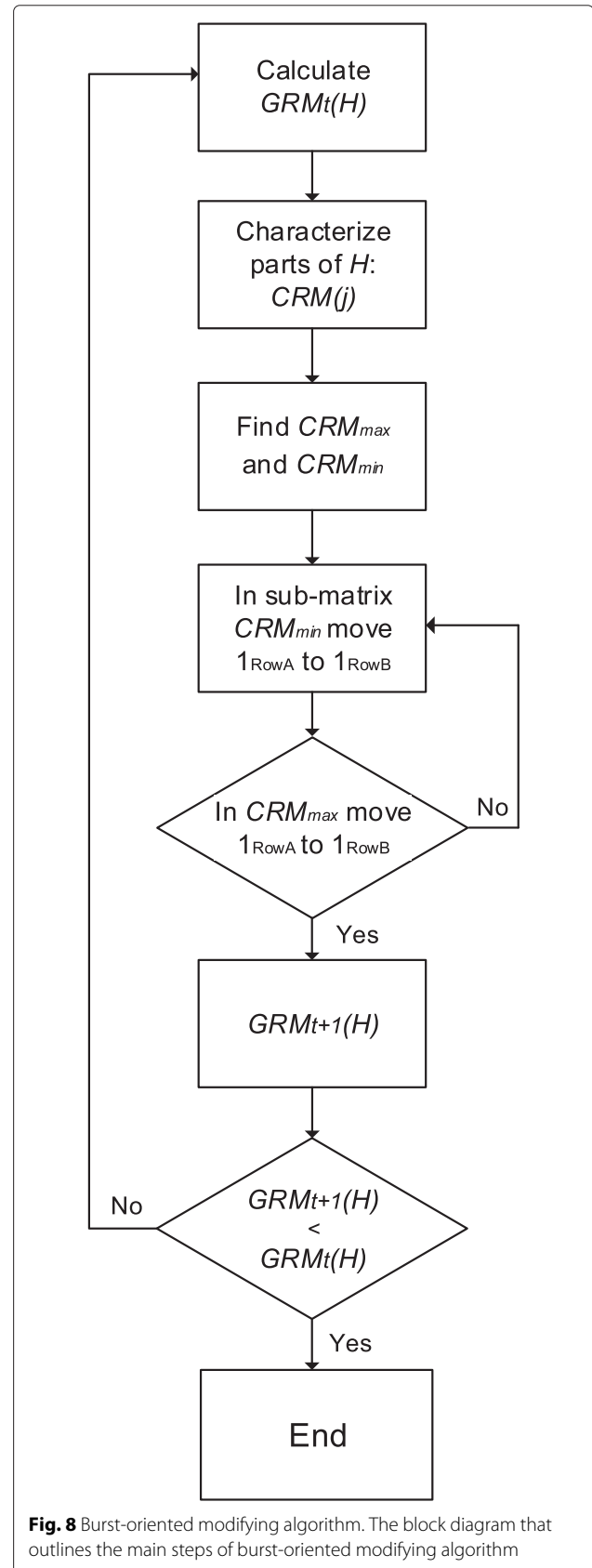
Fig. 8 Burst-oriented modifying algorithm. The block diagram that outlines the main steps of burst-oriented modifying algorithm

Fig. 9 Simplified Gilbert-Elliot model
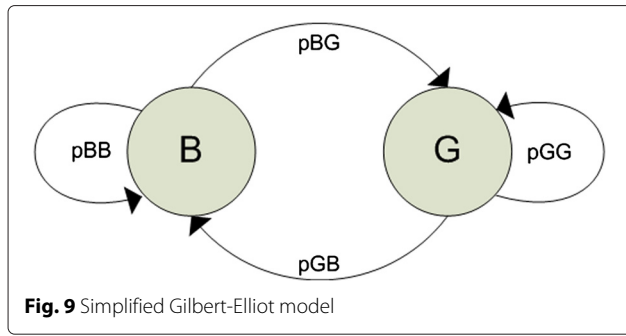
received packet. The transition probabilities are defined by:

$$
\begin{aligned}
p_{\mathrm{BG}} &= \frac{1}{L_m}; \\
p_{\mathrm{BB}} &= 1 - p_{\mathrm{GB}}; \\
p_{\mathrm{GB}} &= \frac{p_{\mathrm{BG}} \cdot \mathrm{PER}}{1 - \mathrm{PER}}; \\
p_{\mathrm{GG}} &= 1 - p_{\mathrm{GB}};
\end{aligned}
\tag{9}
$$

4.3 Simulations

4.3.1 Analysis of the recovery capability of LDBOGM codes

In order to provide a comprehensive evaluation of LDBOGM matrices, we have firstly generated 50 different matrices using the classical stochastic algorithm in [35]
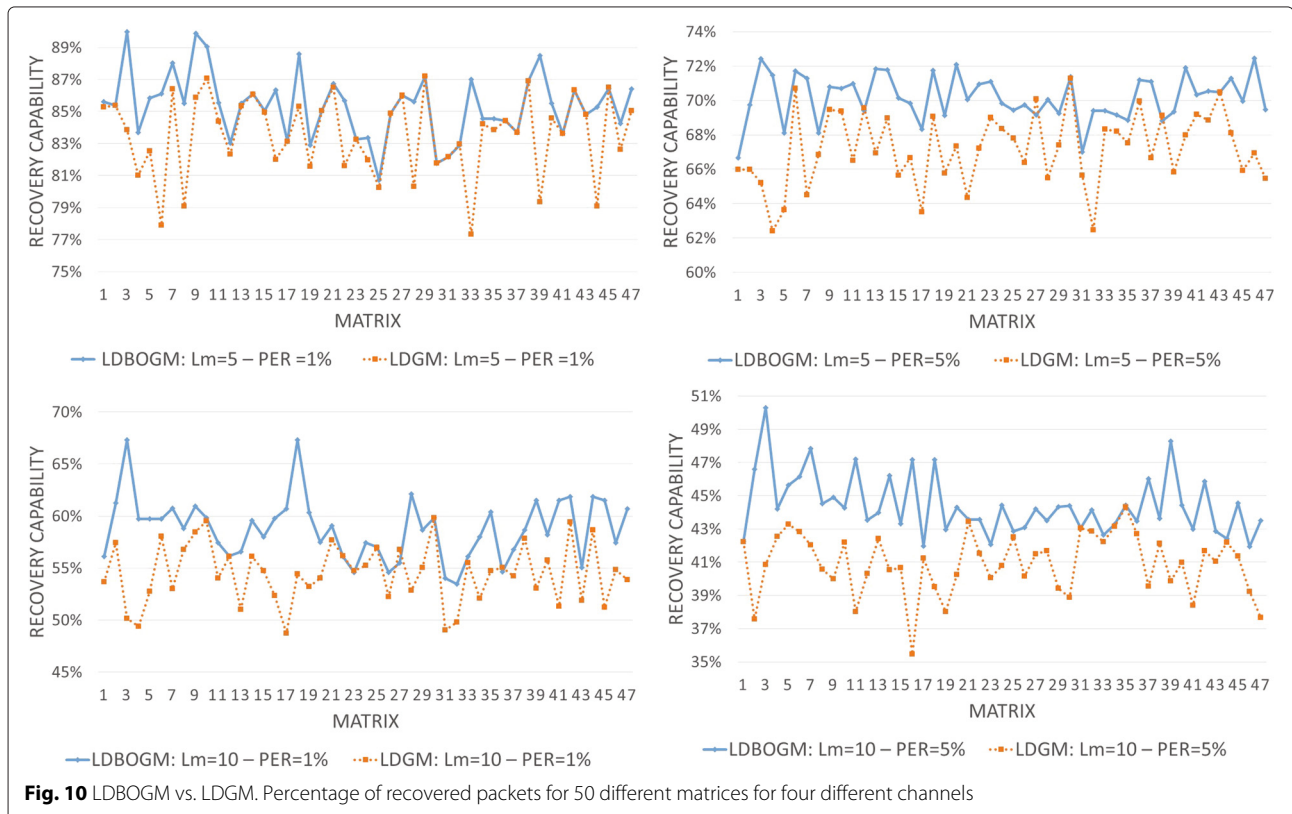
and in [36], keeping n, k, and w_c fixed to values determined in Section 4.1. Every LDGM matrix has been modified using our algorithm, and a corresponding LDBOGM matrix has been generated. Afterwards, for each pair of matrices (original and modified), we have evaluated the recovery capability in a set of experiments. We have done it through the ratio of recovered packets with respect to lost packets.

The experiments have consisted in simulating the transmission of an RTP media stream together with its corresponding FEC stream. The recovery capability is calculated after a total amount of 2000 blocks of k transmitted packets, in order to achieve statistically relevant results.

We have carried out experiments for four different channels as representative of the conditions of wired and wireless networks not only in a normal context but also in congested channels [33, 37–39]. In particular, we have considered channels with two different average burst lengths, $L_m = 5$ and $L_m = 10$, and two values of PER, 1 and 5 %.

Figure 10 shows the obtained results for the LDGM and LDBOGM schemes.

As can be observed, LDBOGMs obtain better results in almost all the experiments. In the best case, our algorithm is able to improve the recovery capability of an LDGM code by up to 10 %. In the worst cases, the recovery

Fig. 10 LDBOGM vs. LDGM. Percentage of recovered packets for 50 different matrices for four different channels

Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 11 of 14

capability remains the same. In Tables 2 and 3, the best, the worst, and the average recovery capability for both schemes are indicated.

Moreover, we consider a scenario where the modification of the matrix is done off-line. Nevertheless, it is interesting to outline that, given $k = 80$, $n = 100$, and $w_c = 3$ (values used in the simulations), the refinement time, calculated as the average time for 50 matrices, is 69.1476 ms. Whereas, the original H matrix creation time is 0.0615 ms. The time measurements are implemented in C++, and they have been carried out in a PC with an Intel Core i7-3540 @ 3 GHz.

4.3.2 Comparative results

To assess the effectiveness of our approach, we have selected the modified matrix that has obtained the best average recovery capabilities in the analysis presented in the previous section (LBOGM) and compared it with: (i) the original matrix, (ii) an XOR-based 2D interleaved code, with 10 columns and 8 rows (identified by Interleaving), and, finally, (iii) a Reed-Solomon scheme, based on the inter-packet approach with $m = 4$ and $n = 60$ proposed in [40] (identified by Reed-Solomon).

The experiments have been carried out following the same methodology described in the previous section, but the evaluation has been extended to a much larger set of channel models. In this sense, we have considered a PER ranging from 0.1 to 20 %, and the average burst length has been extended up to $L_m = 20$. The reason is that in our estimations, burst lengths from 2 to 20 packets are considered. The actual used values are the following: $L_m = 5$, 10, 15, and 20 and PER=0.1, 1, 5, 10, 15, and 20 %.

The results are presented in Fig. 11. They show that our algorithm outperforms the classical LDGM code in all the cases. In addition, they are more robust than the interleaving XOR-based scheme. Finally, one can see that for average burst lengths $L_m > 10$, the performance of the LDBOGM, LDGM, and interleaving decrease significantly compared to one of the Reed-Solomon scheme. This behavior is motivated by two factors: (i) the design of those algorithms is not optimal for burst lengths greater than $L_m = 10$ (for LDBOGM matrices the refinement process has been tuned for a window of $L_m = 10$); (ii) the Reed-Solomon scheme selected as reference uses a larger value of $n - k$. Regarding the size of the analysis window

Table 2 Recovery capability of LDBOGM

| | $L_m = 5$ | $L_m = 5$ | $L_m = 10$ | $L_m = 10$ |
	PER=1 %	PER=5 %	PER=1 %	PER=5 %
Max.	90 %	72.5 %	67 %	50 %
Min.	81 %	67 %	52 %	42 %
Average	85 %	70 %	58 %	44 %

Table 3 Recovery capability of original LDGM

| | $L_m = 5$ | $L_m = 5$ | $L_m = 10$ | $L_m = 10$ |
	PER=1 %	PER=5 %	PER=1 %	PER=5 %
Max.	87 %	71 %	63 %	45 %
Min.	77 %	62 %	49 %	35 %
Average	83 %	67 %	55 %	41 %

of the algorithm ($L_m = 10$), it was determined empirically. We conducted several tests with different window sizes. Short windows did not provide good results for the modified matrices since little columns were involved in the analysis, but long windows did not provide good results either. This might be due to the fact that considering a wide neighborhood at the end of the process changes due to different columns tend to void. Nevertheless, LDBOGM again outperforms all the approaches, with the only exception of the Reed-Solomon codes.

4.3.3 Complexity analysis: LDBOGM code vs. Reed-Solomon codes

In this section, we show that, although Reed-Solomon codes perform better, our LDBOGM code has better computing performance.
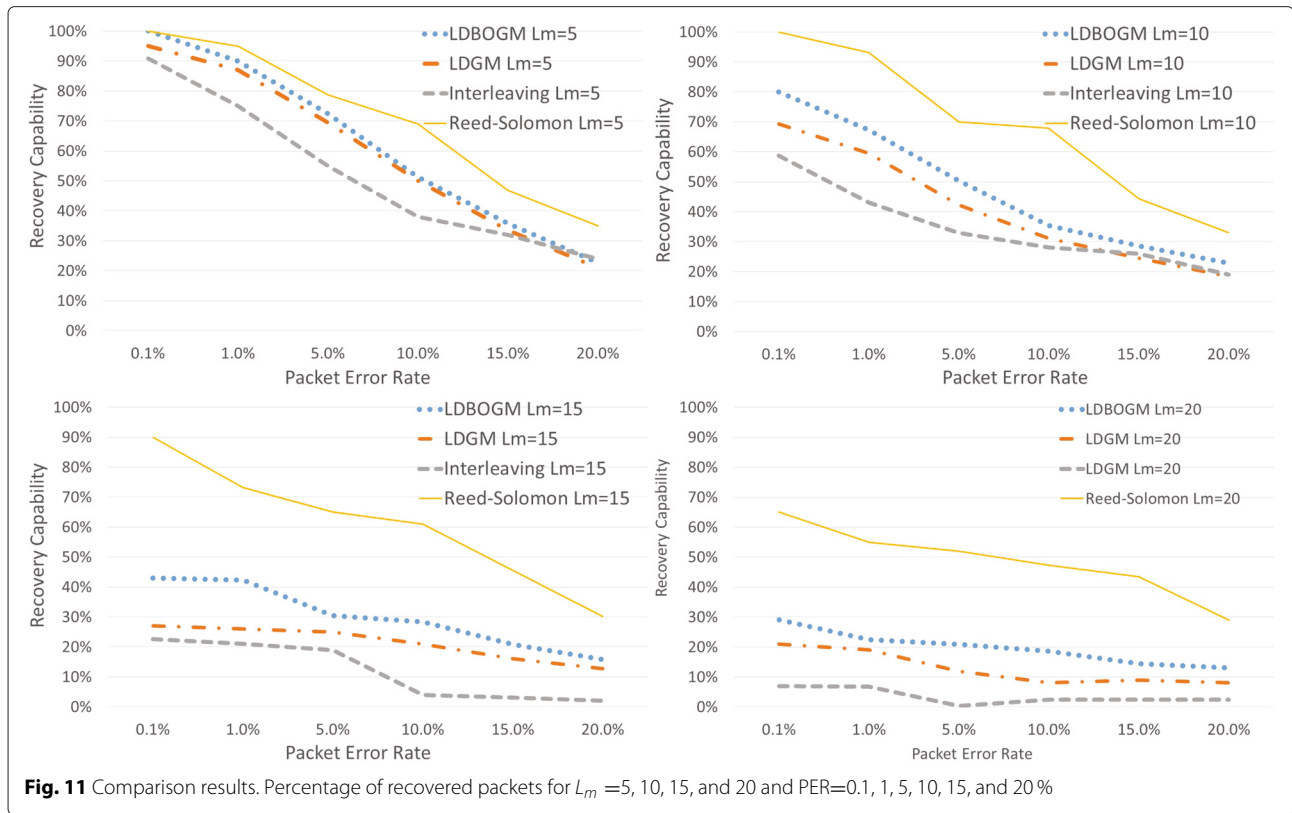
In [41], the authors show that Reed-Solomon codes may have a good performance in terms of complexity requirements, when carefully designed. In this sense, the authors refer to RFC 5510 [42], where the encoding complexity, in the case of the pre-computation of the generator matrix, involves k operations per repair element (vector-matrix multiplication). For the LDGM codes, considered in this paper, the number of operations equals $w_r - 1$, which is far lower than k.

Moreover, regarding the decoding process, in [41], the authors state that, although the decoding speeds are reasonable or high for Reed-Solomon codes, when it comes to small blocks, they are still lower than those of an LDPC-staircase code, which in turn are more complex than the ones used in this work.

In order to estimate the gain in performance that can be achieved using LDBOGM codes with respect to Reed-Solomon codes, we have computed the encoding and the decoding time for both schemes, for given values of n, k, and for a fixed packet size. For the Reed-Solomon implementation, we have used the free codec provided by [43] implemented in C++. Moreover, we have set the Reed-Solomon codec to its shortened version, selecting $n = 100$ and $k = 80$, since these values have been used in our experiments with the LDBOGM codes. The packet size has been set to 1024 bytes. The time measurements have been carried out in a PC with an Intel Core i7-3540 @ 3 GHz. The encoding time for 1024 bytes packets, $n = 100$ and $k = 80$, is shown in Table 4.

Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 12 of 14

Fig. 11 Comparison results. Percentage of recovered packets for $L_m =$ 5, 10, 15, and 20 and PER=0.1, 1, 5, 10, 15, and 20 %

In order to estimate the decoding time of the algorithms, we have simulated the transmission of 300 groups of $n = 100$ packets using four different channel models (average burst lengths $L_m = 5, 10, 15$, and 20, and a PER $= 1\%$). These channel models are a subset of the channel models used in our general experiments. In these simulations, each time a loss has occurred, we have measured the decoding time for each protection scheme. The presented results are the average value per group of $n = 100$ packets, as depicted in Table 5.

As can be observed, in both cases (encoding and decoding) gains in terms of computing time have been obtained using the LDBOGM code: 81 % for the encoding time, from 26 % up to 54 % for the decoding time.

5 Conclusions

In this paper, we have proposed an algorithm, the burst-oriented modifying algorithm, that allows boosting the recovery capability of LDPC codes against bursty packet losses, in contrast to the original randomly generated H matrix, which is oriented to memoryless channels.

Moreover, in this work, we use small values of k (small information blocks), in order to fulfil low transmission latency requirements for time-sensitive communications.

Our approach is based on a novel analysis that evaluates the recovery capability of each part of matrix H, with the aim of finding the weakest and strongest parts against bursts of lost packets. This analysis has been made by defining a local parameter, the CRM, column-based recovery measurement, and a global parameter, the GRM(H), global recovery measurement.

Once the different parts of the matrix H are identified (minima and maxima of CRM), the second step of our algorithm is to modify the weakest parts of the matrix H in order to make these parts stronger against bursts of lost packets, with the aim of improving the GRM(H) value.

Finally, in order to assess the effectiveness of our algorithm, we have carried out two types of experimental tests. The first one consists in the generation of

Table 4 Encoding time for Reed-Solomon codes and LDBOGM codes

LDBOGM	Reed-Solomon	Performance gain
0.009 ms	0.048 ms	81 %

Table 5 Decoding time for Reed-Solomon codes and LDBOGM codes, PER=1 %

	$L_m = 5$	$L_m = 10$	$L_m = 15$	$L_m = 20$
LDBOGM	4.535 ms	6.152 ms	5.891 ms	7.862 ms
Reed-Solomon	8.950 ms	9.910 ms	12.734 ms	10.623 ms
Performance gain	49 %	38 %	54 %	26 %

50 different LDGM matrices, which have been modified using our algorithm to generate the corresponding LDBOGM matrices. Afterwards, we have evaluated the recovery capability for each pair of matrices (original and modified) for four different bursty channels in an RTP transmission. We have demonstrated that LDBOGMs obtain better results in almost all the experiments. In the second test, we have selected the modified matrix that has obtained the best average recovery capabilities in the analysis presented in the previous section (LBOGM) and compared it with (i) the original matrix, (ii) an XOR-based interleaved code, and (iii) a Reed-Solomon scheme. After a simulated RTP transmission in 24 different bursty channels, we state that our algorithm outperforms the classical LDGM codes and the interleaving XOR-based scheme. In addition, the recovery capabilities of the LDBOGM code approaches the values reached by the Reed-Solomon codes in some cases.

References
1. KR Rao, ZS Bojkovic, DA Milovanovic, *Multimedia communication systems: techniques, standards, and networks*. (Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002)
2. Mvd Schaar, PA Chou, *Multimedia over IP and wireless networks: compression, networking, and systems*. (Academic Press, Inc., Orlando, FL, USA, 2007)
3. A Talari, S Kumar, N Rahnavard, S Paluri, J Matyjas, Optimized cross-layer forward error correction coding for H.264 AVC video transmission over wireless channels. EURASIP J. Wireless Commun. Netw. **2013**(1) (2013)
4. K Matsuzono, J Detchart, M Cunche, V Roca, H Asaeda, in *Local Computer Networks (LCN), 2010 IEEE 35th Conference On*. Performance analysis of a high-performance real-time application with several AL-FEC schemes (IEEE, Denver, CO, USA, 2010), pp. 1–7
5. M Luby, T Stockhammer, M Watson, IPTV systems, standards and architectures: part II—application layer FEC in IPTV services. Comm. Mag. **46**(5), 94–101 (2008)
6. H Schulzrinne, S Casner, R Frederick, V Jacobson, RTP: a transport protocol for real-time applications. Technical report, RFC 3550 (2003). http://www.ietf.org/rfc/rfc3550.txt
7. C Lamoriniere, A Nafaa, L Murphy, in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. Dynamic switching between adaptive FEC protocols for reliable multi-source streaming (IEEE, Honolulu, HI, USA, 2009), pp. 1–6
8. M-F Tsai, T-C Huang, C-K Shieh, K-C Chu, Dynamical combination of byte level and sub-packet level FEC in HARQ mechanism to reduce error recovery overhead on video streaming over wireless networks. Comput. Netw. **54**(17), 3049–3067 (2010)
9. M Watson, A Begen, V Roca, Forward error correction (FEC) framework. Techn. report. RFC 6363 (2011). http://www.ietf.org/rfc/rfc6363.txt
10. SB Wicker, *Reed-Solomon codes and their applications*. (IEEE Press, Piscataway, NJ, USA, 1994)
11. L Rizzo, Effective erasure codes for reliable computer communication protocols. ACM SIGCOMM Comput. Commun. Rev. **27**(2), 24–36 (1997)
12. A Nafaa, T Taleb, L Murphy, Forward error correction strategies for media streaming over wireless networks. Commun. Mag. IEEE. **46**(1), 72–79 (2008)
13. P-M Forum, Transmission of professional MPEG-2 transport streams over IP networks. Code of practice #3 release 2, Pro-Meg Forum (July 2004) http://www.pro-mpeg.org/documents/wancop3.pdf
14. RG Gallager, Low-density parity-check codes. Inf. Theory, IRE Trans. **8**(1), 21–28 (1962)
15. TJ Richardson, RL Urbanke, Efficient encoding of low-density parity-check codes. Inf. Theory IEEE Trans. **47**(2), 638–656 (2001)
16. A Bennatan, D Burshtein, On the application of LDPC codes to arbitrary discrete-memoryless channels. Inf. Theory IEEE Trans. **50**(3), 417–438 (2004)
17. V Roca, Z Khallouf, J Laboure, in *Fifth International Workshop on Networked Group Communication (NGC'03)*. Design and evaluation of a low density generator matrix (LDGM) large block FEC codec (Springer Berlin Heidelberg, Honolulu, HI, USA, 2003)
18. SJ Johnson, Burst erasure correcting LDPC codes. Commun. IEEE Trans. **57**(3), 641–652 (2009)
19. J Xu, L Chen, L Zeng, L Lan, S Lin, Construction of low-density parity-check codes by superposition. IEEE Trans. Commun. **53**(2), 243–251 (2005)
20. YY Tai, L Lan, L Zeng, S Lin, KAS Abdel-Ghaffar, Algebraic construction of quasi-cyclic LDPC codes for the AWGN and erasure channels. Commun. IEEE Trans. **54**(10), 1765–1774 (2006)
21. G Hosoya, H Yagi, T Matsushima, S Hirasawa, A modification method for constructing low-density parity-check codes for burst erasures. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E89-A**(10), 2501–2509 (2006)
22. E Paolini, M Chiani, Construction of near-optimum burst erasure correcting low-density parity-check codes. Commun. IEEE Trans. **57**(5), 1320–1328 (2009)
23. V Roca, C Neumann, D Furodet, Low density parity check (LDPC) staircase and triangle forward error correction (FEC) schemes. Tech. report, RFC 5170 (2008)
24. V Roca, C Neumann, Design, evaluation and comparison of four large block FEC codecs, LDPC, LDGM, LDGM Staircase and LDGM Triangle, plus a Reed-Solomon small block FEC codec. Rapport de recherche RR-5225, INRIA (2004). http://hal.inria.fr/inria-00070770
25. M Fossorier, in *Information Theory, 2006 IEEE International Symposium On*. Universal burst error correction (IEEE, Seattle, WA, USA, 2006), pp. 1969–1973
26. M Fossorier, Hybrid burst erasure correction of LDPC codes. Commun. Lett. IEEE. **13**(4), 260–261 (2009)
27. E Martinian, C-EW Sundberg, Burst erasure correction codes with low decoding delay. Inf. Theory IEEE Trans. **50**(10), 2494–2502 (2004)
28. DJC MacKay, Good error-correcting codes based on very sparse matrices. Inf. Theory IEEE Trans. **45**(2), 399–431 (1999)
29. A Shokrollahi, LDPC codes: an introduction, 2 (2003)
30. W Ryan, An introduction to LDPC codes (2003). http://www.telecom.tuc.gr/~alex/papers/ryan.pdf
31. TJ Richardson, MA Shokrollahi, RL Urbanke, Design of capacity-approaching irregular low-density parity-check codes. IEEE Trans. Inf. Theory. **47**(2), 619–637 (2001)
32. M Mazzotti, E Paolini, M Chiani, B Gadat, C Bergeron, R Fracchia, in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. Analysis of packet-level forward error correction for video transmission (IEEE, Yokohama, Japan, 2011), pp. 1–5
33. C Diaz, J Cabrera, F Jaureguizar, N Garcia, A video-aware FEC-based unequal loss protection system for video streaming over RTP. IEEE Trans. Consum. Electron. **57**(2), 523–531 (2011)
34. Network model for evaluating multimedia transmission performance over internet protocol. Technical report, ITU-T Recommendation G.1050 (November 2007). http://www.telecom.tuc.gr/~alex/papers/ryan.pdf
35. V Roca, C Neumann, M Cunche, J Laboure, LDPC-Staircase/LDPC-Triangle/LDGM Codec Reference Implementation. http://planete-bcast.inrialpes.fr
36. OpenFEC Project. http://openfec.org
37. V Miguel, J Cabrera, F Jaureguizar, N Garcia, in *IEEE International Conference on Consumer Electronics (ICCE), 2011*. High-definition video distribution in 802.11g home wireless networks (IEEE, Las Vegas, NV, USA, 2011), pp. 211–212

Casu *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:187

Page 14 of 14

38. C Diaz, J Cabrera, F Jaureguizar, N Garcia, in *Visual Communications and Image Processing (VCIP), 2012 IEEE*. Adaptive protection scheme for MVC-encoded stereoscopic video streaming in IP-based networks (IEEE, San Diego, CA, USA, 2012), pp. 1–6

39. AC Begen, in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*. Error control for IPTV over XDSL networks (IEEE, Las Vegas, NV, USA, 2008), pp. 632–637

40. F Casu, J Cabrera, F Jaureguizar, N Garcia, in *IEEE Symposium on Computers and Communications (ISCC), 2011*. Inter-packet symbol approach to Reed-Solomon FEC codes for RTP-multimedia stream protection (IEEE, Kerkyra, Greece, 2011), pp. 49–54

41. V Roca, M Cunche, C Thienot, J Detchart, J Lacan, in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference On*. RS LDPC-Staircase codes for the erasure channel: standards, usage and performance (IEEE, Lyon, France, 2013), pp. 638–644

42. J Lacan, V Roca, J Peltotalo, S Peltotalo, Reed-Solomon forward error correction (FEC) schemes. Technical Report 5510 (April 2009). Internet Engineering Task Force. http://www.ietf.org/rfc/rfc5510.txt

43. Schifra Reed-Solomon error correcting code library. www.schifra.com