Residual Packet Loss Rate Analysis of 2-D Parity Forward Error Correction★

Firouzeh Golaghazadeh, Stéphane Coulombe*, Jean-Marc Robert

Software and IT Engineering Department, École de technologie supérieure, Montréal, Canada

Abstract

To improve the robustness of video communications over IP networks against packet losses, numerous IPTV and broadcasting standards have adopted the 2D application-layer forward error correction (FEC) method presented in Pro-MPEG COP3. This method uses a two-dimensional encoding matrix in which redundant checksums for each row and each column of the data matrix are generated and transmitted with the data packets. The approach has been recently extended and applied to layered video streaming. But despite its wide adoption and importance, a formal analysis of the correction performance of the Pro-MPEG COP3's 2D FEC method is still lacking. In this paper, we formulate and solve the problem using an abstract mathematical framework, which can be used also for some natural extensions of this method. We derive tight bounds on the residual packet loss rate as a function of the 2D FEC matrix dimensions, and the packet loss rate assuming that the losses are independently and randomly distributed. We then provide simple approximations on these bounds for low packet loss rates. The analysis leads to the observation that the addition of a single checksum packet to the Pro-MPEG COP3 method would lead to a significant reduction in residual packet loss rate. The expressions derived are validated experimentally using Monte Carlo simulations. We also apply the results to the case of packet losses occurring in bursts. Finally, we study the impact of the matrix dimensions on the performance of the 2D FEC method, leading to valuable recommendations to system designers.

Keywords: Forward error correction (FEC), 2D FEC, dual FEC, 2-D Parity FEC, Pro-MPEG COP3, SMPTE 2022-1, SMPTE 2022-5, video transmission, residual packet loss rate

1. Introduction

During the last decade or so, video services have started migrating from circuit-switched (e.g. point-to-point coax) to packet-switched networks in both wireline and wireless infrastructures [1, 2]. This ongoing transition to IP systems enables broadcasters and service providers to reduce their costs, and use their networks more efficiently [3–5]. It also permits to achieve higher operational flexibility and density using either software-based or cloud infrastructures [6–9]. To support all these efforts, the Society of Motion Picture and Television Engineers (SMPTE) has developed the SMPTE ST 2110 Suite of Standards [10], and a family of related standards known as SMPTE ST 2022 [11–17] specifying how to send digital audiovisual content over IP networks. While the former standards address issues such as synchronization, traffic shaping and network delivery timing, the latter ones define the encapsulation of both compressed and uncompressed video data into RTP/UDP/IP streams. SMPTE ST 2022 has undoubtedly been the most prominent family of standards enabling the transition of video broadcasting systems to IP networks. To deal with the transmission errors and losses inherent to packet-switched networks, these standards propose forward error correction (FEC) methods in ST 2022-1 [11] and ST 2022-5 [15]. The former standard defines the 2D (row/column) FEC method, which will be referred to as 2D FEC in this paper, based on the Pro-MPEG code of practice 3 release 2 (COP3) [18]. This method is illustrated in Fig. 1. It arranges IP video packets into logical rows and columns, and appends one redundant FEC packet to each row and each column. As long as a single packet is lost or damaged in a given row (or column), it can be perfectly recovered from the remaining packets of that row (or column). The latter standard 2022-5 expands on ST 2022-1 to allow larger row/column FEC configurations to support higher bitrate media transport.

Although several application-layer FEC (AL-FEC) methods have been proposed, the one presented in Pro-MPEG COP3 has been widely adopted and deployed due to its very low computational complexity with respect to both generating the redundant FEC packets as well as recovering the lost data packets [19]. Unlike other state-of-the-art error-correcting codes, such as Raptor or RaptorQ [20], a FEC packet is simply the *exclusive or* (XOR) of all the data packets of the associated row or column. Recovering a lost packet only requires the XOR computation of the packets of the related row (or col-

★This work was partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant and by École de technologie supérieure (ÉTS).

*Corresponding author.

Email addresses: Firouzeh.Golaghazadeh.1@ens.etsmtl.ca (Firouzeh Golaghazadeh), Stephane.Coulombe@etsmtl.ca (Stéphane Coulombe), Jean-Marc.Robert@etsmtl.ca (Jean-Marc Robert)

umn). This 2D AL-FEC method has been adopted in a number of IPTV and video broadcasting standards, such as the DVB standard for IPTV (ETSI TS 102 034 [21, 22]), and other proposals of various organizations (e.g., ATIS, ETSI, ITU-T, Open IPTV Forum/HbbTV Association [23]). More recently, it has been adopted in the next-generation media transport standard MPEG Media Transport (MMT) developed as a part of ISO/IEC 23008 [24, 25], and in IETF to protect RTP packets [26]. It has also been recommended as part of the highly popular WebRTC [27, 28] real-time communication framework for browsers and mobile applications.

The Pro-MPEG COP3's AL-FEC method has been recently extended and applied to layered video streaming. An excellent survey reviews the various layer-aware FEC (LA-FEC) methods in this new context [29]. In this set-up, one of the main problems is the fact that the enhancement layers (ELs) become useless when the base layers (BLs) are lost. To address this problem, an enhanced Pro-MPEG COP3 method to protect two-layered video streams has been proposed [19]. Instead of having independent 2D FEC-protected data matrices for both layers, the enhancement layer (EL) matrices are composed of shuffled packets of the base layer (BL) to increase the robustness of this latter layer. The Pro-MPEG COP3 method has also been extended to offer unequal protection by grouping packets according to their importance, or organizing them into matrices of different sizes [30]. This led to complex optimization problems which were approximated by using simulated annealing [31] and tabu search [32]. Finally, XOR-based and diagonal XOR-based AL-FEC methods were recently suggested to improve tolerance to burst errors compared to Pro-MPEG COP3 and its proposed extensions [33, 34].

Considering the wide adoption and the importance of Pro-MPEG COP3's AL-FEC method, evaluating its performance is crucial. It is straightforward to evaluate its overhead, code rate, and latency. However, determining analytically its residual packet loss rate (RPLR) after the recovery of the lost packets, as a function of the packet loss rate (PLR) and matrix dimensions, represents a very challenging problem. In the literature, RPLR has been measured empirically in different application contexts in its original [35, 36], and extended forms [19, 30–34]. To gain insight on the impact of the FEC matrix dimensions on the correction performance, the work of Westerlund [36] is particularly interesting. The author integrated the AL-FEC method into an existing video streaming application, and tested the implementation with several matrix dimensions in both simulated and real-world wireless conditions.

Unfortunately, a formal and extensive analysis of the correction performance of Pro-MPEG COP3's AL-FEC method, as a function of the dimensions of the matrix, is still lacking. To the best of our knowledge, the only attempt to derive such mathematical expressions is the work of Battisti *et al.* [37]. The authors proposed an approximation equation for RPLR but, although the approximation is rather accurate for high PLR values, it is inaccurate for low ones.

In this paper, we present the mathematical tools required to formally and rigorously evaluate the residual packet loss rate of the Pro-MPEG COP3 AL-FEC method. We analyse and compare the performance of various FEC matrix configurations, and derive these results as functions of the matrix dimensions with the sole assumption that the packet losses are independently and randomly distributed. More specifically, the contributions of this work are as follows:

- We formulate and solve the problem using a mathematical framework permitting not only to solve the problem of interest but also similar ones (e.g. if the lowest-rightmost packet of the matrix were present, improving significantly the error correction capability but departing therefore from the standard).
- We derive tight theoretical lower and upper bounds on RPLR as a function of PLR and matrix dimensions. Simple approximations on these bounds are also given for low PLR values.
- We validate these expressions experimentally using Monte Carlo simulations.
- We compare the performance of various matrix dimensions and shapes (square versus rectangular) for broad ranges of PLR.
- We finally show how these bounds on RPLR can be used to analyse bursts of packet losses under the Gilbert model.

This study can be extremely useful to system designers to determine the appropriate matrix dimensions under packet loss, overhead, and latency constraints. Furthermore, since recent standards mentioned earlier offer various FEC strategies, which include Pro-MPEG COP3's AL-FEC method, it is important to provide mathematical tools comparing its performance with newer methods. This will allow the selection of the best FEC strategy among those available based on various criteria beyond error protection alone. For instance, in low-cost low-energy devices such as those found in the multimedia Internet of things (M-IoT) applications [38], Pro-MPEG COP3's AL-FEC method may be preferred to other ones due to its implementation simplicity and low computational complexity.

It is important to note that, although we assume herein that packet losses are independently and randomly distributed rather than occurring in burst, the study is valuable for several reasons. Firstly, some networks really exhibit independently and randomly distributed errors. This is actually for these kinds of losses that 1D row FEC is used. As an example, Lemonnier studied the delivery of video packets with 2D FEC in a fixed Internet scenario with a uniform distribution of packet loss errors [39]. Secondly, although bursts of lost packets may occur (e.g. in wireless networks), their impact can, sometimes, be mitigated by using packet interleaving in conjunction with FEC [3, 21, 40]. This approach randomly shuffles packets before any processing. It introduces additional processing delays which are not desirable but some services may tolerate them. Thus, a balance between the expected network performance and FEC/interleaving strategies is required to provide the best FEC protection while meeting the user requirements in terms of delay [3]. Thirdly, some aspects of our study may be extended to evaluate the performance of recently proposed methods where FEC matrices are comprised of nonconsecutive (shuffled) packets. This is the case in [30] where packets are arranged in m matrices of sizes $D_i \times L_i$, $1 \leq i \leq m$, based on their importance. Finally, the proposed mathematical framework and results can serve as a basis and be extended to evaluate the residual packet loss rate of the Pro-MPEG COP3 AL-FEC approach under burst errors, especially under the classical Gilbert model. This paper is a first step in that direction.

The paper is organized as follows. In Section 2, we present the 2D FEC as proposed in Pro-MPEG COP3. We describe the relevant performance metrics for 1D and 2D FEC in Section 3. In Section 4, we introduce the theoretical tools to solve the difficult problem of evaluating the 2D FEC residual packet loss rate. We derive RPLR simple but accurate approximations for low PLR values for both 1D and 2D FEC in Section 5. In Section 6, we consider the problem of estimating the residual packet loss rate under the classical Gilbert model of error bursts. The bounds derived for the random error model are essential to analyse this latter problem. Finally, we present and analyse the experimental results in Section 7, and conclude this paper and suggest future works in Section 8.

2. Overview of Pro-MPEG COP3 2D FEC

In this section, we present an overview of the 2D FEC method proposed in Pro-MPEG COP3 [18]. We first show how the method organizes the data packets, and adds redundant information to deal with transmission losses. Then, we describe the configurations of packet losses leading to deadlocks, i.e. cases where packets cannot be recovered. We finally present some statistics on the types of deadlock configurations to justify the importance of a rigorous approach to evaluate the performance of the method.

2.1. Organization of packets in Pro-MPEG COP3

The Pro-MPEG COP3 correction codes are erasure codes operating at the application layer [32]. At that layer, networks behave like packet erasure channels where packets are either perfectly received, or lost but their locations are nevertheless known. Events leading to packet losses include the packet discards at intermediate routers (e.g., due to network congestion), the transmission timeouts (e.g., when a packet is delayed too long to be of interest), or the rejections of corrupted packets (e.g. when a received packet contains bit errors that lower layers cannot correct).

Erasure codes add redundant information to the communication to better protect the data. They are used in several situations: (i) when the PLR is high, retransmissions are inefficient since they would occur too frequently, leading to a reduced system throughput (ii) when the communication is unidirectional (e.g. broadcast), no return channel is available to request retransmissions, and (iii) when the round-trip time (RTT) delay is too high to be acceptable in a real-time application [41]. Because these codes are paramount to avoid losing data, the amount of added redundancy can be substantial.

The FEC method proposed in Pro-MPEG COP3 is a two-dimensional encoding using redundant checksums to recover lost packets. In this paper, this technique is referred to as 2D FEC, but it has been presented under different names in the literature: 2D FEC [42], dual FEC [18], and 2-D Parity FEC [43].

As described in the standard [18], the real-time transport protocol (RTP) packets transmitting the video data are grouped into a two-dimensional $D{\times}L$ data matrix to which $D{+}L$ extra checksum packets, called FEC packets, are added: one for each row and one for each column as illustrated in Fig. 1. These checksums are simply the bitwise exclusive OR (XOR) of the packets on the corresponding row or column. To ensure that all packets are of the same length when performing the operation, padding is used when needed.

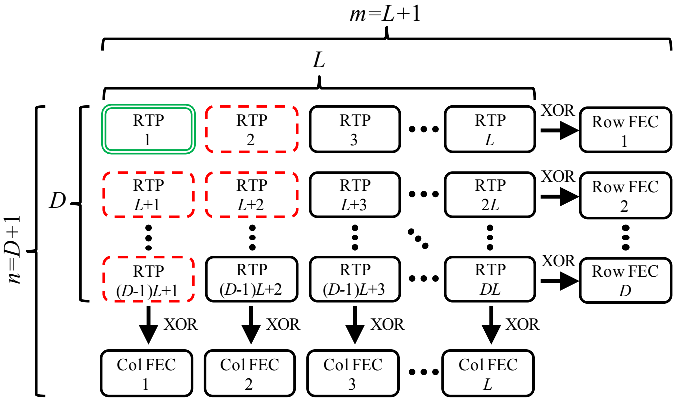


Figure 1: Pro-MPEG COP3 2D FEC matrix showing the logical grouping of RTP data packets into D rows and L columns. Typical values described in this standard encompass data matrices of dimensions $4{\times}25$, $5{\times}20$, and $10{\times}10$ – excluding the extra FEC packets. These matrices are referred to as $D{\times}L$ *data matrices* to describe the configuration of data packets but they contain in fact $(D + 1) \times (L + 1) - 1$ elements due to the presence of FEC packets and the absence of the lowest-rightmost packet. In this paper, they are referred to as $n \times m$ matrices (or full matrices).

The 2D FEC method enhances the robustness of video packets against losses at the cost of increasing the transmission overhead and latency. Although a two-dimensional arrangement can provide a better robustness, a one-dimensional arrangement is also proposed in Pro-MPEG COP3. In such a case, the FEC packets associated with the *columns* are generated and transmitted. This choice provides a better robustness against bursts of lost packets compared to the choice of using the FEC packets associated with the *rows* only. However, this latter choice would reduce the latency, though.

2.2. Recoverable and non-recoverable configurations

To illustrate how lost packets can be recovered using Fig. 1, suppose that RTP packets 2, L+1, $L+2$and $(D{-}1)L{+}1$ (shown in red dashed rectangles) have been lost. Fortunately, these packets can be recovered in this particular case. RTP 2 is recovered from the XOR of the other packets in the first row, including row FEC 1. Similarly, RTP $L+2$ is recovered from the newly recovered RTP 2 and the other packets of column 2, RTP $L+1$ from RTP $L+2$ and the other packets of row 2, and finally RTP

$(D-1)L+1$ from the other packets of row D. Naturally, this is not the only way to recover them.

However, if RTP 1 were also lost in the previous example, only RTP $(D-1)L+1$ could have been recovered, and it would have been impossible to recover the remaining lost packets. Indeed, the 2×2-submatrix defined by the RTP 1, RTP 2, RTP L+1 and RTP $L+2$ packets has two lost packets in each of its rows and columns. This prevents the recovery of any of these packets. Although, such a rectangular packet loss pattern always leads to a situation where these packets cannot be recovered, more complex patterns leading to the same result exist. These patterns are referred to as *deadlock configurations*. For the specific case of rectangular loss patterns, these inter-blocking configurations are referred to as *4-cycle* deadlock configurations. In general, we will observe *2c-cycle* deadlocks for $c \geq 2$. This terminology is derived from the classical graph theory concepts used in this paper [44]. This terminology would be clearer in Section 4 with the examples presented in Fig. 2.

2.3. Percentage of deadlock configurations

To determine the robustness of the 2D FEC method against packet losses, the number of *recoverable* configurations has to be established, for any given number of lost packets k. These configurations are the only ones, which do not lead to a deadlock, among all the possible configurations of k losses. As presented in Table 1, it is obvious that the percentage of deadlock configurations (i.e. percentage of configurations containing at least one deadlock) increases as the number of losses increases. Furthermore, the proportion of complex deadlock configurations with cycles with more than 4 nodes also increases as the number of losses increases. For instance, in a 10×10 data matrix, 4.7% of all the 8-loss configurations have at least one deadlock configuration preventing the full recovery of the transmitted packets. Among these configurations, 7% have a simple deadlock configuration composed of a 6- or 8-cycle. This value increases to 19% for 12-loss configurations. In this case, 6-, 8-, 10-, and 12-cycle are considered.

Table 1: Percentage of deadlock configurations with respect to all possible configurations for a $D \times L = 10 \times 10$ data matrix

Nb of losses	**Deadlocks**	**Among those deadlocks with single 2c-cycles, $c > 2$**
6	1.23%	2.3%
8	4.7%	7%
10	12.7%	13%
12	27.6%	19%

This means that with high packet loss rates, relying *only* on the number of deadlock configurations with at least 4-cycle is not sufficient to accurately estimate the resulting residual packet loss rate. It is therefore important to characterize *all* the deadlock configurations for any number of lost packets in a FEC matrix. As shown in the next sections, determining the residual packet loss rate as a function of packet loss rate is a challenging problem. Its difficulty is exacerbated by the fact that the lowest-rightmost packet of the matrix is intrinsically lost.

3. 1D and 2D FEC Performance Metrics

In this section, the classical performance metrics used to evaluate the 1D and 2D FEC methods are presented. More importantly, analytic expressions for each of them are derived.

To simplify the notations henceforward, the variables $n = D + 1$ and $m = L + 1$ will be used rather than D and L. Hence, the full dimension of the entire FEC matrix of Fig. 1 is $n \times m$.

The first criterion is the *transmission overhead* defined as

$$\text{Overhead}_{\text{FEC}} \triangleq \text{FEC packets/data packets}$$

i.e., the ratio of the number of added checksum (FEC) packets to the number of data (useful) packets. In communication theory, a related concept is the *code rate* defined as

$$\text{CodeRate} \triangleq \text{data packets/sent packets}$$

i.e., the ratio of useful information. Note that the total number of sent packets comprises *both* the data and the FEC packets. These first two metrics are strictly related to each other since $\text{CodeRate} = 1/(1 + \text{Overhead}_{\text{FEC}})$.

As one can expect, a higher overhead would lead to a higher robustness against losses. To measure this second criterion, the *residual packet loss rate* of an error-correcting method is defined as the expectation of the ratio of the number of unrecovered packets to the total number of sent packets. This metric has been chosen to evaluate the performance of the FEC method – as chosen by Battisti *et al.* [37].

Throughout this paper, it is important to distinguish *lost* packets from *unrecovered* ones after the FEC method has been used. The residual packet loss rate is defined as

$$\text{RPLR} = \mathbb{E}\left[\frac{\text{unrecovered}}{\text{sent packets}}\right] \tag{1}$$

where the mathematical expectation $\mathbb{E}[\cdot]$ is evaluated according to a given loss distribution model. Typically, independently and randomly distributed losses, or burst errors are considered. The RPLR criterion can therefore be expressed as

$$\text{RPLR} = \sum_{k} \mathbb{E}\left[\frac{\text{unrecovered}}{m \times n - 1}\middle| k \text{ losses}\right] \Pr[k \text{ losses}]. \tag{2}$$

The last metric is the *latency* defined as the number of buffered packets before being able to process all packets [36]. It is denoted $\text{Latency}_{\text{FEC}}$. The latency is only present at the receiver since the transmitter can send each packet as it becomes available. Obviously, this metric is important for real-time and near real-time communication systems.

3.1. Evaluation of 1D FEC Performance Metrics

In general, the 1D FEC method can be implemented as 1D column FEC, or as 1D row FEC [43]. In the former case, a FEC packet, shown as Col FEC in Fig. 1, is computed for each column. This approach is particularly effective for bursts of lost packets. In the latter case, a FEC packet, shown as Row FEC in Fig. 1, is computed for each row. This approach is effective for random packet losses, and has a lower latency.

As mentioned earlier, Pro-MPEG COP3 only considers the 1D column FEC method. However, since the 1D FEC structure is very similar for both the row and the column FEC methods, except for the latency metric, only the 1D column FEC method will be mentioned henceforward. Hence, the overhead, the code rate and the latency [36] of the 1D FEC method are given by

$$\begin{aligned}\mathrm{Overhead}_{\mathrm{1DFEC}} &= 1/(n-1)\\ \mathrm{CodeRate}_{\mathrm{1DFEC}} &= (n-1)/n\\ \mathrm{Latency}_{\mathrm{1DFEC}} &= \begin{cases} m & \text{for 1D row FEC.}\\ (m-1)n & \text{for 1D column FEC.}\end{cases}\end{aligned}$$

The latency metric is expressed in terms of packets, and relates to the time it takes to process them, assuming that they arrive at a constant rate. For the row 1D FEC method, the receiver must wait until all the row data packets, and the FEC packet have arrived before it starts to process the data. Similarly, for the column 1D FEC method, the receiver must wait until the last column has arrived.

The 1D FEC method can recover *only one* packet. By considering all the cases with at least two lost packets, the residual packet loss rate can be computed from Eq. 2 as follows

$$\begin{aligned}\mathrm{RPLR}_{\mathrm{1D}} &= \sum_{i=2}^{n} \frac{i}{n}\binom{n}{i}\rho^{i}(1-\rho)^{n-i}\\ &= \rho\sum_{i=1}^{n-1}\binom{n-1}{i}\rho^{i}(1-\rho)^{n-1-i}\\ &= \rho(1-(1-\rho)^{n-1}) \qquad (3)\end{aligned}$$

where ρ is the packet loss rate (PLR).

3.2. Evaluation of 2D FEC Performance Metrics

The 2D FEC method is effective for the random as well as the burst packet loss models. The overhead, the code rate and the latency metrics for this method are given by

$$\begin{aligned}\mathrm{Overhead}_{\mathrm{2DFEC}} &= \frac{n+m-2}{(n-1)\times(m-1)}\\ \mathrm{CodeRate}_{\mathrm{2DFEC}} &= \frac{(n-1)\times(m-1)}{n\times m-1}\\ \mathrm{Latency}_{\mathrm{2DFEC}} &= n\times m-1.\end{aligned}$$

For a given matrix overall size, a simple argument based on the derivative and the extrema computations can show that the overhead is minimized when the matrix is as square as possible. Simply compare 11 × 11 matrices with 20% overhead and 6 × 21 matrices with 25%. On the other hand, larger is the square matrix lower is its overhead but higher is its propagation delay – simply compare 11 × 11 matrices with 20% overhead and 21×21 matrices with 10%, whilst their latency are 120 and 440 packets, respectively.

Deriving an accurate mathematical expression for the residual rate $\mathrm{RPLR}_{\mathrm{2D}}$ is quite challenging. To the best of our knowledge, no such expression has been published before. Battisti *et al.* derived some approximations for straightforward cases [37]. However, their approximations only hold for high PLR values. In the next section, accurate lower and upper bounds for $\mathrm{RPLR}_{\mathrm{2D}}$ are given for both low *and* high packet loss rates.

4. Theoretical Bounds of $\mathrm{RPLR}_{\mathrm{2D}}$ for the Random Model

To evaluate the robustness of the 2D FEC method, differentiating configurations with packet losses that can be recovered from the ones that cannot is essential. Straightforward lower and upper bounds on its robustness can be obtained by evaluating the probability of deadlock configurations for a given number of lost packets. In the worst pessimistic case, all the missing packets can be assumed to be lost if a deadlock happens. In the best optimistic case, only three packets can be assumed to be unrecovered – this is the minimal size of a deadlock configuration. The main contribution of the paper is the formal analysis determining the probability that a deadlock occurs for a given number of independent and randomly distributed packet losses.

4.1. The Random Packet Loss Model

Assume that the packet losses are independently and randomly distributed in an $n \times m$ matrix. As mentioned in the introduction, such errors actually occur in some networks, while packet interleaving can sometimes be used to obtain such behaviour in others. More importantly, we will show later in the paper that the results derived for this case can be applied to the more complex case of packet losses occurring in bursts. With this assumption, the probability distribution of the number of lost or erroneous packets $E_{\rho}^{n,m}$ is given by

$$\Pr[E_{\rho}^{n,m}=k] \quad = \quad \tbinom{nm-1}{k}\rho^{k}(1-\rho)^{(nm-1-k)} \qquad (4)$$

where the packet loss rate ρ is the probability that any single packet is lost or damaged during its transmission. Note that since the lowest-rightmost packet is not part of the 2D FEC scheme, there are only $nm-1$ transmitted packets.

The next step is to evaluate the probability that a FEC matrix having lost exactly k packets does not contain any deadlock and can be in fact fully recovered. In order to determine such a probability, the recoverable configurations have to be characterized and enumerated. If their number is denoted $\mathcal{F}_k(n,m)$, this probability can be expressed as

$$\Pr[\text{No deadlock}|E_{\rho}^{n,m}=k] \quad = \quad \frac{\mathcal{F}_k(n,m)}{\tbinom{nm-1}{k}}. \qquad (5)$$

Once this probability has been determined, the residual packet loss rate $\mathrm{RPLR}_{\mathrm{2D}}$ can be derived from Eq. 2, using the chain rule, to obtain

$$\begin{aligned}\mathrm{RPLR}_{\mathrm{2D}} = \sum_{k}\Bigg(\mathbb{E}\left[\frac{\text{unrecovered}}{nm-1}\middle|\text{Deadlocks}\wedge E_{\rho}^{n,m}=k\right]\\ \times\Pr[\text{Deadlocks}|E_{\rho}^{n,m}=k]\times\Pr[E_{\rho}^{n,m}=k]\Bigg). \qquad (6)\end{aligned}$$

4.2. Enumerating the Recoverable Configurations

To ease the evaluation of $\mathcal{F}_k(n,m)$, assume in a first step that the FEC matrix is a full $n \times m$ matrix whose lowest-rightmost packet is also transmitted. Conceptually, this extra packet should correspond to the checksum of the row (or, equivalently, the column) FEC packets. Later on, this assumption will be dropped to correspond to the Pro-MPEG COP3 specification.

The $n \times m$ FEC matrix $\mathcal{M}$ can be used to define a labelled *bipartite graph* $G_{\mathcal{M}} = (R, C, E)$ whose disjoint sets of labelled red nodes $R = \{r_1, r_2, \cdots, r_n\}$ and labelled cyan nodes $C = \{c_1, c_2, \cdots, c_m\}$ represent its rows and its columns, respectively. Assuming that $\mathcal{M}[x, y] = \perp$ if the packet in the x^{th} row and y^{th} column of the matrix has been lost during its transmission, the edges in E represent these packets as follows: $E = \{(r_i, c_j) | (\exists r_i \in R)(\exists c_j \in C)[\mathcal{M}[i, j] = \perp]\}$. Thus, any edge of E connects one node of R (a row) and one node of C (a column). By construction, the *degree* of a node (i.e., the number of incident edges) represents the number of lost packets in the corresponding row or column. Elementary graph theory concepts are used in this section. Readers are referred to any classical reference on graph theory such as the seminal book of Bondy and Murty [44] for more details.

Figure 2 shows an example of an 8 × 8 matrix that can be recovered (Fig. 2(a)), and one with a deadlock that cannot (Fig. 2(b)). It also shows their corresponding bipartite graphs.

The lost packets can be recovered in many cases. If there is a row or a column missing only one packet, this packet can be recovered as the checksum of the remaining packets. This approach can be applied iteratively until the matrix is fully recovered (as in Fig. 2(a)), or a deadlock has been found (as in Fig. 2(b)). Any deadlock configuration must have at least two losses in each of its rows and columns.

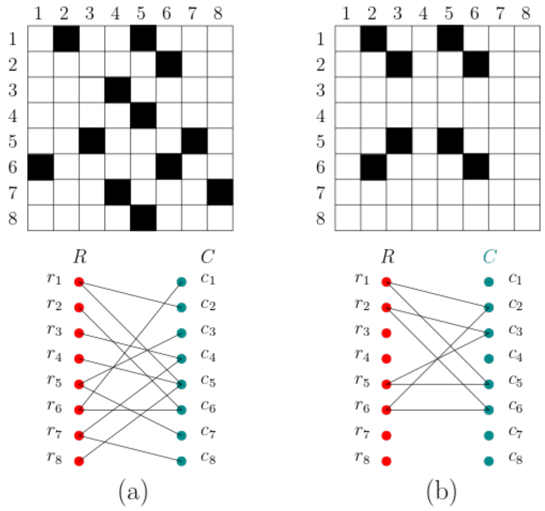

1 2 3 4 5 6 7 8
1
2
3
4
5
6
7
8
R
C
r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8
c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8
(a)
(b)

Figure 2: FEC matrices and their bipartite encoding: (a) a recoverable configuration and (b) a deadlock. In the former case, the matrix can be recovered as follows. In the first iteration, the rows 2, 3, 4, and 8 (or, similarly columns 1, 2, 3, 7 and 8) can be recovered in parallel. In the second iteration, all the remaining columns (or rows) can be recovered in parallel.

Henceforward, the problem of enumerating all the recoverable configurations for a given matrix topology and a given number of lost packets will be reformulated as a classical graph theory enumeration problem. The first step is to characterize these configurations.

Lemma 1. An $n \times m$ FEC matrix $\mathcal{M}$ can be recovered iff the corresponding bipartite graph $G_{\mathcal{M}}$ is a bipartite tree (or a forest of disjoint trees) i.e., if it does not contain any cycle.

Proof. If $G_{\mathcal{M}}$ is a non-trivial bipartite tree with a least one edge (or a forest of such trees), $\mathcal{M}$ must have at least two rows or columns, or at least one row and one column having only one lost packet[1]. These packets can be recovered and the resulting new graph associated to the new matrix still represents a forest of bipartite trees having at least one edge less in total. Thus, all the lost packets in the matrix $\mathcal{M}$ can be recovered iteratively.

Now, if $G_{\mathcal{M}}$ is not a bipartite tree (or a forest of bipartite trees), it must then have a cycle of even length[2]. Let $(r_{i_1}, c_{j_1}, r_{i_2}, c_{j_2}, \cdots, r_{i_l}, c_{j_l}, r_{i_1})$ be such a cycle of minimal length, which implies that $r_{i_s} \neq r_{i_t}$ and $c_{i_s} \neq c_{i_t}$, $\forall s \neq t$. This is a subset of l rows and l columns with at least two losses in each of them. Such a configuration cannot be recovered. □

Corollary 1. For any $n \times m$ FEC matrix $\mathcal{M}$, if there is an $x \times y$ submatrix $\mathcal{M}'$ with at least $x + y$ lost packets, there must be a deadlock in $\mathcal{M}$.

It is straightforward to prove this corollary. If the subgraph $G_{\mathcal{M}'}$ has at least $x + y$ edges, it has to have a cycle since it contains only $x + y$ nodes[3]. Hence, following Lemma 1, it must contain a deadlock. This result gives an upper bound on the maximum number of packets that can be recovered.

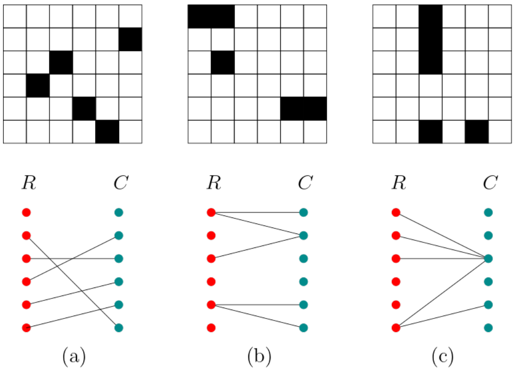


Figure 3: Bipartite forests – without cycles: (a) five edges, five non-degenerate trees with one edge, and a total of seven trees – single nodes are degenerate trees, (b) five edges, two non-degenerate trees, and a total of seven trees (c) five edges, one non-degenerate tree, and a total of seven trees.

Once the recoverable configurations have been properly characterized as distinct forests of bipartite trees, the next step is to determine their number. Let $\mathcal{F}^*_k(n, m)$ denote the number of distinct bipartite forests defined by two labelled sets of n and m nodes, respectively, and k *unrestricted* edges – i.e., without the implicit edge related to the missing lowest-rightmost packet of the FEC matrices. At one end of the spectrum, the k edges can define k *disjoint non-degenerate trees*. At the other end of the spectrum, these k edges can define one *single connected tree*, as illustrated in Fig. 3. Thus, $\mathcal{F}^*_k(n, m) = 0$, if $k \geq n + m$. This

[1] Any tree having at least one edge must have at least two nodes with degree one (a.k.a. *leaf* nodes) – [44], Corollary 2.2.

[2] By definition, an acyclic graph is a tree and a bipartite graph cannot have a cycle of odd length – [44], Theorem 1.2.

[3] In general, an acyclic graph (i.e., a tree) of n nodes can only have $n - 1$ edges – [44], Theorem 2.2.

follows readily from Corollary 1. If the single nodes are considered as degenerate trees without any edge, the total number of trees is constant in all the cases shown in Fig. 3. In general, these k edges can be partitioned to define t *disjoint* trees with respectively $n_1, \cdots, n_t$ nodes of one set, $m_1, \cdots, m_t$ nodes of the other set, and $k_1, \cdots, k_t$ edges[4]. Since the trees are disjoint, $\sum n_i = n$, $\sum m_i = m$ and $n_i + m_i = k_i + 1$, for all i. The total number of edges $\sum k_i = k$ is therefore equal to $n + m - t$, explaining the fact that the number of trees is constant for a given number of edges. Thus, the number of distinct bipartite forests $\mathcal{F}^*_k(n,m)$ with k edges corresponds to the number $\mathcal{T}_{n+m-k}(n,m)$ of distinct bipartite forests with $n + m - k$ *disjoint* trees, which is given by the following result:

Theorem 1. ([45], p. 32) The number of forests composed of t bipartite trees on p labelled red nodes and q labelled cyan nodes is given by

$$\mathcal{T}_t(p,q) = \sum_{h=0}^{t} \frac{(-1)^h}{h!} \frac{p!}{(p-h)!p^h} \frac{q!}{(q-h)!q^h} \Bigg[\sum_{\substack{a+b=t-h\\ a,b\geq 0}} [p(h+b) + q(h+a) - (h+a)(h+b)]\times \binom{p-h}{a}\binom{q-h}{b} p^{q-1-b} q^{p-1-a} \Bigg].$$

This result would be sufficient to get the probability given in Eq. 5, if the FEC matrices were complete $n \times m$ matrices.

Unfortunately, more tools have to be used to deal with the classical FEC matrices. A rooted tree is a tree with a special node that has been defined as the root of the tree. A similar result for *rooted* forests is given by

Theorem 2. ([45], Eq. 4.7) The number of forests composed of r red and s cyan *rooted* bipartite trees on p labelled red nodes and q labelled cyan nodes is given by

$$LR(p,q,r,s) = \binom{p}{r}\binom{q}{s}\Big[ps + qr - rs\Big] p^{q-s-1} q^{p-r-1}.$$

As explained earlier, the number of edges in these forests of $r + s$ trees composed of $p + q$ nodes is given by $p + q - (r + s)$.

An obvious corollary of this lemma can be stated as follows

Corollary 2. The number of forests composed of bipartite trees rooted at some *fixed* r red and s cyan *roots* on p labelled red nodes and q labelled cyan nodes is given by

$$LR^*(p,q,r,s) = [ps + qr - rs] p^{q-s-1} q^{p-r-1}.$$

Now is the time to drop the assumption that the FEC matrix is a full matrix with the extra lowest-rightmost packet. Let $\mathcal{F}_k(n,m)$ denote the number of recoverable configurations for a *classical* FEC matrix of the SMPTE 2022-1/5 standards, i.e., the number of distinct bipartite forests defined by two labelled sets of n and m nodes, respectively, and k unrestricted edges plus the implicit edge corresponding to the missing lowest-rightmost packet of the matrix. Obviously, $\mathcal{F}_k(n,m) < \mathcal{F}^*_{k+1}(n,m)$. The recoverable configurations with k lost packets plus the extra implicit one is a subset of all the potential recoverable configurations with $k + 1$ lost packets for the full matrix. As for the unrestricted $\mathcal{F}^*_{k+1}(n,m)$, $\mathcal{F}_k(n,m) = 0$, if $k \geq n+m-1$.

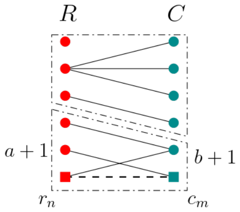


Figure 4: The forest is formed of the tree having the implicit edge (the dashed line joining the square nodes) with some $a + 1$ red nodes and $b + 1$ cyan nodes, and the remaining trees with the other nodes.

If the trivial upper bound on $\mathcal{F}_k(n,m)$ may be sufficient in some cases, an exact formula can be derived with the tools used so far. Thus, $\mathcal{F}_k(n,m)$ can be expressed as

$$\begin{aligned}\mathcal{F}_k(n,m) &= \sum_{a=0}^{k}\sum_{b=0}^{k-a}\binom{n-1}{a}\binom{m-1}{b}\Big[LR^*(a+1,b+1,1,1) \\ &\quad \times\ \mathcal{F}^*_{k-a-b}(n-1-a, m-1-b)\Big]. \qquad (7)\end{aligned}$$

As shown in Fig. 4, the forest of bipartite trees can be decomposed into (i) the tree having the implicit edge with $p = a + 1$ red nodes, and $q = b+1$ cyan nodes, and therefore $a+b$ explicit edges, and (ii) the other trees with the remaining $n - a - 1$ red nodes, and $m - b - 1$ cyan nodes, and the $k - a - b$ explicit edges. In the former case, Corollary 2 can be used to determine the number of these trees. They can be formed by connecting a subtree rooted at the red node r_n (thus, $r = 1$) and a subtree rooted at the cyan node c_m (thus, $s = 1$) with the implicit edge. In the latter case, the number of potential remaining trees is given by $\mathcal{F}^*_{k-a-b}(n - 1 - a, m - 1 - b)$, i.e. the number of distinct bipartite forests defined by two labelled sets of $n-1-a$ and $m - 1 - b$ nodes, respectively, and $k - a - b$ unrestricted edges. Hence, the formula just sums up all the potential partitions of the nodes, for any possible values of a and b. Unfortunately, no closed-form formula can be derived for this summation.

4.3. Simple Deadlock Configurations

Knowing the probability of having a deadlock configuration with a given number of losses is not sufficient. In many cases, the underlying structures of these configurations have to be determined. In this section, some lower bounds on the number of deadlock configurations with some simple structures are presented. Examples of such configurations are 4-cycles (i.e., two rows and two columns forming a rectangle) as shown in Fig. 5, and complete $K_{2,3}$ bipartite graphs (i.e., two rows and three columns forming two adjacent rectangles) as shown in Fig. 6 – or the symmetric $K_{3,2}$ graphs. These lower bounds can be used either to fill Table 1 and justify the exhaustive approach used in

[4] In the case of a degenerate tree, $k_i = 0$.

this paper, or to give better approximations of lower bounds on the residual packet loss rate in the next section.

The simplest deadlock configurations can be associated to bipartite graphs with a single of $2c$-cycle. As mentioned in the introduction, many prior empirical analyses focused only on deadlock configurations with 4 interblocking losses, i.e., with a 4-cycle. It would be useful to find the number of such simple configurations. Austin [46] has presented such a result for generic bipartite graphs. This approach has to be adapted to deal with bipartite graphs with the presence of the implicit edge.

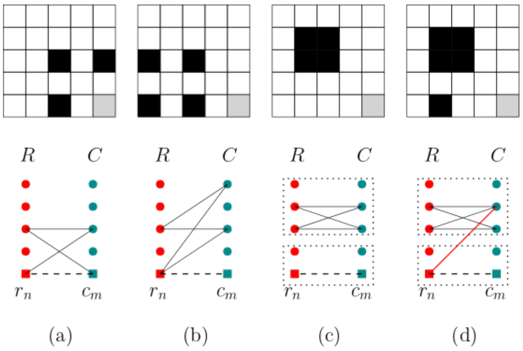


Figure 5: The four cases for the $2c$-cycle with respect to the implicit edge corresponding to the lowest-rightmost corner of the FEC matrix: (a) the cycle includes the implicit edge – $bg_{5a}(p,q,c)$ cases, (b) the cycle is incident to the implicit edge – $bg_{5b}(p,q,c)$ cases, (c) the cycle and the implicit edge are in two different connected components – $bg_{5c}(p,q,c)$ cases, and finally (d) the cycle and the implicit edge are in the same connected component (but not covered by the first two cases) – $bg_{5d}(p,q,c)$ cases.

In order to determine the number of labelled connected bipartite graphs on p red nodes and q cyan nodes with *one* cycle of length $2c$, knowing there is an implicit edge connecting the two given (squared) nodes, the following approach can be used

- Select the c red , and the c cyan nodes for the cycle and determine the number of potential cycles.
- Determine the number of labelled bipartite forests rooted on these $2c$ nodes of the cycle, or on the nodes of the implicit edge with the remaining nodes – Corollary 2.

If the implicit edge *is part of* the $2c$-cycle (i.e., Fig. 5 (a)), there are only c rooted red nodes and c rooted cyan nodes. In this case, the number of labelled connected bipartite graphs on $p \geq c$ red nodes and $q \geq c$ cyan nodes with one cycle of length $2c$, which is denoted $bg_{5a}(p,q,c)$, is given by

$$\begin{aligned} bg_{5a}(p,q,c) &= \binom{p-1}{c-1}\binom{q-1}{c-1}(c-1)!(c-1)!LR^*(p,q,c,c) \\ &= \frac{p!q![c(p+q-c)]p^{q-c-2}q^{p-c-2}}{(p-c)!(q-c)!}. \end{aligned} \tag{8}$$

Given the selected nodes of the $2c$-cycle, the number of rooted forests is given by $LR^*(p,q,c,c)$. These forests are composed of $(p+q)-2c$ edges. On the other hand, the $2c$-cycle composed of the selected roots has obviously $2c-1$ edges. Thus, the number of edges in these connected components is $(p+q-2c)+(2c-1) = p+q-1$.

Similarly, if the implicit edge is *incident* to the $2c$-cycle (i.e., Fig. 5 (b)) , the number of labelled connected bipartite graphs on $p \geq c$ red nodes and $q \geq c+1$ cyan nodes with one $2c$-cycle, which is denoted $bg_{5b}(p,q,c)$, is given by,

$$\begin{aligned} bg_{5b}(p,q,c) &= \binom{p-1}{c-1}\binom{q-1}{c}\frac{c!(c-1)!}{2}LR^*(p,q,c,c+1) \\ &= \frac{p!q![c(p+q-c-1)+p]p^{q-c-3}q^{p-c-2}}{2(p-c)!(q-1-c)!}. \end{aligned} \tag{9}$$

There are only c rooted red nodes and but $c+1$ rooted cyan nodes, as shown in Fig. 5 (b). The symmetric case, which is denoted $bg_{5bS}(p,q,c)$, has to be considered. In both cases, the number of edges in these connected components is $p+q-1$.

Now, if the implicit edge is *not connected* to the cycle of length $2c$ (i.e., Fig. 5 (c)), the number of labelled *disconnected* bipartite graphs on $p \geq c+1$ red nodes and $q \geq c+1$ cyan nodes with one connected component containing the cycle of length $2c$ and another one composed of a tree containing the implicit edge, which is denoted $bg_{5c}(p,q,c)$, is given by

$$\begin{aligned} bg_{5c}(p,q,c) &= \binom{p-1}{c}\binom{q-1}{c}\frac{c!(c-1)!}{2}LR^*(p,q,c+1,c+1) \\ &= \frac{p!q![(c+1)(p+q-c-1)]p^{q-c-3}q^{p-c-3}}{2c(p-1-c)!(q-1-c)!}. \end{aligned} \tag{10}$$

There are $c+1$ rooted red nodes and $c+1$ rooted cyan nodes (i.e., the nodes of the cycle plus the ones of the implicit edge), as shown in Fig. 5 (c). The number of edges in these non connected components is $p+q-2$.

To conclude this first part of the analysis, the final configurations to be considered are shown in Fig. 5 (d). These configurations correspond to connected bipartite graphs composed of a cycle of length $2c$, and the implicit edge – assuming that the implicit edge is used to connect the trees. In these cases, the implicit edge does not belong to or is incident to the cycle (as in the first two cases). Their number is denoted $bg_{5d}(p,q,c)$. These configurations can be constructed as follows. The set of red nodes is partitioned into a set of a nodes associated to the implicit edge, and a set of the remaining $p-1-a$ node associated to the $2c$-cycle. The set of cyan nodes is partitioned similarly. Once the nodes are split and formed a connected component containing the implicit edge, and one containing the cycle – similarly as in Fig. 4, these two components have to be connected by connecting either the red node r_n with any of the $q-b-1$ opposite cyan nodes, or similarly the cyan node c_m with any of the $p-a-1$ opposite red nodes of the other component. These configurations can therefore be enumerated as follows

$$\begin{aligned} bg_{5d}(p,q,c) = \sum_{a=0}^{p-c-1}\sum_{b=0}^{q-c-1}\binom{p-1}{a}\binom{q-1}{b}LR^*(a+1,b+1,1,1) \\ \times\frac{c!(c-1)!}{2}\times LR(p-1-a,q-1-b,c,c) \\ \times(p-a-1+q-b-1). \end{aligned} \tag{11}$$

The number of edges in these connected components is $p+q-1$.

With the above results, the number of simple (but large) deadlock configurations with 5, 7 or 9 interblocking losses – thus, with the missing lowest-rightmost packet – can be determined with Eq. 8. Similarly, the number of deadlock configurations with one cycle of length 6, 8 or 10 can be determined with Eqs. 9, 10, and 11. These equations have been used to fill Table 1 in the introduction and show the importance of considering *all* the deadlock configurations to obtain accurate evaluation of the performance of the FEC technique.

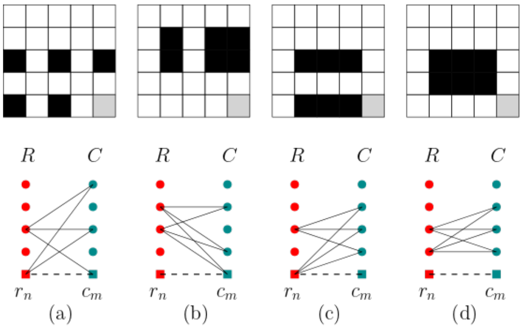


Figure 6: The four cases of a complete $K_{2,3}$ bipartite graph with respect to the implicit edge corresponding to the lowest-rightmost corner of the FEC matrix.

To conclude this section, the last cases needed in the coming section to establish more accurate bounds on the effective packet loss rates are composed of configurations with three nodes fully connected to two nodes (i.e., a $K_{2,3}$ or $K_{3,2}$ bipartite subgraph) as presented in Fig. 6. There are strong relationships between these cases and the ones presented in Fig. 5. For example, the number of labelled connected bipartite graphs on p red nodes and q cyan nodes with a $K_{2,3}$ blocking configuration with the implicit edge (i.e., Fig. 6 (a)) is given by $\frac{2}{c!(c-1)!}$ ×Eq. 9. The correcting factor corresponds to the fact that there is only one way to connect $K_{2,3}$. The other cases are treated similarly.

4.4. Bounds on the Residual Packet Loss Rate

There are two diametrically opposed approaches to deal with packet losses. At one end of the spectrum, if there are still some missing packets after the correcting phase of the FEC method, one drastic approach is to discard all the packets associated to the transmitted matrix. A more realistic one would simply consider the unrecovered packets of the transmitted matrix. Unfortunately, the expected number of these packets is very hard to estimate in general. In the worst case, none of the loss packets can be recovered. In the best one, only three packets cannot be recovered – this corresponds to the minimal deadlock configuration with the lowest-rightmost missing packet. Thus, looking back at Eq. 6, some bounds on the residual packet loss rate RPLR_{2D} can be derived. Once the probability of having a deadlock given k errors, and the probability of having k errors have been expanded with Eq. 5 and Eq. 4, respectively, the pessimistic upper bound $\mathrm{RPLR}^{\mathrm{UB}}_{2D}$ is given by

$$\mathrm{RPLR}^{\mathrm{UB}}_{2D} = \sum_{k=3}^{nm-1}\frac{k}{nm-1}\left[1-\frac{\mathcal{F}_k(n,m)}{\binom{nm-1}{k}}\right]\binom{nm-1}{k}\rho^k(1-\rho)^{nm-1-k}. \quad (12)$$

This bound discards totally the error correction capability of the FEC method. It assumes that none of the k lost packets can be recovered.

On the other hand, the optimistic lower bound $\mathrm{RPLR}^{\mathrm{LB}}_{2D}$ is given by

$$\mathrm{RPLR}^{\mathrm{LB}}_{2D} = \sum_{k=3}^{nm-1}\frac{3}{nm-1}\left[1-\frac{\mathcal{F}_k(n,m)}{\binom{nm-1}{k}}\right]\binom{nm-1}{k}\rho^k(1-\rho)^{nm-1-k}. \quad (13)$$

This bound assumes the presence of the simplest unrecoverable configuration possible with only three lost packets, i.e, a 4-cycle deadlock involving the bottom-rightmost missing packet.

However, the number of deadlock configurations with small numbers of unrecovered packets can be derived from the equations given in the previous section. For example, the only deadlock configurations with *three* unrecovered losses are described in Fig. 5(a), while the ones with *five* unrecovered losses are described in generalizing Fig. 5(a) to have a 6-cycle with the implicit edge, i.e. by picking two extra red nodes and two extra cyan nodes instead of only one, and connecting them with the nodes r_n and c_m, and the $K_{2,3}$ complete graph in Fig. 6(a) with the $K_{3,2}$ symmetric case. On the other hand, the deadlock configurations with *four* unrecovered losses are described in Fig. 5(b-c), while the ones with *six* unrecovered losses are described in generalizing Fig. 5(b-d) to have a 6-cycle, and the complete $K_{2,3}$ graph in Fig. 6(b-d). Above six losses, the enumeration is too complex. For example, there can be two interlocking components with four losses.

Improved lower $\mathrm{RPLR}^{\mathrm{ILB}}_{2D}$ and upper $\mathrm{RPLR}^{\mathrm{IUB}}_{2D}$ bounds can be derived by isolating the special cases with few unrecovered packets. First, notice that $\mathcal{F}_k(n,m) = 0$, if $k \geq n+m-1$. The summations can therefore be split into two. Then, for $k \leq n+m-2$, the probability of having a deadlock with k packet losses $1-\mathcal{F}_k(n,m)/\binom{nm-1}{k}$ can be decomposed into disjoint cases of having only $i \leq k$ unrecovered packets out of the k losses. The probabilities of these cases are denoted $p^{n,m}_{i,k}$. Unfortunately, these probabilities can only be defined for $i \leq 6$. The probability $p^{n,m}_{7+,k}$ simply denotes the probability of having at least seven unrecovered packets. This probability is given by $1-\sum_{i=3}^{6} p^{n,m}_{i,k}$. Thus, the bounds can be refined as follows

$$\sum_{k=3}^{n+m-2}\left[\sum_{i=3}^{6}\frac{i}{nm-1}p^{n,m}_{i,k}+\frac{\beta}{nm-1}p^{n,m}_{7+,k}\right]\binom{nm-1}{k}\rho^k(1-\rho)^{nm-1-k} + \sum_{k=n+m-1}^{nm-1}\frac{\beta}{nm-1}\binom{nm-1}{k}\rho^k(1-\rho)^{nm-1-k}.$$

The parameter β is either equal to seven for the lower bound, or to k for the upper bound.

For instance, the probability $p^{n,m}_{3,k}$ of having three unrecovered losses is given by the configurations in Fig. 5(a). Thus,

$$p^{n,m}_{3,k} = \frac{1}{\binom{nm-1}{k}}\sum_{p=1}^{\min\{k,n\}}\sum_{q=1}^{\min\{k,m\}}\binom{n-1}{p-1}\binom{m-1}{q-1}\times bg_{5a}(p,q,2)\times\mathcal{F}^*_{k-(p+q-1)}(n-p,m-q).$$

This probability depends on the number of configurations obtained by selecting $p-1$ red , and $q-1$ cyan nodes to build a connected component with a 4-cycle (with the implicit edge), which have $p + q - 1$ edges (or losses). The number of these components is given by the expression $bg_{5a}(p, q, 2)$. Then, the remaining $k - (p + q - 1)$ edges connect the remaining nodes to form a bipartite forest. The number of forests is given by the expression $\mathcal{F}^*_{k-(p+q-1)}(n - p, m - q)$. Thus, multiplying these two numbers give the number of bipartite graphs with a 4-cycle for the given $p - 1$ red , and $q - 1$ cyan nodes.

Next, the probability $p_{4,k}$ is given by

$$\begin{aligned} p^{n,m}_{4,k} \quad = \quad & \frac{1}{\binom{nm-1}{k}} \sum_{p=1}^{\min\{k,n\}} \sum_{q=1}^{\min\{k,m\}} \binom{n-1}{p-1}\binom{m-1}{q-1} \times \\ & \big[(bg_{5b}(p, q, 2) + bg_{5bS}(p, q, 2) + bg_{5d}(p, q, 2)) \times \\ & \mathcal{F}^*_{k-(p+q-1)}(n - p, m - q) + \\ & bg_{5c}(p, q, 2) \times \mathcal{F}^*_{k-(p+q-2)}(n - p, m - q)\big]. \end{aligned}$$

Since the configurations described in Fig. 5 (b) and (d) are composed of $p+q-1$ edges, and the ones described in Fig. 5 (c) are composed of only $p + q - 2$ edges, they have to be treated separately. Otherwise, this probability is defined as the previous one. The probabilities $p^{n,m}_{5,k}$ and $p^{n,m}_{6,k}$ can be obtained similarly.

5. RPLR Approximations for the Random Model

Approximations of the residual packet loss rates $\mathrm{RPLR}_{\mathrm{1D}}$ and $\mathrm{RPLR}_{\mathrm{2D}}$ can be derived for low packet loss rates. These bounds can be useful to design FEC-based systems by providing simple yet accurate performance expressions for various FEC configurations. Intuitively, the leading term of the summation defining any of these bounds dominates the remaining of the summation, if the packet loss rate ρ is small enough. Hence, they can be approximated by expressions of the form $\alpha\rho^{\gamma}$, for some fixed constants $\alpha, \gamma \geq 1$, depending on the minimum size of nonrecoverable configurations, and the parameters of the FEC methods. Such approximations yield linear relationships of slope γ between log(RPLR) and $\log \rho$.

5.1. *Approximation for* $\mathrm{RPLR}_{\mathrm{1D}}$

The one-dimensional case is first considered to show how the approximations can be obtained. The second line of Eq. 3 can be rewritten as follows – with the same convention as before where $n = D + 1$ – by extracting the first term of the summation

$$(n - 1)\rho^2(1 - \rho)^{n-2} + \rho \sum_{i=2}^{n-1} \tbinom{n-1}{i}\rho^i(1 - \rho)^{n-1-i}.$$

If ρ is small enough (e.g., $\rho = \frac{\epsilon}{n-2}$, for a fixed constant $\epsilon < 1$), the first term tends toward[5] $e^{-\epsilon}(n - 1)\rho^2$. On the other hand, since

$$\frac{\binom{n-1}{i}\rho^i(1 - \rho)^{n-1-i}}{\binom{n-1}{i+1}\rho^{i+1}(1 - \rho)^{n-2-i}} = \left[\frac{i + 1}{n - 1 - i}\right]\left[\frac{1 - \rho}{\rho}\right] > 2, \text{for } i \geq 1,$$

the series converges faster than the geometric series $\sum_{i=1} 2^{-i}$, which tends toward 2×2^{-1} i.e. twice its first term. Thus, $\mathrm{RPLR}_{\mathrm{1D}}$ can be approximated by $2 \times e^{-\epsilon}(n - 1)\rho^2$.

If even smaller probabilities (e.g., $\rho = \frac{1}{n^c}$, for $c > 1$) were considered, the first term would become even more predominant. Using the same approach, the first term converges simply to $(n - 1)\rho^2$, whilst the remaining of the series is bounded by $1 + \frac{1}{n^c-1}$ times the first term, since the ratio of two consecutive terms is at least n^{c-1}. Hence, the residual rate $\mathrm{RPLR}_{\mathrm{1D}}$ can be approximated by

$$\delta^{1d}_{\rho}(n - 1)\rho^2 \text{ for low PLR values,}$$

where the constant $\delta^{1d}_{\rho} = 2e^{-\epsilon}$, in the former case, and $\delta^{1d}_{\rho} = 1 + \frac{1}{n^c-1}$ (or simply 1), in the latter one.

5.2. *Approximations for* $\mathrm{RPLR}_{\mathrm{2D}}$

For the two-dimensional case, the pessimistic upper bound $\mathrm{RPLR}^{\mathrm{UB}}_{\mathrm{2D}}$ given by Eq. 12 is given by

$$\rho \sum_{k=2}^{nm-2} \tbinom{nm-2}{k}\rho^k(1 - \rho)^{nm-2-k}\left(1 - \frac{\mathcal{F}_{k+1}(n,m)}{\binom{nm-1}{k+1}}\right). \tag{14}$$

Since the number of deadlock configurations defined by three lost packets is $\binom{n-1}{1}\binom{m-1}{1}$, i.e, all the opposite corners to the lowest-rightmost corner, $\mathcal{F}_3(n, m) = \binom{nm-1}{3} - (n - 1)(m - 1)$, and the first term of Eq. 14 is given by

$$\begin{aligned} \rho \times \binom{nm-2}{2}\rho^2(1 - \rho)^{nm-4} \times \frac{(n - 1)(m - 1)}{\binom{nm-1}{3}} \quad & = \\ 3(1 - \rho)^{nm-4}\left[\frac{(n - 1)(m - 1)}{nm - 1}\right]\rho^3. & \end{aligned}$$

As seen earlier, if ρ is small enough, the first term tends toward $3e^{-\epsilon}\frac{(n-1)(m-1)}{nm-1}\rho^3$ (if $\rho = \frac{\epsilon}{nm-4}$, for $\epsilon < 1$), or even $3\frac{(n-1)(m-1)}{nm-1}\rho^3$ (if $\rho = \frac{1}{(nm)^c}$, for $c > 1$).

The last step is to bound the convergence of the summation. Without a good approximation of the number of recoverable configurations $\mathcal{F}_k(n, m)$, it is hard to show that it converges rapidly. However, it is simple to validate empirically that the ratio of any two consecutive terms of Eq. 14 is greater than 2, if $\rho = \frac{0.25}{nm-1}$. Similarly, it is greater than $(nm)^{c-1}$, if $\rho = \frac{0.25}{(nm)^c}$. As in the one-dimensional case, the upper bound converges toward $\delta^{2d}_{\rho}\frac{(n-1)(m-1)}{nm-1}\rho^3$, where $\delta^{2d}_{\rho} = 2 \times 3e^{-\epsilon}$ in the former case, and $\delta^{2d}_{\rho} = 3\left(1 + \frac{1}{(nm)^c-1}\right)$ in the latter one.

Remark that the leading term of Eq. 13 defining the lower bound $\mathrm{RPLR}^{\mathrm{LB}}_{\mathrm{2D}}$ is the same as the one defining the upper bound $\mathrm{RPLR}^{\mathrm{UB}}_{\mathrm{2D}}$. Hence, the lower bound is at least $3e^{-\epsilon}\frac{(n-1)(m-1)}{nm-1}\rho^3$, or $3\frac{(n-1)(m-1)}{nm-1}\rho^3$, depending on how small ρ is. Thus, for low PLR values, the bounds for packet loss rates can be written as

$$\mathrm{RPLR}^{\mathrm{UB}}_{\mathrm{2D}} \approx \mathrm{RPLR}^{\mathrm{LB}}_{\mathrm{2D}} \approx \delta^{2d}_{\rho}\tfrac{(n-1)(m-1)}{nm-1}\rho^3.$$

The impact of the matrix dimensions on both the upper and the lower approximations corresponds simply to the code rate of the given FEC configuration. Such bounds can be easily used to analyse large FEC configurations for which the exact expectation bounds or simulation cannot be done simply due to the magnitude of the numbers involved.

[5] From elementary calculus, $\frac{1-1/x}{e} \leq \left(1 - \frac{1}{x}\right)^x \leq \frac{1}{e}$, for any $x > 1$. Thus, $\left(1 - \frac{\epsilon}{n-2}\right)^{\epsilon} \times \frac{1}{e^{\epsilon}} \leq \left(1 - \frac{1}{(n-2)/\epsilon}\right)^{\frac{n-2}{\epsilon} \times \epsilon} \leq \frac{1}{e^{\epsilon}}$. Finally, knowing that $(1-x)^{\alpha} \approx 1-\alpha x$, for $x < 1$ and $\alpha x \ll 1$, $(1 - \frac{\epsilon}{n-2})^{\epsilon} \approx 1 - \frac{\epsilon^2}{n-2}$, implying that the left-hand side of the inequality converges toward $1/e^{\epsilon}$.

5.3. *Approximation for* $\mathrm{RPLR}_{2D}^{\mathrm{Full}}$ *of improved system*

At this point, it is interesting to come back on the Pro-MPEG COP3's AL-FEC for which the lowest-rightmost packet of the matrix is not transmitted. Due to this fact, three lost packets are sufficient to have a deadlock configuration, which involves a data packet and the FEC packets on its row and column. If the lowest-rightmost packet of the FEC matrix were included, this minimal number would increase to four. The extra FEC packet would correspond to the XOR of all data packets. By following the same approach used in the previous cases, the leading terms of adapted Eq. 12 and 13 defining the upper and the lower residual packet loss rates for the enhanced FEC method could be approximated by $e^{-\epsilon}(n-1)(m-1)\rho^4$ (if $\rho = \frac{\epsilon}{nm-4}$, for $\epsilon < 1$), or simply $(n-1)(m-1)\rho^4$ (if $\rho = \frac{1}{(nm)^c}$, for $c > 1$).

Similarly, the summation defining the pessimistic upper bound converges as fast. The same factors can be used to multiply the first term. Hence, the main differences on these equations are (i) the summations start at $k = 4$, (ii) the number of recoverable configurations with four lost packets is $\binom{n}{2}\binom{m}{2}$, and obviously (iii) the number of *unrestricted* trees is used to determine the probability of deadlock configurations. Adding this extra packet to the Pro-MPEG COP3 matrix has a tremendous impact on the performance. For example, the bounds on the residual packet loss rates of this method tend toward $e^{-\epsilon}(n-1)(m-1)\rho^4$ The RPLR is thus reduced by a factor proportional to $(nm\rho)$. In this case, the dimensions of the FEC matrices would have a significant impact.

6. Theoretical Bounds of $\mathbf{RPLR_{2D}}$ for the Burst Model

Although a thorough performance analysis of the 2D FEC method under bursts of packet losses is beyond the scope of this paper, this subsection presents an overview of the challenges it entails as well as some directions to solve this highly complex problem. However, the main results derived in Section 4 for random packet losses can be reused in this context. First, in order to study burst packet losses, an appropriate burst model reflecting the behaviour of the network of interest must be selected. Many models have been proposed in the literature. They include the *two-state Markov chain* model, also known as the *Simple Gilbert* model, the *Gilbert* model and the *Gilbert-Elliot* model [47]. As illustrated in Fig. 7, these models are defined by two-state machines with a *good* state (G) and a *bad* state (B), and the probabilities of transitioning between them. However, their main differences come from the assumptions made on the packet losses happening in these states. In the Simple Gilbert model, the packets are never lost in the good state, and always lost in the bad one. In the Gilbert model, the packets are only lost uniformly and independently with a given probability in the bad state, whilst in the Gilbert-Elliot model, they can be lost in both states. Characterizing these different models is rather challenging, especially the last most generic one.

Another important problem is the specification of the parameters characterizing the network performance. For instance, Díaz *et al.* [19] have performed some experiments following the criteria specified in the DVB-IP Phase 1 Handbook (ETSI TS 102 542-3-2) [48]. In their work, the selection of an application context forces some specific constraints in terms of delay, burst model, PLR range, etc. A mixed model considering independent and uniformly distributed losses with PLRs ranging from 10^{-3} to 10^{-6}, and bursts of losses have been studied. Their burst model follows the repetitive electrical impulse noise (REIN) model with a duration of 8 ms, implying that the bursts have a fixed length. This can be seen as the generic Gilbert-Elliot model where $q = 1$ and the transitions from State B to State G happen deterministically after a fixed amount of time. Assuming that bursts are rare events, it is simple to set a lower bound on the number of columns of the 2D FEC matrices. The only remaining parameter to set is the number of rows of that matrix. As it will be seen in Section 7.2, it depends on the maximum tolerable latency, and the overall overhead looked for.

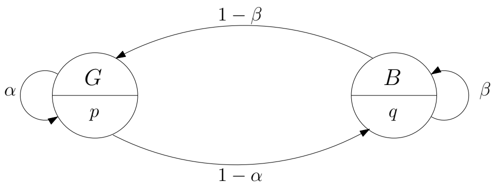


Figure 7: The generic Gilbert-Elliot burst model. In the *G*ood state, the packet losses are independently and randomly distributed with probability p. In the *B*ad state, the packet losses are independently and randomly distributed with probability q – obviously greater than p. Conceptually, the transitions between these states happen just between the packet slots.

Nevertheless, the bounds derived in Section 4 are essential to analyse the performance of the Gilbert model. According to this model, the packet losses are independently and randomly distributed with probability q in the bad state. The average burst length μ_B can be easily estimated. Starting in State B, the state machine will first leave that state at the $(t+1)^{th}$ transition with probability $\beta^t(1-\beta)$. Thus, the average μ_B is given by $\sum_{t\geq 0} t\beta^t(1-\beta) = \frac{\beta}{1-\beta}$[6]. This is independent of the size of the 2D FEC matrix, since it characterizes the network behaviour. More interestingly, the probability that the length of a burst is greater than or equal to $c\mu_B$, for some constant $c \geq 2$, is given by $\sum_{t\geq c\mu_B} \beta^t(1-\beta) = (1-\beta)\left[\sum_{t\geq 0}\beta^t - \sum_{t=0}^{c\mu_B-1}\beta^t\right] = 1 - (1-\beta^{c\mu_B}) = \beta^{c\mu_B}$. As β tends toward one, the value of $\beta^{\beta/(1-\beta)}$ converges rapidly toward $\frac{1}{\mathrm{e}}$[7]. Hence, this probability is about $1/\mathrm{e}^c$. These are elementary results in stochastic process / Markov chain [49].

Now, assuming that having more than one burst in the selected 2D FEC $n \times m$ matrix is a rare event[8], a matrix with at least $m > c\mu_B$ columns would have a small residual packet loss rate of at most $\sum_{t=m+2}^{nm-1} \frac{t}{nm-1}\beta^t(1-\beta)$, under the *Simple Gilbert* model where all the packets are lost in State B. This is the

[6] For $\rho \neq 1$, $\sum_{i=0}^{n-1} i\rho^i = (\rho - n\rho^n + (n-1)\rho^{n+1})/(1-\rho)^2$. Also, $\sum_{i=0}^{n-1}\rho^i = (1-\rho^n)/(1-\rho)$. Both sums have simple limits as n goes to infinity.

[7] This can be shown by proving that the limit of $\frac{\beta}{1-\beta}\ln\beta$ is equal to -1 as β tends toward one (by the L'Hospital rule).

[8] Following the requirements of [48], Díaz *et al.* [19] have targeted a mean time between unrecovered FEC errors of four hours. Since the errors in this context are due to the occurrence of several bursts of packet losses in a matrix, these events are clearly very rare.

case since only bursts of $m + 2$ or more packet losses will create a deadlock. More interestingly, the burst of potential packet losses in State B would spread over consecutive rows, under the *Gilbert* model. Assuming that the burst is of length at most $c\mu_B$, by setting n' as $\lceil \frac{c\mu_B}{m} \rceil + 1$, the submatrix $n' \times m$ should contain all the potential packet losses of that burst. In such a case, Eqs 12 and 13 give the bounds on the residual packet loss rate based on the packet loss rate q. In fact, these bounds have to be corrected by a factor of $\frac{n'}{n}$, if only one burst is present, as the remaining packets of the matrix are not lost. This should give a worst-case scenario for these bounds. Technically, the submatrix may or may not include the last row of the FEC matrix. If it does not, the error correction capability is even higher as seen with the *full* 2D FEC method.

The probability of having only one burst in the FEC matrix can be easily estimated. The average time between two visits to State B is simply given by the average time μ_G spent during a visit to State G. As seen earlier, this is given by $\mu_G = \frac{\alpha}{1-\alpha}$. Similarly, the probability that the time between two visits is smaller than μ_G/c is given by $\sum_{t=0}^{\mu_G/c-1} \alpha^t(1-\alpha) = 1 - (\alpha^{\mu_G})^{1/c}$, which converges towards $1 - \frac{1}{e^{1/c}}$, as α tends toward one.

To conclude this section, consider the following setup to illustrate how to address the problem for the Gilbert model where $\alpha = \frac{9999}{10000}$ and $\beta = \frac{9}{10}$ (thus, an average burst length of 9). In such a case, a burst of potential losses is greater than 90 only with probability of about $1/e^{10}$. This means that if m is about 30, the potential losses would be spread only in four consecutive rows of the 2D FEC $n \times m$ matrix. Thus, Eqs 12 and 13 with $m = 30$ and $n' = 4$ can be used to estimate the RPLR with respect to the parameter q in the range 10^{-2} to 10^{-3}. As expected, the RPLR for the $n' \times m$ submatrix would be in the range 10^{-6} to 10^{-9} and the RPLR for the entire $n \times m$ matrix, after adjustment, in the range $\left(\frac{n'}{n}\right) \times 10^{-6}$ to $\left(\frac{n'}{n}\right) \times 10^{-9}$. Note that if $n = 8$, the matrix would have more than one burst with a probability around $1 - \frac{1}{e^{8\times30/9999}} \approx 0.024$. In these cases, the higher bounds obtained for an entire matrix affected by random packet losses with probability q can be used given that the presented adjustments are applied.

7. Experimental Results

The theoretical bounds derived for the residual packet loss rate $\mathrm{RPLR_{2D}}$ and their approximations are experimentally validated for a wide range of packet loss rates with Monte Carlo simulations in Section 7.1.

The next step is to evaluate the impacts of the FEC matrix dimensions on $\mathrm{RPLR_{2D}}$. In Fig. 10, 1D and 2D FEC configurations having the same code rate (or overhead) are compared to validate and quantify the advantages of the 2D approach. In Fig. 11, similar performance evaluations with different strategies to transmit a given amount of data packets are done. The objective of these analyses is to determine how to select the most adequate configurations of the 2D FEC method.

Finally, in Section 7.3, the bounds and their approximations proposed in this paper are compared with the results presented in the literature.

7.1. Validation of the Theoretical Bounds

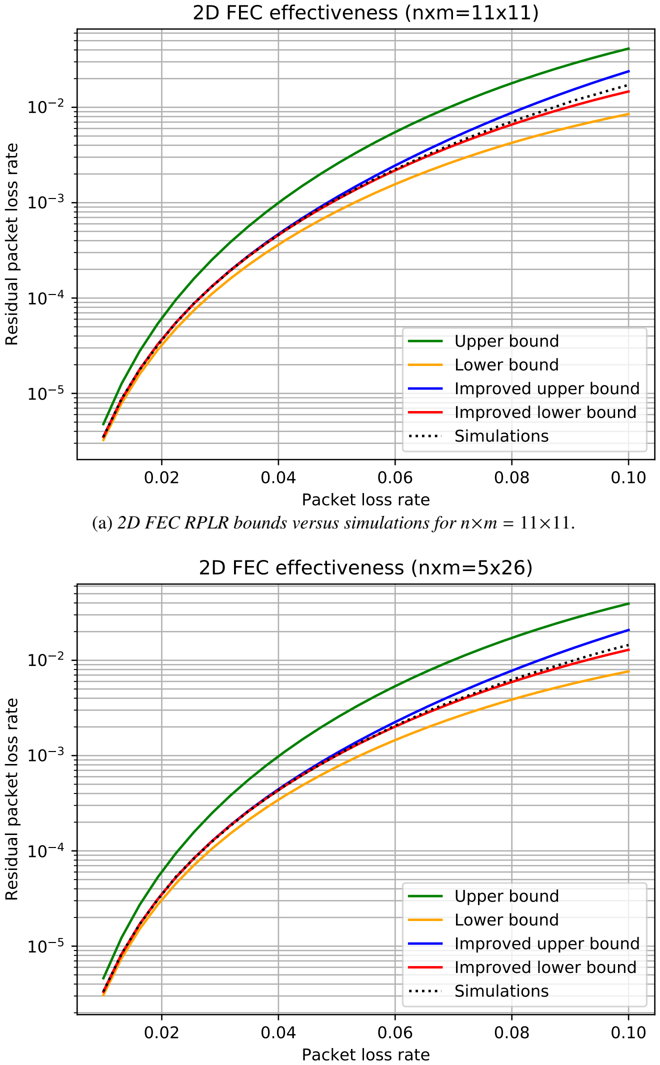
An optimistic lower bound and a pessimistic upper bound on the residual packet loss rate $\mathrm{RPLR_{2D}}$ have been derived from Eq. 6, and refined to obtain tighter bounds by enumerating the deadlock configurations with at most six packet losses. In this section, these bounds are compared to the values observed in Monte Carlo simulations. In these experiments, the typical matrix dimensions as presented in Pro-MPEG COP 3 [18] are used, i.e., 10×10, and 4×25 data matrices. Continuing to refer to matrices by their full dimensions, the dimensions of these matrices are in fact 11×11, and 5×26, respectively.

The Monte Carlo experiments emulate the transmission of packets over a packet erasure channel where independently and randomly distributed packets are lost with probability from $\rho = 10^{-3}$ to 10^{-1}. The correction process of the FEC methods are then used to recover the packet losses of the received matrices. The effective $\mathrm{RPLR_{2D}}$ is computed using Eq. 1 for each of the received matrices. The average of these values is used as an estimate of the overall performance of the FEC methods.

To obtain significant results, it is important to determine the number of the experiments to simulate in terms of the original packet loss rate ρ. Assuming that ρ is relatively small (e.g., $\rho = 10^{-3}$), the probability of having three packet losses in an 11×11 matrix is about 2.5×10^{-4}. On the other hand, the probability of having a deadlock configuration with only three packet losses is about 3.6×10^{-4}. Therefore, the overall probability of such a rare event (i.e., three errors *and* a deadlock) is only 9×10^{-8}. This probability is reduced to 10^{-8}, 9×10^{-10}, and 4×10^{-11}, for four, five and six packet losses, respectively. According to classical Monte Carlo technique theory [50], estimating rare Bernoulli events with 10%-accuracy would need a large number of experiments – at least $\frac{(19.6)^2}{p}$, where p is the success probability of a single Bernoulli event. Therefore, at least 4×10^{11} experiments would be needed to observe deadlocks caused by five packet losses. Hence, the strategy that has been adopted for the simulations presented in the coming sections is either to fix the number of experiments to say 10^6, for high values of packet loss rates, or to $\frac{(19.6)^2}{p}$, where p is the probability of the rare events that have to be observed – i.e., the deadlock configurations with some given packet losses.

Figures 8 and 9 present the derived theoretical bounds for various 2D FEC matrix dimensions with the corresponding values obtained from the Monte Carlo simulations. Observe that the simulation values are always within or overlapping with the bounds derived in this paper. This is more visible at high PLR values (see Fig. 8) where the upper and lower bounds are clearly distinct. As the packet loss rate decreases, the results of the simulations tend toward the lower bounds. This can be easily explained as the number of packet losses involved in the deadlock configurations tends rapidly toward three, which is used in the optimist lower bounds.

Figure 8 clearly shows that the improved bounds are significantly better (tighter) than the original ones at high PLR values. As the packet loss rate decreases, the theoretical upper and lower bounds, their $\delta_\rho^{2d}\rho^3$-approximation derived in Section 5 and the Monte Carlo simulations converge to become one as



(a) *2D FEC RPLR bounds versus simulations for* $n{\times}m = 11{\times}11$.

(b) *2D FEC RPLR bounds versus simulations for* $n{\times}m = 5{\times}26$.

Figure 8: 2D FEC effectiveness: The optimist and pessimist bounds on the RPLR $\text{RPLR}^{\text{LB}}_{\text{2D}}$ and $\text{RPLR}^{\text{UB}}_{\text{2D}}$ versus the improved bounds $\text{RPLR}^{\text{ILB}}_{\text{2D}}$ and $\text{RPLR}^{\text{IUB}}_{\text{2D}}$ compared to Monte Carlo simulations in different configurations at high PLRs.

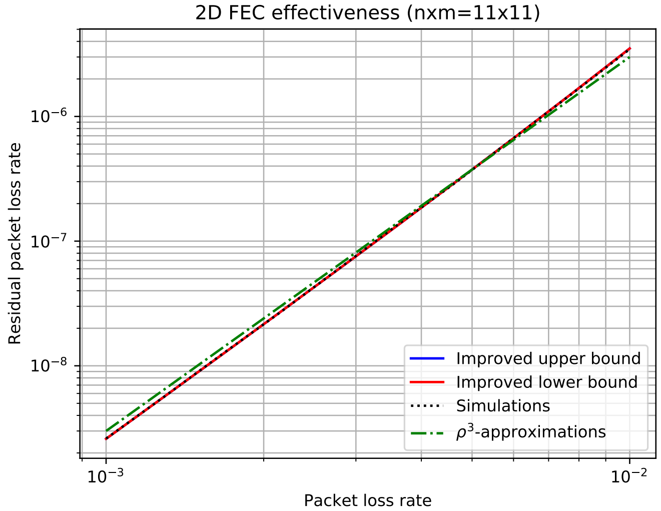


Figure 9: 2D FEC theoretical RPLR bounds compared to simulations for 11×11 matrices.

seen in Fig. 9. Similar observations have been made with other FEC configuration sizes. Hence, the theoretical bounds derived in this paper are in full agreement with the experimental results.

7.2. *Performance Comparisons of the FEC Configurations*

The bounds derived in Section 4, and the related approximations obtained in Section 5 can be used to guide the selection of the adequate FEC parameters to be used in a given network set-up. As already mentioned, three criteria are particularly important: (i) the residual packet loss rate, (ii) the transmission overhead (or, equivalently, the code rate of the FEC correcting code), and, finally, (iii) the overall latency of the communication.

The crucial question for service providers is therefore to determine the most appropriate 2D FEC matrix dimensions to be used according to their network parameters, and their ultimate goals. Assuming that the packet losses are independently and randomly distributed with probability ρ, the approximations of the lower and the upper bounds of the residual packet loss rate can be used to estimate these parameters. If the packet loss rate ρ is of the order of $1/4nm$, the residual packet loss rate is given by $\delta^{2d}\frac{(n-1)(m-1)}{nm-1}\rho^3$, whilst the latency is proportional to nm. The coefficient $\frac{(n-1)(m-1)}{nm-1}$ simply corresponds to the code rate of the selected error correcting code. If the packet loss rate is much smaller than $1/4nm$, the approximation still holds. On the other hand, if the rate is much greater, the approximation is not valid anymore, and the explicit bounds should be used. However, these bounds would be less and less tight as the expected number of packet losses per matrix becomes significant. These cases should be avoided however. Note that in practice, PLRs from 10^{-3} to 10^{-6} are often considered [19, 48].

As the first scenario, consider a network with a packet loss rate $\rho \approx 2{\times}10^{-3}$, i.e., 1 packet loss out of 500 packets. If $4nm \approx 500$, the approximation of the RPLR is $\delta^{2d}\frac{(n-1)(m-1)}{nm-1}{\times}(2{\times}10^{-3})^3$. The overall size of the matrix should be selected to ensure that only very few packet losses occur. Hence, both n and m could be equal to 11, or a more asymmetric matrix could also be used where n is equal to 5 and m is equal to 25, for example. In the former case, the overhead is exactly 20%. In the latter one, the overhead is about 34%. Clearly, the latency is roughly the same in both cases. In Figure 10, the RPLR of these two configurations are presented as well as the packet loss rate of the 1D FEC method with $n = 6$, i.e. with the same overhead as the square matrix, and the packet loss rate if no error correction code is used. The 1D FEC and 2D FEC methods have a linear slope of 2 (as the bound behaves as $\log\delta^{1d}_{\rho}n + 2\log\rho$) and 3 (as the bound behaves as $\log\delta^{2d}_{\rho} + 3\log\rho$), respectively. This represents a significant improvement upon transmissions without any correction mechanism. As clearly seen in the figure, the 2D FEC correcting codes outperform the simple 1D FEC correcting code by a factor of 10^{-3}. On the other hand, both 2D FEC configurations behave similarly with respect to the RPLR.

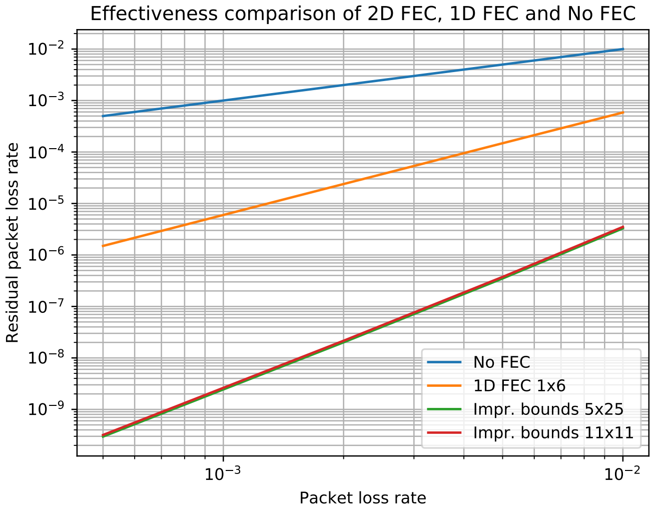


Figure 10: Bounds for ρ = 0.0005, **0.002**, 0.005, and 0.01. The bold probability corresponds to the estimated parameter of the network. The other probabilities are used to show the resilience of the selected configurations. In these cases, the improved lower and upper bounds of the residual packet loss rate coincide in the logarithmic scale set-up.

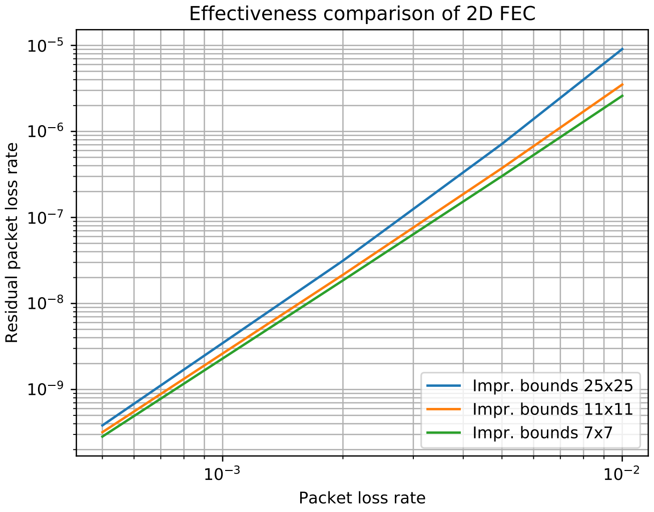


Figure 11: Bounds for ρ = 0.0005, **0.002**, 0.005, and 0.01. Alternatives either to reduce the overall latency, or the overhead with respect to the selected 11×11 matrix.

In such a case, the square configuration minimizing the overhead by a factor of 40% should be selected, without any impact on the overall latency.

At this point, there are two questions. *Can the overall latency be reduced?* This can be done by reducing the total size of the 2D FEC matrix. *Can the code overhead be reduced?* This can be done by augmenting the size of the square matrix. These two objectives are clearly conflicting.

Consider first the objective of reducing the overall latency of the first scenario, whilst keeping the packet loss rate constant. In Figure 11, the selected 11 × 11 FEC matrix is compared to the smaller 7×7 matrix, which has an overhead of $33.\overline{3}\%$. Since the original packet loss rate is still assumed to be 2×10^{-3}, there should be fewer packet losses in the smaller matrix. Ultimately, the RPLR would be smaller, as observed in the figure. But the gain is very marginal compared to the augmentation of the overhead. In order to transmit one million data packets, 1.2 million packets have to be sent with the 11 × 11 matrix, whilst $1.\overline{3}$ million packets is needed with the 7 × 7 matrix. Even if the RPLR is about 2.17×10^{-8} in the former case, and only 1.85×10^{-8} in the latter case, the gain is marginal. Such gain is further decreased as the PLR is reduced. More packets would have to be sent with the smaller matrix anyhow. Hence, the choice between the larger matrix and the smaller one should be mainly driven by the equilibrium between the expected latency, and the overall acceptable overhead in the network.

Now consider the objective of reducing the code overhead to maximize the use of the network. In the same figure, the selected 11×11 FEC matrix is compared to the larger 25×25 matrix, which has an overhead of $8.\overline{3}\%$. Obviously, there should be more packet losses in this larger matrix. Ultimately, the RPLR would be larger, as observed in the figure. But the loss is very marginal compared to the reduction of the overhead. Using the same example, only $1.08\overline{3}$ million packets have to be sent. Even if the RPLR is about 3.14×10^{-8}, the loss is marginal. The same conclusion can therefore be made. The expected latency and the acceptable overall overhead should determine the selection of the 2D FEC configurations – without forgiving that for a given overall size of the matrix, the square one should be picked.

In the next scenario, the initial PLR is 100 times smaller, i.e., 1 packet out of 50 000. If $4nm \approx 50\,000$, the approximation of the RPLR is given by $\delta^{2d\frac{(n-1)(m-1)}{nm-1}} \times (2 \times 10^{-5})^3$. As determined in the first scenario, a very large square matrix such as the 100×100 matrix, which has an overhead of $2.\overline{02}\%$, would be the selected configuration simply based on the RPLR approximations. Such a configuration would have a very large expected latency. A smaller 30 × 30 matrix configuration would reduce the latency by a factor of eleven, whilst augmenting the overhead by a factor of three to 6.9%. On the other hand, a larger 200 × 200 matrix would reduce by half the overhead to 1%.

Naturally, deriving the bounds as in the first scenario is impossible. The number of configurations to be considered is simply too important. A solution to this problem is to restrict the summations in Equation 14 to configurations with at most twenty errors, for example. For the targeted packet loss rate $\rho = 0.00002$, the probabilities of having more than twenty errors are about 10^{-20}, 10^{-32}, and 10^{-53} for the 200 × 200, 100×100, and 30×30 matrices, respectively. For $\rho = 0.00005$, these probabilities should increase to 10^{-13}, 10^{-24}, and 10^{-46}. Bounding the summations do not have any significant impact, at least for the two smallest matrices. Anyhow, increasing the latency by a factor of four to reduce the overhead from 2% to 1% is probably not worth it. In any case, the RPLR should not play a significant role in the decision (see Fig. 12).

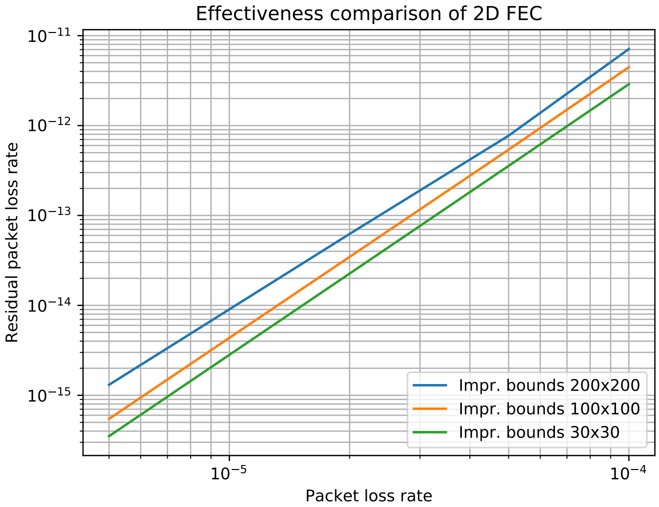


Figure 12: Bounds for $\rho = 0.000005, \mathbf{0.00002}, 0.00005$ and 0.0001. Alternatives either to reduce the overall latency, or the overhead with respect to the selected 100×100 matrix. In order to reduce the computation time, the summations have been computed for at most twenty occurring packet losses.

7.3. *Comparison with Prior Analysis from Literature*

Once the theoretical bounds and their approximations have been validated with Monte Carlo simulations, the final step is to compare them with the analysis presented by Battisti *et al.* [37], which is summarized here. Considering any packet $p_{i,j}$, they proposed to *approximate* the RPLR as $\Pr[p_{i,j} \text{ is not recoverable}] = \Pr[p_{i,j} \text{ has been lost}] \times \Pr[p_{i,j} \text{ is locked}]$, since these two probabilities are independent in the random error model.

The authors enumerated *four disjoint cases* where the single packet $p_{i,j}$ in a $n \times m$ matrix *is not* locked. Knowing that $\Pr[p_{i,j} \text{ is locked}] = 1 - \Pr[p_{i,j} \text{ is not locked}]$, this gives an upper bound on the probability that the packet is unrecoverable. First, assume that row i or column j (or both) does not have other packet losses. This happens with probability $(1-\rho)^{n-1} + (1-\rho)^{m-1} - (1-\rho)^{n+m-2}$. Next, assume there are exactly one extra packet loss in both row i and column j, and at least one of them can be readily recovered. This has a probability $(n-1)\rho(1-\rho)^{n-2} \times (m-1)\rho(1-\rho)^{m-2} \times [(1-\rho)^{n-1} + (1-\rho)^{m-1} - (1-\rho)^{n-m-3}]$ – correcting the original equation presented in [37]. Finally, assume there are exactly one extra packet loss in row i, which can be readily recovered, and at least two extra packet losses in column j. This has a probability $(m-1)\rho(1-\rho)^{m-2} \left[\sum_{k=2}^{n-1} \binom{n-1}{k} \rho^k (1-\rho)^{n-1-k}\right] (1-\rho)^{n-1}$ – correcting the original equation. The last case is symmetric to the previous one by switching row i and column j. Notice there is no special attention to the missing lowest-rightmost packet.

In Fig. 13, the bounds derived in this paper are compared with the result of Battisti *et al.* for a wide range of PLR values. Their simple approximation is relatively accurate for high packet loss rates where $4-$*cycle* deadlocks involving packets other than the lowest-rightmost packet dominate. However, for packet loss rates smaller than 0.1, it is totally inaccurate. In fact, their approximation behaves more like the enhanced full 2D FEC matrix system introduced in Section 5.3. This is not surprising since it ignores the impact of the missing lowest-rightmost packet and only considers deadlock configurations with four packet losses. In contrast, the comprehensive analysis presented in this paper is accurate for all PLR values as proved by Monte Carlo simulations.

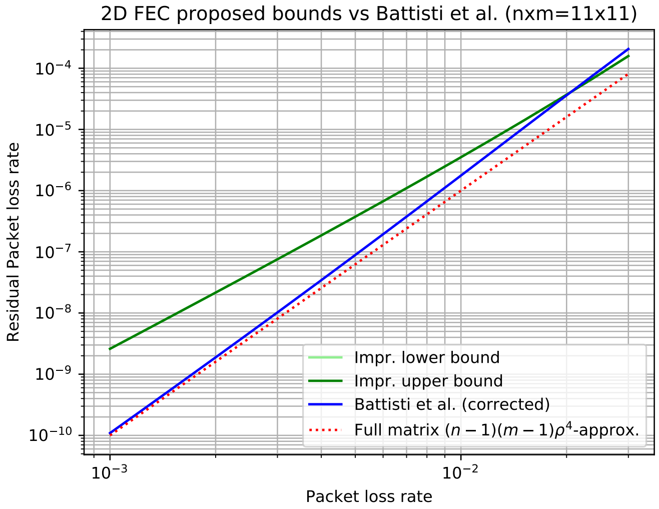


Figure 13: Comparing the bounds derived in this paper with the (corrected) approximation presented by Battisti *et al.* [37] and the full 2D FEC matrix system of Section 5.3 for $n \times m = 11 \times 11$. Note that, at this scale, the improved lower and upper bounds overlap visually.

8. Conclusion

In this work, we rigorously tackle the problem of determining the residual packet loss rate of the two-dimensional FEC method as a function of the matrix dimensions and the packet loss rates. The approach is based on a comprehensive characterisation and enumeration of deadlock configurations. It allows us to derive, for the random error model, tight lower and upper bounds for residual packet loss rates as well as simple approximations on these bounds for low packet loss rate values.

We show that the derived bounds and approximations are in full agreement with the Monte Carlo simulations. We further compare the performance of various matrix dimensions and shapes having the same code rates as well as the same number of data packets. We observe that for low packet loss rate values, the matrix dimension has but a marginal impact on the residual packet loss rates. Therefore, a higher overhead does not translate into a noticeable robustness improvement. At high packet loss rate values, the matrix dimensions play a more significant role. Our analysis shows that (i) a higher overhead has certain noticeable advantages, and (ii) for a given overhead, significantly smaller matrices reduce the residual packet loss rates and the latency.

The powerful framework that has been developed can be used to solve similar problems, such as analysing the performance of: 1) systems in which errors occur in bursts under the Gilbert model, and 2) an enhanced version of the Pro-MPEG COP3 AL-FEC standard using the extra lowest-rightmost packet of the matrix to send the checksum of all the data packets. An unexpected result of this analysis is that the addition of this single packet to the 2D FEC method led to a significant reduction of the residual packet loss rate for low PLR values.

Future work will analyse the overall quality of experience for video communications, which is a very complex multi-faceted problem. It requires setting a range of realistic applicative parameters (video codec and bit rate, packet size, random and/or burst error model, etc.) and selecting an appropriate quality of experience metric to discriminate between low bit rate (lower visual quality) videos with fewer lost packets and high bit rate (higher visual quality) ones with more lost packets. The tools developed in this paper to evaluate the performance of the 1D and 2D FEC for a given bit rate would be essential to evaluate such a metric for various video communications scenarios.

References

[1] S. Kunić and Z. Šego, "Analysis of television technology transformation from SDI to IP production," in *2018 International Symposium ELMAR*, pp. 211–216, Sept. 2018.

[2] K. R. Rao, Z. S. Bojkovic, and D. A. Milovanovic, *Wireless multimedia communications: convergence, DSP, QOS, and security*. CRC Press, 2008.

[3] J. Dale, "VSF video over IP (SMPTE 2022) interoperability demonstration January 2007," *SMPTE Motion Imaging Journal*, vol. 116, no. 10, pp. 456–458, 2007.

[4] S. Sneddon, C. Swisher, and J. Mayzurk, "Engineering design challenges in realization of a very large IP-based television facility," *SMPTE Motion Imaging Journal*, vol. 129, no. 4, pp. 1–9, 2020.

[5] F. Poulin, P. Keroulas, S. Nyamweno, W. Vermost, P. Ferreira, and I. Kostiukevych, "How CBC/Radio-Canada tested media-over-IP devices to build its new facility," *SMPTE Motion Imaging Journal*, vol. 129, no. 4, pp. 35–44, 2020.

[6] T. Witkowski and R. Welsh, "Media cloud fundamentals," *SMPTE Motion Imaging Journal*, vol. 128, no. 9, pp. 26–33, 2019.

[7] D. Luzuriaga, C. Lung, and M. Funmilayo, "Software-based video–audio production mixer via an IP network," *IEEE Access*, vol. 8, pp. 11456–11468, 2020.

[8] T. Koyama, J. Kawamoto, M. Kawaragi, T. Kurakake, and K. Saito, "Implementing 8K vision mixer for cloud-based production system," *SMPTE Motion Imaging Journal*, vol. 128, no. 6, pp. 30–37, 2019.

[9] J. Kawamoto, T. Koyama, M. Kawaragi, K. Saito, and T. Kurakake, "Development of lightweight compressed 8K UHDTV over IP transmission device realizing live remote production," *ITE Transactions on Media Technology and Applications*, vol. 8, pp. 31–39, Jan. 2020.

[10] T. Edwards, "Progress report: ST 2110 update," *SMPTE Motion Imaging Journal*, vol. 127, no. 8, pp. 89–90, 2018.

[11] SMPTE ST 2022-1, "Forward error correction for real-time video/audio transport over IP networks," May 2007.

[12] SMPTE ST 2022-2, "Unidirectional transport of constant bit rate MPEG-2 transport streams on IP networks," May 2007.

[13] SMPTE ST 2022-3, "Unidirectional transport of variable bit rate MPEG-2 transport streams on IP networks," Apr. 2010.

[14] SMPTE ST 2022-4, "Unidirectional transport of non-piecewise constant variable bit rate MPEG-2 streams on IP networks," June 2011.

[15] SMPTE ST 2022-5, "Forward error correction for transport of high bit rate media signals over IP networks (HBRMT)," Feb. 2013.

[16] SMPTE ST 2022-6, "Transport of high bit rate media signals over IP networks (HBRMT)," Oct. 2012.

[17] SMPTE ST 2022-7, "Seamless protection switching of SMPTE ST 2022 IP datagrams," Dec. 2013.

[18] Pro-MPEG Forum, Pro-MPEG Code of practice #3 release 2, "Transmission of professional MPEG-2 transport streams over IP networks," July 2004.

[19] C. Díaz, C. Hellge, J. Cabrera, F. Jaureguizar, and T. Schierl, "Enhancement of Pro-MPEG COP3 codes and application to layer-aware FEC protection of two-layered video transmission," in *2013 IEEE International Conference on Image Processing*, pp. 1598–1602, Sept. 2013.

[20] A. Shokrollahi and M. Luby, "Raptor codes," *Foundations and Trends® in Communications and Information Theory*, vol. 6, p. 213–322, Mar. 2009.

[21] M. Luby, T. Stockhammer, and M. Watson, "IPTV systems, standards and architectures: Part II - application layer FEC in IPTV services," *IEEE Communications Magazine*, vol. 46, no. 5, pp. 94–101, 2008.

[22] ETSI TS 102 034 v1.5.1, "Digital video broadcasting (DVB): Transport of MPEG-2 TS-based DVB services over IP-based networks," May 2014.

[23] ETSI TS 102 993 v1.1.1, "Digital video broadcasting (DVB): Upper layer FEC for DVB systems," Feb. 2011.

[24] T. Nakachi, "MPEG media transport technologies and its application to immersive media communication services," in *2017 7th International Conference on Integrated Circuits, Design, and Verification (ICDV)*, pp. 111–116, Oct. 2017.

[25] ISO/IEC 23008-1, "Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 1: MPEG media transport (MMT)," 2017.

[26] M. Zanaty, V. Singh, A. Bengen, and G. Mandyam, "RTP payload format for flexible forward error correction (FEC)." IETF, RFC 8627, Jul. 2019. [Online]. Available: https://tools.ietf.org/html/rfc8627.

[27] WebRTC. [Online]. Available: https://webrtc.org/.

[28] J. Uberti, "WebRTC forward error correction requirements." IETF, draft-ietf-rtcweb-fec-10, Jul. 2019. [Online]. Available: https://tools.ietf.org/html/draft-ietf-rtcweb-fec-10.

[29] Y. Huo, C. Hellge, T. Wiegand, and L. Hanzo, "A tutorial and review on inter-layer FEC coded layered video streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1166–1207, 2015.

[30] C. Díaz, J. Cabrera, F. Jaureguizar, and N. García, "An extension to the PRO-MPEG COP3 codes for unequal error protection of real-time video transmission," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 927–931, Sept. 2015.

[31] C. Díaz, J. Cabrera, F. Jaureguizar, and N. García, "Optimization of unequal error protection AL-FEC scheme via adapted simulated annealing," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 345–346, Jan. 2016.

[32] C. Díaz, J. Cabrera, F. Jaureguizar, and N. García, "Application-layer FEC scheme configuration optimization via hybrid simulated annealing," *IEEE Transactions on Broadcasting*, vol. 63, pp. 479–493, Sept. 2017.

[33] J. Kawamoto and T. Kurakake, "XOR-based FEC to improve burst-loss tolerance for 8K ultra-high definition TV over IP transmission," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Dec. 2017.

[34] J. Kawamoto and T. Kurakake, "Diagonal XOR-based FEC method to improve burst-loss tolerance for 4K/8K UHDTV transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1983–1994, 2021.

[35] S. Fonnes, "Reducing packet loss in real-time wireless multicast video streams with forward error correction," Master's thesis, University of Oslo, Norway, 2018.

[36] J. Westerlund, "Forward error correction in real-time video streaming applications," Master's thesis, Umeå University, Sweden, 2015.

[37] F. Battisti, M. Carli, E. Mammi, and A. Neri, "A study on the impact of AL-FEC techniques on TV over IP quality of experience," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 86, 2011.

[38] A. Nauman, Y. A. Qadri, M. Amjad, Y. B. Zikria, M. K. Afzal, and S. W. Kim, "Multimedia internet of things: A comprehensive survey," *IEEE Access*, vol. 8, pp. 8202–8250, 2020.

[39] P. Lemonnier, "Safe video contribution and distribution over IP networks," *Proceedings of the Public Broadcasting Service Technology Conference*, 2005.

[40] R. Razavi, M. Fleury, and M. Ghanbari, "Adaptive packet-level interleaved FEC for wireless priority-encoded video streaming," *Advances in Multimedia*, vol. 2009, no. 982867, pp. 1–14, 2009.

[41] K. Kralevska, *Applied Erasure Coding in Networks and Distributed Storage*. PhD thesis, Norwegian University of Science and Technology (NTNU), Norway, 2016.

[42] J. W. Evans and C. Filsfils, *Deploying IP and MPLS QoS for multiservice*

networks: Theory & practice. Elsevier Science, 2010.

[43] A. C. Begen, “RTP payload format for non-interleaved and interleaved parity FEC.” IETF, draft-begen-fecframe-1d2d-parity-scheme-01, Sep. 2008. [Online]. Available: https://tools.ietf.org/html/draft-begen-fecframe-1d2d-parity-scheme-01.

[44] J. A. Bondy and U. S. R. Murty, *Graph theory with applications*. American Elsevier Publishing Company, 1976.

[45] J. W. Moon, *Counting labelled trees*. William Clowes and Sons, 1970.

[46] T. L. Austin, “The enumeration of point labelled chromatic graphs and trees,” *Canadian Journal of Mathematics*, vol. 12, pp. 535–545, 1960.

[47] C. A. G. D. Silva and C. M. Pedroso, “MAC-layer packet loss models for Wi-Fi networks: A survey,” *IEEE Access*, vol. 7, pp. 180512–180531, 2019.

[48] ETSI TS 102 542-3-2, “Digital video broadcasting (DVB):guidelines for the implementation of DVB-IP phase 1 specifications. part 3: Error recovery. subpart 2: Application layer FEC,” May 2011.

[49] N. Privault, *Understanding Markov Chains*. 357, Springer, 2013.

[50] G. Rubino and B. E. Tuffin, *Rare event simulation using Monte Carlo methods*. John Wiley & Sons, 2009.