

Optimum Physical-Layer Frame Size for Maximising the Application-Layer Rateless Code's Effective Throughput

T. Stevens, R. G. Maunder, and L. Hanzo
 ECS, University of Southampton, SO17 1BJ, UK.
 Corresponding author: lh@ecs.soton.ac.uk

Abstract—The tolerable packet-loss ratio of an Internet Protocol (IP) based wireless networks varies according to the specific services considered. File transfer for example must be error free but tolerates higher delays, whereas maintaining a low delay is typically more important in interactive Voice Over IP (VOIP) or video services. Classic Forward Error Correction (FEC) may be applied to the data to provide resilience against bit errors. A wireless IP network provides the opportunity for the inclusion of FEC at the physical, transport and application layers. The demarcation between the analogue and digital domain imposed at the Physical layer (PHY) predetermines the nature of the FEC scheme implemented at the various layers. At the PHY individual packets may be offered FEC protection, which increases the likelihood of their error-free insertion into the protocol stack. Higher layers receive packets that are error free and the purpose of a FEC scheme implemented here is to regenerate any missing packets obliterated for example by the Binary Erasure Channel (BEC) of the IP network's routers. A rateless code may be beneficially employed at a higher Open Systems Interconnection (OSI) layer for replenishing the obliterated packets, but unless the characteristics of the channel are considered, the ultimate rate achieved by such a code may be compromised, as shown in this contribution.

I. INTRODUCTION

Rateless codes, such as the Random Linear Fountain Code (RLFC) [1] and the Luby Transform (LT) code [2], offer packet based FEC codes, which have an initially undefined rate. The latter, in the guise of a Raptor code [3], has gained acceptance into a number of global communications standards [4], [5]. Both schemes are capable of providing a limitless number of parity packets through the random selection and linear combination of subsets of the original source data. Ideally, only the minimum number of additional encoded packets should be sent with the stream, until the receiver acknowledges that the original data has been successfully recovered. This philosophy differs from that of a traditional fixed-rate code, which incorporates a fixed number of parity bits. Hence, if the protection afforded by the selected code rate is insufficient, the original source data will not be recovered. This limitation of fixed-rate codes may be mitigated to some extent by Automatic Repeat Request (ARQ) techniques transmitting redundant redundancy upon request by the receiver.

Typical applications proposed for the family of rateless codes are the transfer of large files and multi-cast communications. In the former case the potentially slower growth of the number of excess parity packets and the adoption of the acknowledgement protocol minimises the total number of packets transmitted. By contrast, in a multi-cast regime the main benefit derives from the random nature of the encoded packets, no packet has any particular significance in the decoding process and the probability of recovering all of the original source data depends solely upon the number of and not on the specific order of the encoded packets received. It is this particular observation that motivates this study. If the Packet Erasure Ratio (PER) ratio of the link is known, then a sufficiently high number of encoded packets may be transmitted that guarantees a specific quality of service agreement based on the successful packet recovery ratio. Further, this approach may obviate the need for packet retransmission and hence potentially dispenses with a feedback link.

In Section II we primarily investigate the compromise that must be made in the selection of the source-packet payload size. We assume that a fixed-length header is prepended to each packet and hence a larger payload will reduce the overhead incurred by the header. Furthermore, we assume having a link subject to bit errors imposed by a wireless channel rather than having only statistical multiplexing induced packet erasures. Owing to the assumption of having independent bit error

events, shorter packets comprising fewer bits are more likely to be error-free upon reception than a larger packet. The adoption of an IP based network means that packets contaminated by bit errors will be erased by the lower OSI layers of the protocol stack and hence will be unavailable to the application layer. Since we include these erased packets in our measure of the overhead, we have to identify the payload size that provides the best compromise and jointly mitigates these two effects. In Section II we introduce the two specific rateless codes analysed in this paper and describe the relationship between the amount of data transmitted in each packet, the underlying channel Bit Error Ratio (BER), and the mean code rate achieved by the coding scheme. Our simulations results are presented in Section III followed by our conclusions.

II. FORWARD ERROR CORRECTION

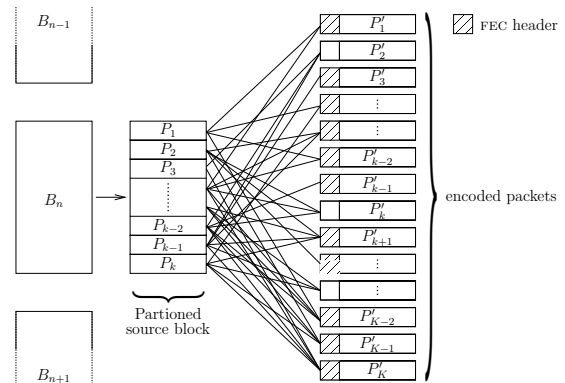


Fig. 1. Block encoding using a non-systematic FEC scheme. Each source block, B_n , is divided into $n_p = k$ equally sized partitions. The FEC header is attached after the selected partitions have been modulo-2 combined.

A. Fountain codes

The source data arrives at the encoder as a block of data as depicted in Fig. 1. When a source block, B_n , is delivered to the encoder, it is divided into n_p partitions of equal size. This set of partitions form the elementary units for the encoding process and should not be confused with the packets of data which are ultimately delivered over the network. The payload of each packet is formed by linearly combining an appropriately selected subset of the n_p partitions.

The encoding process selects d of the n_p partitions for linear combination. This number d defines the degree of the packet. It is drawn randomly from a distribution characterised by a predetermined Probability Density Function (PDF), which is known as the degree distribution. Then d of the n_p partitions are selected at random, with equal probability, to be modulo-2 summed, in order to yield the encoded payload. After encoding an FEC header is attached in order to facilitate the decoding process. It informs the decoder as to which particular partitions were combined to yield the associated payload [6]. Below is a brief discussion of the RLFC and the LT code using the Robust Soliton Degree Distribution (RSDD) [2].

1) *Random linear fountain codes*: The RLFC adopts a binomial degree distribution, $B(n_p, 0.5)$, where the encoder randomly selects with a probability of $p=0.5$, whether a source partition will be included in the bitwise modulo-2 summation that yields the encoded data [1]. At

the decoder the FEC header is used to construct a matrix representing the encoded packets which have been received. A row is assigned to each encoded packet received and the columns correspond to the source partitions contained therein. The symbol identifier extracted from the FEC header, allows the decoder to deduce which particular source partitions were summed to generate each encoded packet and thence to assign a zero or one accordingly. When the modulo-2 inversion of this matrix yields a full column rank, every source partition can be recovered from the collection of encoded packets available to the decoder.

An upper bound [1] on the probability of a block recovery failure after the reception of $(n_p + \epsilon)$ packets is given by

$$\delta(\epsilon) \leq 2^{-\epsilon}. \quad (1)$$

Later in Section III we will consider a Target Block Recovery Ratio (TBRR) of 0.95. From (1) this requires the reception of five excess packets, independent of n_p . The RLFC benefits from a simple coding scheme and the existence of a bound that characterises the number of excess packets required for a certain Block Recovery Ratio (BRR). The main deficiency of this approach is its complexity, since it necessitates the solution of $(n_p + \epsilon)$ linear simultaneous equations.

2) *Luby transform coding*: The encoding of packets within Luby's scheme [2] is again through the modulo-2 addition of a carefully selected subset of source partitions. Here, the partitions are selected according to the RSDD [2], which aims to provide a high probability of recovering the source block, whilst using a decoding process based on a message passing algorithm [1], which has a lower complexity than that required for the RLFC.

B. The normalised overhead

Each of our candidate schemes is evaluated in terms of the normalised overhead imposed. A system that introduces a lower normalised overhead implies that fewer bytes are sent by the source to meet the TBRR at the decoder. It is the ratio of the total data transmitted to the quantity of data recovered from successfully decoded, complete source blocks. Given a header length of n_h and source block size of n_B the normalised overhead, $O(\cdot)$, is evaluated against the payload size, n_d , as follows:

$$O(n_d) = \frac{s(n_d + n_h)}{\beta n_B}, \quad (2)$$

where s is the number of encoded packets that are transmitted and are measured in bytes. This definition facilitates the identification of a compromise between code-rate selection and the robustness of the application to lost packets. The value of $O(n_d)$ will always be less than unity for two reasons; each packet transmitted over the network contains additional protocol headers and the coding scheme requires that a number of excess encoded packets be transmitted. Although the analysis is performed in terms of the $O(n_d)$, many of the results will be presented in terms of $R(\cdot)$, the rate achieved by the code, where we have:

$$R(n_d) = \frac{1}{O(n_d)} = \frac{\beta n_B}{s(n_d + n_h)}. \quad (3)$$

There are other limitations imposed upon the maximum packet size by the implementation of the PHY. For example, the maximum payload of the IEEE802.11 standard is limited to about 2300 bytes, depending upon the configuration considered, and to 1500 bytes for Ethernet. The following analysis imposes no upper limit on the payload's size. Additionally, the FEC schemes considered require that each of the partitions contain an equal number of bits. We present a continuous-valued model that allows the definition of fractional bits and we do not consider the impact of padding the source block which is required in order to provide more flexibility in partitioning the source block. In Section III the consequence of discrete packet sizes is then considered.

Let us now develop the relationships that will act as a guide in selecting the number of data bytes to assign to the payload. We first consider a retransmission scheme which will act as an upper bound on the payload size. Then a general relationship is derived to identify

the optimum payload. The existence of a closed-form tight bound expression for the block recovery Cumulative Distribution Function (CDF) enables us to find an expression for the RLFC. A method of assessing codes that have no such closed-form expression is then discussed. An LT code using a RSDD is analysed using the proposed technique in Section III.

1) *Upper bound due to an idealised retransmission scheme*: The idealised retransmission scheme sends each packet the minimum number of times necessary in order to guarantee their successful reception. This can be achieved by assuming the presence of a zero latency, error-free reverse channel used by the decoder to acknowledge the success or failure of each packet's reception. An alternative interpretation is that the source has full channel knowledge and, therefore, may insert sacrificial dummy packets into the output stream.

Although we are concerned with a FEC scheme operating at a higher OSI layer, where the PER is the natural choice for quantifying the probability of data loss we choose to introduce the BER in order to ensure that the length of the selected payload is appropriately considered. The PER associated with a particular BER, b , depends on the payload size, n_d , as follows:

$$p(n_d) = 1 - (1 - b)^{8(n_d + n_h)}. \quad (4)$$

In this retransmission scheme we consider individual packets, rather than source blocks, hence we have $n_B = n_d$. Furthermore, since the reception of every packet is guaranteed, we have $\beta = 1$ and

$$O_{\text{RTX}}(n_d) = \frac{1}{1 - p(b, n_d, n_h)} \frac{n_d + n_h}{n_d}. \quad (5)$$

We wish to minimise (5) in order to identify the optimum payload, which is achieved by differentiation:

$$\frac{\delta}{\delta n_d} \{O_{\text{RTX}}(b, n_d, n_h)\} = \frac{1 - 8(n_d + n_h) \log(1 - b)}{n_d(1 - b)^{8(n_d + n_h)}} - \frac{n_d + n_h}{n_d^2(1 - b)^{8(n_d + n_h)}}. \quad (6)$$

Setting (6) equal to zero gives an optimum payload of:

$$n'_d = -\frac{\sqrt{2} \sqrt{2n_h^2 \log^2(1 - b) - n_h \log(1 - b)}}{4 \log(1 - b)} - \frac{n_h}{2}. \quad (7)$$

The bound on the payload size against the BER for the benchmark retransmission scheme is presented in Fig. 4.

C. Payload size to minimise the normalised overhead

In this section we examine the selection of the payload size that minimises the normalised overhead incurred, while meeting the TBRR. We first consider a RLFC operating over an error-free link and then extend the analysis to include independent bit errors.

1) *Payload for the random linear fountain code operating over an error-free link*: As noted in Section III-B, the upper bound for the TBRR, (1), determines the minimum number of excess packets that must be transmitted. Even though an error-free link has been assumed, there is no guarantee that any given source block will be recovered. It is important to recognise that the successful recovery of a source block depends also upon the contents of the packets received and not purely upon their number. Given ϵ excess packets and an error-free link associated with $p = 0$, the encoder has to send a total of

$$s = \frac{n_B}{n_d} + \epsilon \quad (8)$$

packets. This results in a normalised overhead of

$$O_{\text{RLFC}}(n_d) = \left(\frac{n_B}{n_d} + \epsilon \right) \frac{n_d + n_h}{\beta n_B}, \quad (9)$$

while the particular payload size, n''_d , which minimises $O_{\text{RLFC}}(n_d)$ is found by differentiating (9) with respect to n_d :

$$\frac{d}{dn_d} O_{\text{RLFC}}(n_d) = -\frac{n_h}{\beta n_d^2} + \frac{\epsilon}{\beta n_B}, \quad (10)$$

and gives the minimum when the payload size is

$$n_d'' = \sqrt{\frac{n_B n_h}{\epsilon}}. \quad (11)$$

In contrast to the idealised retransmission scheme, where increasingly larger payload sizes may be used as the BER falls, the RLFC has a finite optimum payload size of n_d'' for transmission over an error-free link. If sufficient flexibility exists within the system to vary the source block size, n_B , or the header length, n_h , then substitution of (11) into (9) shows that increasing n_B or reducing n_h will reduce $O_{\text{RLFC}}(n_d)$. The effect of both of these changes is to increase the number of partitions, (12). However, if the total header length is determined by the protocols selected and the source block size constrained by the associated delay requirements, then the optimum payload size is fixed.

From (11), the associated number of partitions becomes

$$n_p = \frac{n_B}{n_d''} = \sqrt{\epsilon \frac{n_B}{n_h}}, \quad (12)$$

giving an associated rate of

$$R = \frac{\beta n_B}{(n_p + \epsilon)(n_d'' + n_h)} = \frac{\beta n_B}{(\sqrt{n_B} + \sqrt{\epsilon n_h})^2}. \quad (13)$$

This confirms that we have $R \rightarrow \beta$, as determined by ϵ , when $n_B \rightarrow \infty$. Hence (13) allows us to identify the block size that yields a specific achieved rate, R , for a system:

$$n_B = \frac{\epsilon \beta n_h R}{(1 - \sqrt{R})^2}. \quad (14)$$

The relationship between R and the normalised block size, n_B/n_h , is presented in Fig. 2 when $\beta = 0.95$ ($\epsilon = 5$) and $\beta = 0.999$ ($\epsilon = 10$). The normalised block size that yields an achieved rate, which is half of the maximum attainable rate, i.e. $R = \frac{\beta}{2}$, is given by

$$\frac{n_B}{n_h} \Big|_{R=\frac{\beta}{2}} = \frac{\epsilon}{3 - 2\sqrt{2}}. \quad (15)$$

The values of (15) are indicated by the hollow squares in Fig. 2. For comparison, the variation of the achieved rate as a function of the normalised block size is presented in Fig. 2, when n_d is fixed to ensure that $(n_h + n_d)$ fills a 1500-byte frame. For $n_d = 1418$, $n_h = 82$ (see Section III-A) and $\epsilon = 5$ the block size that results in the maximum achievable rate of $R_{\text{max}} = 0.894$, is $n_B \approx 122\,600$. When the block size assumes any other value, the achieved rate falls. In the system considered previously we had, $n_B = 6360$, $n_h = 82$, $\beta = 0.95$ ($\epsilon = 5$) and a maximum rate of $R_{\text{max}} = 0.605$, which occurs when we assigned $n_d = 322.96$. By contrast, the rate falls to $R = 0.425$, when a 1500-byte frame is employed. More explicitly, for this scenario, an additional overhead of $\approx 42\%$ is imposed.

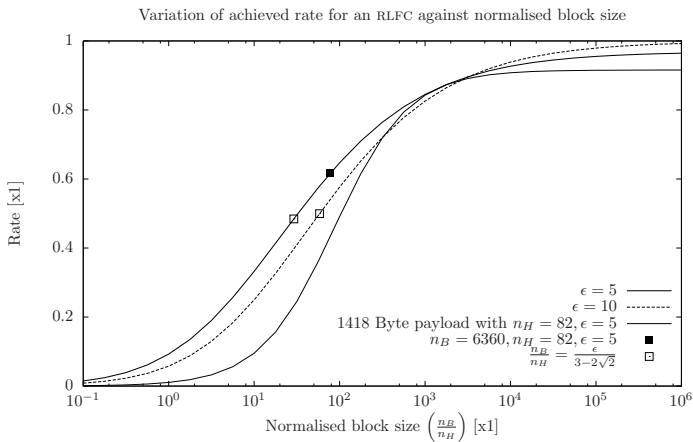


Fig. 2. The achieved rate increases with the normalised block size. The result of adopting a fixed payload, n_d , is shown together with the points corresponding to the rate being half that achieved with an infinite block size.

2) *The general case for operation in the presence of bit errors:* Let the function $\chi(\cdot)$ represent the number of packets that must be transmitted in order to meet the TBRR. Substituting $\chi(\cdot)$ for s in (2) and differentiating gives

$$\frac{\delta}{\delta n_d} O(n_d) = \frac{\chi(\cdot) + (n_d + n_h) \frac{\delta}{\delta n_d} \chi(\cdot)}{\beta n_B}. \quad (16)$$

This expression shows that the minimum normalised overhead depends on both the number of packets that must be transmitted and on the derivative of $\chi(\cdot)$. In the following sections the impact of this dependence will be examined for both the RLFC and for LT code.

3) *Random linear Fountain codes in the presence of bit errors:* To proceed with the analysis of the RLFC we first develop an expression for the number of packets, $\chi(\cdot)$. For a payload of n_d , $\frac{n_B}{n_d}$ packets must be transmitted in addition to the excess packets so, $(\frac{n_B}{n_d} + \epsilon)$ packets must be received in order to meet the target. The number of packets that the encoder transmits must be increased in order to compensate for the prevailing PER, giving

$$\chi(n_d) = \frac{n_B + n_d \epsilon}{n_d (1 - b)^{8(n_d + n_h)}} \quad (17)$$

and a derivative of

$$\frac{\delta}{\delta n_d} \chi(n_d) = -\frac{8 n_d \log(1 - b) (n_B + n_d \epsilon) + n_B}{n_d^2 (1 - b)^{8(n_d + n_h)}}. \quad (18)$$

Combining this result with (16) directly gives the derivative of the normalised overhead as:

$$\begin{aligned} \frac{\delta}{\delta n_d} O_{\text{RLFC}}(n_d, b) &= \frac{n_B + \epsilon n_d}{\beta n_d n_B (1 - b)^{8(n_h + n_d)}} \\ &- \frac{8 \log(1 - b) (n_h + n_d) (n_B + \epsilon n_d)}{\beta n_d n_B (1 - b)^{8(n_h + n_d)}} \\ &- \frac{n_h + n_d}{\beta n_d^2 (1 - b)^{8(n_h + n_d)}}. \end{aligned} \quad (19)$$

The optimum payload that minimises the of the RLFC is found by setting (19) equal to zero and solving for n_d . Also, letting $b \rightarrow 0$ in (19) confirms the result obtained for an error-free link in (10). The optimum payload is found for an RLFC using this result in Section III.

The existence of a closed form expression for a tight bound provides a continuous, differentiable function $\chi(\cdot)$ for the RLFC. In general, this is not the case but corresponding analysis is still tractable, provided that a suitable function can be identified to model the fountain code under consideration.

D. Number of encoded packets required

Our aim is to identify the minimum number of packets, s_{min} , to be transmitted which ensure that the probability of recovering the source block exceeds the TBRR. The decoding process associated with the fountain codes is stochastic in that there is an uncertainty associated with the success of the decoding process, given the availability of r encoded packets. The availability of more encoded packets will aid the probability of recovering the source blocks, provided that they are error-free. It is plausible that this relationship is monotonically non-decreasing with the number of encoded packets available and may be represented by a CDF, $C(r)$.

Under a known steady state packet erasure probability, p , the probability of receiving r packets from the s packets sent is

$$P(r, s, p) = {}^s C_r (1 - p)^r p^{s-r} \quad (20)$$

for independent packet erasure events. Combining (20) with the CDF associated with the packet recovery distribution gives the probability of decoding the received block as

$$P(\text{decode} | r, s, p) = C(r) P(r, s, p) \quad (21)$$

and, may be expressed independent of r as:

$$P_{\text{decode}}(s, p) = P(\text{decode} | s, p) = \sum_{r=0}^s C(r) P(r, s, p). \quad (22)$$

The minimum number of packets to send, s_{\min} , is then

$$s_{\min} \Big|_p = \arg \min_{s: P_{\text{decode}}(s,p) \geq \beta} \sum_{r=0}^s C(r) \mathcal{P}(r, s, p). \quad (23)$$

The PDF of the number of successfully detected packets at the receiver is affected by the PER, which may be modelled by the Binomial distribution $B(s_{\min}, p)$. At low PERs the distribution exhibits only a slight spread and the number of packets to be transmitted only has to be slightly higher than that indicated by $C(r)$. As the PER increases, the distribution spreads, which necessitates an increase in the number of packets sent to guarantee meeting the BRR. Fig. 3 shows the PDF of the number of packets successfully received for the PERs of $p=0.03$ and $p=0.3$. The increase in the PER results in a spread in the distribution of the number of received packets observed at the receiver. As a consequence, the number of packets the transmitter must send increases, according to (23), from 26 to 37 in order to achieve the TBRR of $\beta=0.95$.

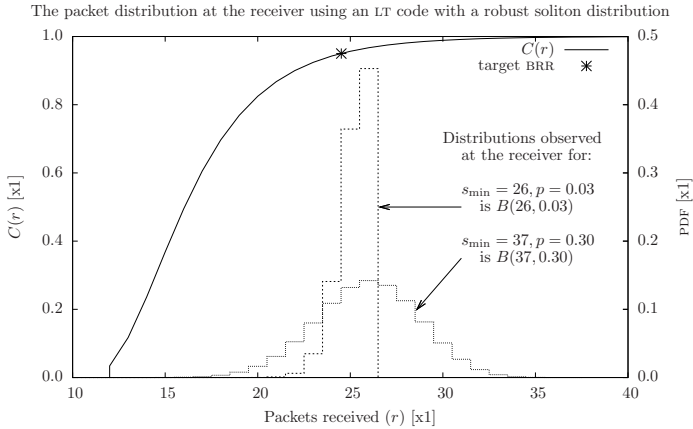


Fig. 3. The number of packets transmitted must be increased to compensate for higher packet loss ratio incurred as the BER increases. The CDF of the BRR for an LT code using the RSDD optimised for $n_p=12$ is shown along with the PDF of the number of received packets for $p=0.03$ and $p=0.3$. The number of packets s_{\min} that had to be sent, (23), were 26 and 37, respectively.

III. RESULTS

A. System description

A source block size of $n_B=6360$ was selected for the transmission of 288×352 pixel Common Intermediate Format (CIF) video frames, at a transmission rate of approximately 0.5 bit/pixel. Section II-B describes the simplifying assumptions made to support the analysis. These additional requirements, namely that all partitions are of the same size and that no padding bits are considered leads to a set representing the values that n_p may take. This set, \mathcal{P} , is reduced so that only payloads larger than the header size, n_h , are considered:

$$\mathcal{P} = \{6, 8, 10, 12, 15, 20, 24, 30, 40, 53, 60\}. \quad (24)$$

A TBRR of $\beta=0.95$ was adopted throughout because a PER of 5% results in a subjectively unobjectionable video quality [7].

The size of each encoded packet of Fig. 1 is affected by the addition of a number of protocol headers as it traverses through the protocol stack. We assumed that each block, which represents a CIF video frame, has the FEC applied at the application layer. The small block FEC code [6] defines an 8-byte FEC header. At the transport layer, this video stream was concatenated with a 12-byte Real Time Protocol (RTP) [8] header and an 8-byte User Datagram Protocol (UDP) [9] header. An IP Version 4 header [10] of 20 bytes is applied at the network layer. Finally, we incorporated a further 34 bytes to represent the IEEE802.11 Medium Access (MAC) header [11]. These five protocol headers constitute the header size, $n_h=82$, which is attached to each packet that was transmitted over the link.

B. Random linear fountain code

Fig. 4 shows the optimum payload when the number of excess packets required to meet the TBRR, $\beta=0.95$, is $\epsilon=5$ from (1).

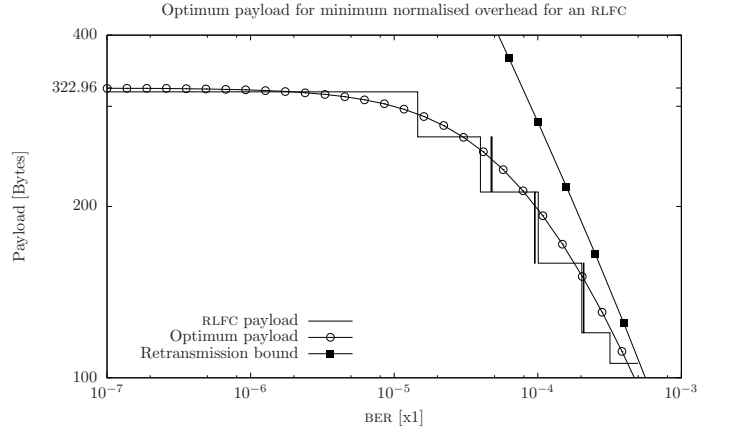


Fig. 4. Optimum packet size minimising the normalised overhead associated with an RLFC. The bounds developed in (7) and (19) of Section II are shown together with the results of an exhaustive search over \mathcal{P} .

C. Luby transform using the robust-soliton degree distribution

An example of the number of packets required for successful block recovery is shown in Figure 7 of [1], where the specific selection of the RSDD parameters, c and δ , was shown to have a significant effect on the number of packets required at the decoder for a given BRR. Below we study the impact of the ultimate packet payload rather than that of a particular Fountain code. For this reason, a gradient search was employed to identify beneficial values of c and δ that minimise the number of packets that must be received for each of the available partitions in \mathcal{P} . For comparison, Table I presents the value of s_{\min} associated with the optimum parameters for $n_p=10$, together with the configurations adopted in [1] for a TBRR of $\beta=0.95$. Table I

Original figure [1]	Parameters		Packets for 0.95 recovery rate	
	c	δ	$n_p = 10000$	$n_p = 10$
7-a	0.01	0.5	11518	—
7-c	0.03	0.5	11071	—
7-b	0.10	0.5	10847	23
—	0.40	1.0	12569	21

TABLE I

NUMBER OF RECEIVED PACKETS REQUIRED FOR $\beta=0.95$. PARAMETERS COMPARED ARE FROM FIGURE 7 OF [1] AND THE OPTIMUM FOR $n_p=10$.

demonstrates the dependence of c and δ on the number of packets that must be received in order to meet the TBRR. It also shows that the overhead incurred might be increased by the inappropriate choice of parameters for the RSDD.

Since no closed-form expression exists that relates the number of partitions, n_p , to the minimum number of excess packets that must be received to achieve the TBRR we use a numerical method. Our aim is to develop a simple model, hence we have elected to adopt a least-squares fitted polynomial, relating s_{\min} to n_p for a low-BER link. The nominal values for s_{\min} were identified by simulation. The impact of packet erasures was incorporated into the model through the inclusion of a $(\frac{1}{1-p})$ term. The accuracy of this simple packet erasure term degrades as the BER increases due to the spreading of the PDF associated with the number of correctly received packets; see Section II-D and Fig. 5.

Fig. 5 shows the relationship between s_{\min} and n_p for three different BERs evaluated from (25). The lower trace, corresponding to a BER of 10^{-7} , provides the eleven points selected for the least-squares curve fit of a cubic polynomial, yielding:

$$\chi(n_p) = \frac{9.844 \cdot 10^{-5} n_p^3 - 1.418 \cdot 10^{-2} n_p^2 + 2.163 n_p + 0.8358}{(1-b)^{8(\frac{n_B}{n_p} + n_h)}} \quad (25)$$

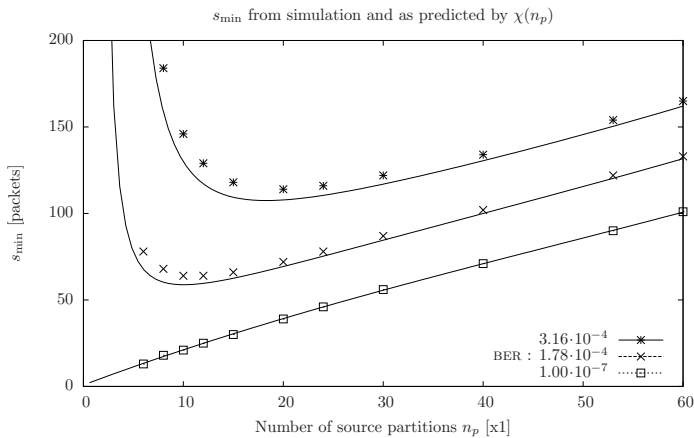


Fig. 5. The least-squares fitted cubic polynomial expression for $\chi(\cdot)$, (25), is compared with simulations for discrete values of $n_p \in \mathcal{P}$ at three BERs.

The upper two traces of Fig. 5 exhibit the inflexion that results from the effects of having an increasing BER at a low number of partitions associated with larger packets. Observe in Fig. 5 that $\chi(n_p)$ is underestimating s_{\min} at higher error rates, because the error model adopted is not capable of differentiating between single and multiple loss events. The expression of $\chi(n_d)$ in (25) is processed further in a similar manner to that adopted in the case of the RLFC and is described in Section III-B. The solution for the optimum payload size and that identified by an exhaustive search over all values of $n_p \in \mathcal{P}$ in (24) is shown in Fig. 6.

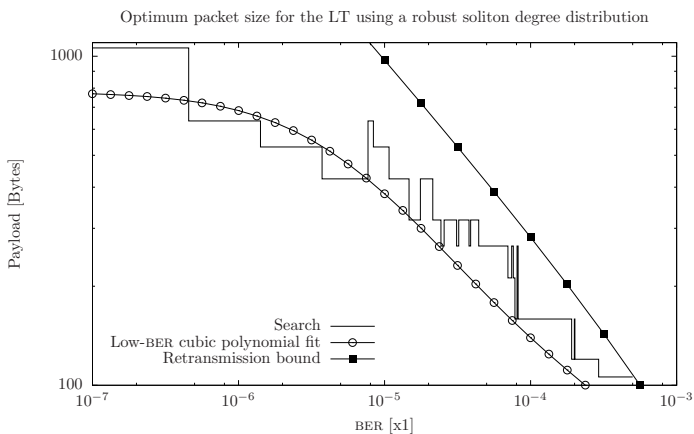


Fig. 6. Optimum payload size to minimise the normalised overhead associated with an LT code using the RSDD. The bounds developed in Section II are shown together with the results of an exhaustive search.

The optimum payload size, under low BER conditions, of 322.96 bytes identified for the RLFC is significantly smaller than for the LT code, which was 768.98 bytes. The above mentioned full search was performed to identify the optimum payload, selected using $n_p \in \mathcal{P}$, which was found to be 318 bytes for $n_p = 20$ and 1060 bytes for $n_p = 6$, respectively. Furthermore, in the lower trace of Fig. 5, corresponding to a BER of 10^{-7} , it can be seen that for the LT scheme we have $s_{\min} = 13$, when $n_p = 6$. If we were to use this configuration for the RLFC, where $\epsilon = 5$, we would expect $s_{\min} = n_p + \epsilon = 11$. This suggests that there is only a limited difference between the schemes in terms of the minimum number of packets to be transmitted. Hence the question arises, as to why there is such a significant change in the optimum payload size associated with each scheme?

To elaborate on this issue a little further, the RLFC requires an excess of ϵ packets to be conveyed beyond the minimum necessary number, in order to reconstruct the source block. Naturally, the selection of a reduced payload size will reduce the total contribution of ϵ smaller packets to the gross overhead. However, selecting progressively smaller payloads implies that the fixed-length headers associated with every packet become increasingly more significant. Contrasting the value of

$\chi_{\text{RLFC}}(n_p) = n_p + \epsilon$ with (25) reveals that the expected number of packets grows at the same rate as n_p for the RLFC whereas the LT code exhibits a more rapid increase. This erodes any potential benefit achieved by the adoption of more partitions.

Our simulation results are presented in Table II for $n_p \in \{6, 20, 40\}$ when s_{\min} packets are transmitted, as calculated in Section II-D. The corresponding mean BRR and achieved rate are presented using a CDF for each coding scheme observed over 10^6 independent trials. The LT scheme, using a RSDD, has an achieved rate which dropped from $R = 0.407$ to $R = 0.143$ as the BER increased from 10^{-7} to 10^{-4} , while the number of partitions remained at $n_p = 6$. However, increasing the number of partitions to $n_p = 40$ at the higher BERs increased the achieved rate by a factor of two, to $R = 0.289$. Similar trends can be observed in Table II, when an RLFC was used.

FEC code	n_p	BER = 10^{-7}			BER = 10^{-4}		
		s_{\min}	BRR	R	s_{\min}	BRR	R
LT	6	13	0.951	0.407	37	0.951	0.143
	20	39	0.954	0.389	55	0.952	0.275
	40	71	0.954	0.355	87	0.953	0.289
RLFC	6	11	0.968	0.490	32	0.955	0.166
	20	25	0.989	0.629	37	0.954	0.410
	40	45	0.968	0.568	58	0.978	0.445

TABLE II

THE ACHIEVED RATE VARIATION WITH THE NUMBER OF PARTITIONS AND THE BER.

In conclusion, RLFC and LT code aided schemes designed for the transmission of blocks of source data were proposed, while meeting a TBRR at the maximum achievable rate. We have shown that the adoption of a constant payload size can have a deleterious effect on the efficiency of the data transfer. It has been demonstrated that the adoption of a fountain code may require relatively small payloads, even when operating over error-free links. A cross-layer design that allows the simultaneous adaptation of both application-layer and PHY channel coding, in addition to optimising the payload size, will form the basis of our future study in diverse multi-user scenarios.

ACKNOWLEDGMENT: The financial support of the EPSRC, UK and that of the EU under the auspices of the Ptmix project is gratefully acknowledged.

REFERENCES

- [1] D. J. C. MacKay, "Fountain Codes," in *IEE Proceedings - Communications*, vol. 152, December 2005, pp. 1062–1068.
- [2] M. Luby, "LT codes," in *Foundations of Computer Science '02: Proceedings of the 43rd Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 271–282.
- [3] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006. [Online]. Available: <http://dx.doi.org/10.1109/TIT.2006.874390>
- [4] 3GPP, "Multimedia Broadcast/Multicast Service (MBMS)," 3rd Generation Partnership Project, Technical Specification 26.346, December 2007.
- [5] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Raptor forward error correction scheme for object delivery," Internet Engineering Task Force, RFC 5053, September 2007. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5053.txt>
- [6] M. Luby, L. Vicisano, J. Gemmel, L. Rizzo, M. Handley, and J. Crowcroft, "Forward error correction (FEC) building block," Internet Engineering Task Force, RFC 3452, Dec. 2002. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3452.txt>
- [7] L. Hanzo, P. Cherriman, and J. Streit, *Wireless Video Communications: Second to Third Generation and Beyond*. Wiley-IEEE Press, 2008.
- [8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," Internet Engineering Task Force, RFC 3550, July 2003. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3550.txt>
- [9] J. Postel, "User datagram protocol," Internet Engineering Task Force, RFC 768, August 1980. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc768.txt>
- [10] —, "Internet protocol," Internet Engineering Task Force, RFC 791, Sept. 1981. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc791.txt>
- [11] IEEE Computer Society, "IEEE std 802.11-2007," IEEE, 3 Park Avenue, New York, Tech. Rep., June 2007.