# New Sequences of Linear Time Erasure Codes approaching the Channel Capacity

M. Amin Shokrollahi

Bell Labs, Room 2C-353, 700 Mountain Ave, Murray Hill, NJ 07974, USA
amin@research.bell-labs.com

**Abstract.** We will introduce a new class of erasure codes built from irregular bipartite graphs that have linear time encoding and decoding algorithms and can transmit over an erasure channel at rates arbitrarily close to the channel capacity. We also show that these codes are close to optimal with respect to the trade-off between the proximity to the channel capacity and the running time of the recovery algorithm.

## 1 Introduction

A linear error-correcting code of block length $n$ and dimension $k$ over a finite field $\mathbb{F}_q$—an $[n,k]_q$-code for short—is a $k$-dimensional linear subspace of the standard vector space $\mathbb{F}_q^n$. The elements of the code are called codewords. To the code $C$ there corresponds an *encoding map* Enc which is an isomorphism of the vector spaces $\mathbb{F}_q^k$ and $C$. A sender, who wishes to transmit a vector of $k$ elements in $\mathbb{F}_q$ to a receiver uses the mapping Enc to encode that vector into a codeword. The *rate* $k/n$ of the code is a measure for the amount of real information in each codeword. The minimum distance of the code is the minimum Hamming distance between two distinct codewords. A linear code of block length $n$, dimension $k$, and minimum distance $d$ over $\mathbb{F}_q$ is called an $[n,k,d]_q$-code.

Linear codes can be used to reliably transmit information over a noisy channel. Depending on the nature of the errors imposed on the codeword during the transmission, the receiver then applies appropriate algorithms to *decode* the received word. In this paper, we assume that the receiver knows the position of each received symbol within the stream of all encoding symbols. We adopt as our model of losses the *erasure channel*, introduced by Elias [3], in which each encoding symbol is lost with a fixed constant probability $p$ in transit independent of all the other symbols. As was shown by Elias [3], the capacity of this channel equals $1-p$.

It is easy to see that a code of minimum distance $d$ is capable of recovering $d-1$ or less erasures. In the best case, it can recover from any set of $k$ coordinates of the encoding which means that $d-1=n-k$. Such codes are called MDS-codes. A standard class of MDS-codes is given by Reed-Solomon codes [10]. The connection of these codes with polynomial arithmetic allows for encoding and decoding in time $O(n\log^2 n\log\log n)$. (See, [2, Chapter 11.7] and [10, p. 369]). However, these codes do not reach the capacity of the erasure channel, since there is no infinite sequence of such codes over a fixed field.

Elias [3] showed that a random linear code can be used to transmit over the erasure channel at any rate $R < 1 - p$, and that encoding and decoding can be accomplished with $O(n^2)$ and $O(n^3)$ arithmetic operations, respectively. Hence, we have on the one hand codes that can be encoded and decoded faster than general linear codes, but do not reach the capacity of the erasure channel; and on the other hand we have random codes which reach the capacity but have encoding and decoding algorithms of higher complexity.

The paper [1] was the first to design codes that could come arbitrarily close to the channel capacity while having linear time encoding and decoding algorithms. Improving these results, the authors of [8] took a different approach and designed fast linear-time algorithms for transmitting just below channel capacity. For all $\epsilon > 0$ they were able to produce rate $R = 1 - p(1 + \epsilon)$ codes along with decoding algorithms that could recover from the random loss of a $p$ fraction of the transmitted symbols in time proportional to $n \ln(1/\epsilon)$ with high probability, where $n$ is the block length. These codes could also be encoded in time proportional to $n \ln(1/\epsilon)$. They belong to the class of low-density parity check codes of Gallager [4]. In contrast to Gallager codes, however, the graphs used to construct the asymptotically good codes obtained in [8] are highly irregular.

The purpose of the present paper is twofold. First, we prove a general trade-off theorem between the proximity of a given Gallager code to the channel capacity in terms of the loss fraction and the running time of the recovery algorithm of [8]. We show that in this respect, the codes constructed in that paper are close to optimal. Next, we exhibit a different sequence of asymptotically close to optimal codes which have better parameters than the codes in [8]. An interesting feature of these codes is that the underlying bipartite graphs are *right regular*, i.e., all nodes on the right hand side of the graph have the same degree. Since they are theoretically better than their peers, we expect them to also perform better in practice.

The organization of the paper is as follows. In the next section we will review the construction of Gallager codes. Next, we prove upper bounds on the maximum tolerable loss fraction in terms of the running time of the decoding algorithm. The last two sections are concerned with the derivation of the sequence of right regular erasure codes.

## 2 Codes from Bipartite Graphs

In this section, we will briefly review the class of codes we are interested in, and the erasure recovery algorithm associated to them.

Our codes are similar to the Gallager codes [4] in that they are built from sparse bipartite graphs. In contrast to Gallager codes, however, our codes will be constructed from graphs that have a highly irregular degree pattern on the left.

Let $G$ be a bipartite graph with $n$ nodes on the left and $n - k$ nodes on the right. $G$ gives rise to a binary code of block-length $n$ and dimension $\geq k$ in the

following way: let the adjacency matrix of the graph $G$ be given as

$$A = \left( \begin{array}{c|c} 0 & H^\top \\ \hline H & 0 \end{array} \right),$$

where $H$ is some $(n-k) \times n$ matrix describing the connections in the graph. The code defined by the graph is the code with parity check matrix $H$. A different way of describing the code is as follows: we index the coordinate positions of the code with the $n$ nodes on the left hand side of the graph. The code consists of all binary vectors $(c_1, \ldots, c_n)$ such that for each right node in the graph the sum of the coordinate places adjacent to it equals zero. The block-length of this code equals $n$, and its dimension is at least $k$ since we are imposing $n-k$ linear conditions on the coordinates of a codeword. Expressed in terms of the graph, the fraction of redundant symbols in a codeword is at most $a_L/a_R$ where $a_L$ and $a_R$ are the average node degrees on the left and the right hand side of the graph, respectively. In other words, the rate of the code is at least $1 - a_L/a_R$. This description of the rate will be useful in later analysis. In the following, we will assume that the rate is in fact equal to this value. This is because the statements we will prove below will become even stronger if the rate is larger.

The above construction needs asymptotically $O(n^2)$ arithmetic operations to find the encoding of a message of length $k$, if the graph is sparse. One can apply a trick to reduce the running time to $O(n)$ by a modification of the construction. Details can be found in [8].

Suppose now that a codeword $(c_1, \ldots, c_n)$ is sent and that certain erasures have occurred. The erasure recovery algorithm works as follows. We first initialize the contents of the right hand nodes of $G$ with zero. Then we collect the non-erased coordinate positions, add their value to the current value of their right neighbors, and delete the left node and all edges emanating from it from the graph. After this stage, the graph consists of the erased nodes on the left and the edges emanating from these nodes. In the next step we look for a right node in the graph of degree one, i.e., a node that has only one edge coming out of it. We transport the value of this node to its unique left neighbor $\ell$, thereby recovering the value of $c_\ell$. We add $c_\ell$ to the current value of all the right neighbors of $\ell$, delete the edges emanating from $\ell$, and repeat the process until we cannot find a node of degree one on the right, or until all nodes on the left have been recovered.

It is obvious that, on a RAM with unit cost measure, the amount of arithmetic operations to finish the algorithm is at most proportional to the number of edges in the graph, i.e., to $na_L$, where $a_L$ is the average node degree on the left. The aim is thus to find graphs with constant $a_L$ for which the recovery algorithm finishes successfully.

The main contribution of [8] was to give an analytic condition on the maximum fraction of tolerable losses in terms of the degree distribution of the graph. More precisely, define the *left* and the *right* degree of an *edge* in the graph as the degree of the left, resp. right node it is emanating from. Further, denote by $\lambda_i$ and $\rho_i$ the fraction of edges of left, resp. right degree $i$, and consider the generating functions $\lambda(x) := \sum_i \lambda_i x^{i-1}$ and $\rho(x) := \sum_i \rho_i x^{i-1}$. In the following we will call the pair $(\lambda, \rho)$ a *degree distribution*.

**Table 1.** Rate 1/2 codes built from heavy tail/Poisson distribution

| $N$ | $a_R$ | $\theta$ | $\delta/(1-R)$ | $\delta$ | $\hat{\delta}$ | $\delta/\hat{\delta}$ |
|---|---|---|---|---|---|---|
| 8 | 5.9266 | 5.9105 | 0.91968 | 0.45984 | 0.49085 | 0.93682 |
| 16 | 7.0788 | 7.0729 | 0.95592 | 0.47796 | 0.49609 | 0.96345 |
| 27 | 8.0054 | 8.0027 | 0.97256 | 0.48628 | 0.49799 | 0.97648 |
| 47 | 9.0256 | 9.0243 | 0.98354 | 0.49177 | 0.49902 | 0.98547 |
| 79 | 10.007 | 10.007 | 0.98990 | 0.49495 | 0.49951 | 0.99087 |
| 132 | 10.996 | 10.996 | 0.99378 | 0.49689 | 0.49975 | 0.99427 |
| 221 | 12.000 | 12.000 | 0.99626 | 0.49813 | 0.49988 | 0.99650 |

The main theorem of [8] states that if the graph is chosen at random with degree distribution $(\lambda, \rho)$ and if the erasures occur at random positions, then the above erasure recovery algorithm can correct a $\delta$-fraction of losses if $\rho(1 - \delta\lambda(x)) \geq 1 - x$ for $x \in [0, 1]$. In a later paper [6], this condition was slightly relaxed to

$$\delta\lambda\left(1 - \rho(1 - x)\right) < x \quad \text{for } x \in (0, \delta]. \tag{1}$$

The paper [8] further exhibited for any $\epsilon > 0$ and any rate $R$ an infinite sequence of degree distributions $(\lambda, \rho)$ giving rise to codes of rate at least $R$ such that the above inequality is valid for $\delta = (1-R)(1-\epsilon)$, and such that the average left degree of the graph is $O(\log(1/\epsilon))$. In other words, $\delta$ can get arbitrarily close to its optimal value $(1 - R)$ with a "logarithmic" sacrifice in the running time of the algorithm. Explicitly, for any given $0 < \epsilon < 1$, one chooses an integer $N$ close to $1/\epsilon$ and considers the pair $(\lambda_N, \rho_N)$ with

$$\lambda_N(x) = \frac{1}{H(N-1)} \sum_{k=1}^{N-1} \frac{x^k}{k}, \quad \rho_N(x) = \mathrm{e}^{\theta_N \cdot (x-1)}, \tag{2}$$

where $\theta_N$ is chosen to make the fraction of the nodes on the left and the right equal to $1 - R$, and $H(N-1)$ is the harmonic sum $\sum_{k=1}^{N-1} 1/k$. This sequence is referred to as the *heavy tail/Poisson* sequence in the following. Table 1 gives an example of the performance of this sequence for some codes of rate 1/2. In that table, $a_R$ denotes the average right degree, $\delta$ is the maximum tolerable loss fraction, and $\hat{\delta}$ is a theoretical upper bound on $\delta$, see Remark 1 following Corollary 1.

In the following we will show that this sort of relationship between the average degree and the maximum tolerable loss fraction is the best possible. Furthermore, we will exhibit a new sequence of degree distributions for which the same type of relationship holds between the average degree and the maximum tolerable loss fraction.

## 3 Upper Bounds for the Maximum Tolerable Loss Fraction

In this section we will prove some upper bounds on the maximum tolerable loss fraction $\delta$ for which the algorithm given in the previous section is successful when applied to a random graph with a given degree distribution. Our main tool will be the inequality (1). We will first need a preliminary result.

**Lemma 1.** *Let $G$ be a bipartite graph with edge degree distribution $(\lambda, \rho)$. Then, the average left, resp. right, node degree of $G$ equal*

$$\frac{1}{\int_0^1 \lambda(x)\mathrm{d}x}, \qquad \frac{1}{\int_0^1 \rho(x)\mathrm{d}x},$$

*respectively. Let the polynomials $\Lambda$ and* R *be defined by*

$$\Lambda(x) := \frac{\int_0^x \lambda(t)\mathrm{d}t}{\int_0^1 \lambda(t)\mathrm{d}t}, \qquad \mathrm{R}(x) := \frac{\int_0^x \rho(t)\mathrm{d}t}{\int_0^1 \rho(t)\mathrm{d}t}.$$

*Then the coefficient of $x^i$ in $\Lambda(x)$, resp. $R(x)$ equals the fraction of* nodes *of degree $i$ on the LHS, resp. RHS of $G$.*

*Proof.* Let $L$ denote the number of nodes on the LHS of $G$. Then the number of edges in $G$ equals $a_L L$, and the number of edges of degree $i$ equals $\lambda_i a_L L$. Hence, the number of nodes of degree $i$ on the LHS of the graph equals $\lambda_i a_L L/i$, since each such node contributes to exactly $i$ edges of degree $i$. Hence, the fraction of nodes of degree $i$ equals $\lambda_i a_L/i$. Since the sum of all these fractions equals 1, we obtain $a_L^{-1} = \sum_i \lambda_i/i = \int_0^1 \lambda(x)\mathrm{d}x$ and this yields both assertions for the LHS of the graph. The assertions on the RHS follow similarly. □

The following theorem shows a rather strong upper bound on $\delta$.

**Theorem 1.** *Let $\lambda$ and $\rho$ denote the right and left edge degree distributions of a bipartite graph. Let $\delta$ be a positive real number such that*

$$\delta\lambda\left(1-\rho(1-x)\right) \leq x$$

*for $0 < x \leq \delta$. Then we have*

$$\delta \leq \frac{a_L}{a_R}\left(1-(1-\delta)^{a_R}\right),$$

*where $a_L$ and $a_R$ are the average node degrees of the graph on the left, resp. right, hand side.*

*Proof.* As a real valued function the polynomial $\lambda(x)$ is strictly increasing for positive values of $x$. Hence it has a unique inverse $\lambda^{-1}$ which is also strictly increasing. The first of the above inequalities is thus equivalent to

$$1-\rho(1-x) \leq \lambda^{-1}\left(x/\delta\right)$$

for $0 < x \leq \delta$. Integrating both sides from 0 to $\delta$ and simplifying the expressions, we obtain

$$\int_0^1 \rho(x)\mathrm{d}x - \int_0^{1-\delta} \rho(x)\mathrm{d}x \geq \delta \int_0^1 \lambda(x)\mathrm{d}x.$$

(Note that $\int_0^1 \lambda^{-1}(x)\mathrm{d}x = 1 - \int_0^1 \lambda(x)\mathrm{d}x$.) Invoking the previous lemma, we thus have

$$\delta \leq \frac{a_L}{a_R} - a_L \int_0^{1-\delta} \rho(x)\mathrm{d}x = \frac{a_L}{a_R}\left(1 - \frac{\int_0^{1-\delta}\rho(x)\mathrm{d}x}{\int_0^1 \rho(x)\mathrm{d}x}\right) = \frac{a_L}{a_R}\left(1 - \mathrm{R}(1-\delta)\right),$$

where the polynomial R is defined as in the statement of Lemma 1. To finish the proof we only need to show that $\mathrm{R}(1-\delta) \geq (1-\delta)^{a_R}$. To see this, first note that if $a_1, \ldots, a_M$ are nonnegative real numbers adding up to 1 and $\mu$ is any nonnegative real number, then

$$\sum_i a_i \mu^i \geq \mu^{\sum_i i a_i}. \tag{3}$$

This follows from taking logarithms of both sides and noting that the log-function is concave. Denoting by $a_i$ the coefficients of $\mathrm{R}(x)$ and setting $\mu := 1 - \delta$, this gives our desired inequality: the $a_i$ are by the previous lemma the fractions of nodes of degree $i$, and hence $\sum_i i a_i$ is the average right degree $a_R$. □

**Corollary 1.** *With the same assumptions as in the previous theorem, we have*

$$\delta \leq \frac{a_L}{a_R}\left(1 - (1 - a_L/a_R)^{a_R}\right).$$

*Proof.* Follows from $\delta \leq a_L/a_R$. □

*Remark 1.* A more refined upper bound for $\delta$ is given by the following. Let $r$ denote the quantity $a_L/a_R$, i.e., one minus the rate of the code. Suppose that $a_R > 1/r$—an assumption that is automatically satisfied if the average left degree is larger than one—and consider the function $f(x) = x - r(1-(1-x)^{a_R})$. We have $f(0) = 0$, $f(1) = 1 - r > 0$, $f'(0) < 0$, and $f'(x)$ has exactly one root. Hence, $f$ has exactly one nonzero root, denoted by $\hat{\delta}_{a_R, r}$ and this root is between 0 and 1. The previous theorem asserts that the maximum $\delta$ satisfying the inequality (1) is smaller than $\hat{\delta}_{a_R, a_L/a_R}$.

The following definition is inspired by Corollary 1.

**Definition 1.** *A sequence $(\lambda_m, \rho_m)$ of degree distributions giving rise to codes of rate $R \geq R_0$ for a fixed $R_0 > 0$ is called* asymptotically quasi-optimal *if there exists a constant $\mu$ (possibly depending on $R$) and for each $m$ there exists a positive $\delta_m$ with $\delta_m \lambda_m(1 - \rho_m(1-x)) \leq x$ for $x \in (0, \delta_m]$ such that the average right degree $a_R$ satisfies $a_R \leq \mu \log(1-\delta/(1-R))$.*

The significance of quasi-optimal is the following: we want to construct codes that can recover as many erasures as possible in as little as possible time. The running time of the erasure recovery algorithm is proportional to the block-length of the code times $a_R$, the average right degree. Hence, we are looking for a trade-off between $a_R$ and $\delta$. Corollary 1 shows that we cannot make $a_R$ too small. In fact, it implies that $a_R \geq \log(1-\delta/(1-R))/\log R$. However, we might hope that we can make $a_R$ smaller than $\log(1-\delta/(1-R))$ times a (negative) constant $\mu$. In this way, we have maintained a qualitative optimality of the code sequence in the sense that the running time increases only logarithmically with the relative increase of the erasure correction capability.

The heavy tail/Poisson sequence (2) is asymptotically quasi-optimal as is shown in [8]. Starting in the next section, we will introduce another quasi-optimal sequence which has certain advantages over the heavy tail/Poisson sequence.

We close this section by stating another useful upper bound on the maximum possible value of $\delta$.

**Lemma 2.** *Suppose that $\lambda$ and $\rho$ are polynomials and $\delta$ is such that $\delta\lambda(1-\rho(1-x)) < x$ for $x \in (0,\delta]$. Then $\delta \leq \rho'(1)/\lambda'(0)$.*

*Proof.* Let $f(x) = \delta\lambda(1-\rho(1-x)) - x$. By assumption, $f(x) < 0$ for $x \in (0,\delta]$. This implies that $f'(0) \leq 0$, where $f'(x)$ is the derivative of $f$ with respect to $x$. But $f'(0) = \delta\lambda'(0)\rho'(1) - 1$, which gives the assertion. □

## 4 Fractional Binomial Coefficients

As can be seen from their description (2), the heavy tail/Poisson sequence is closely related to the Taylor series of $-\ln(1-x)$. The new sequence that we will describe in the next section will be related to the Taylor expansion of $(1-x)^\alpha$ where $1/\alpha$ is an integer. The coefficients of this expansion are fractional binomial coefficients. For this reason, we will recall some well-known facts about these numbers.

For real $\alpha$ and a positive integer $N$ we define

$$\begin{aligned}\binom{\alpha}{N} &:= \frac{\alpha(\alpha-1)\cdots(\alpha-N+1)}{N!} \\ &= (-1)^{N-1}\frac{\alpha}{N}\left(1-\frac{\alpha}{N-1}\right)\cdots\left(1-\frac{\alpha}{2}\right)(1-\alpha)\,.\end{aligned}$$

For convenience, we also define $\binom{\alpha}{0} := 1$. We have the Taylor expansion

$$1-(1-x)^\alpha = \sum_{k=1}^{\infty}\binom{\alpha}{k}(-1)^{k+1}x^k. \tag{4}$$

Note that for $0 < \alpha < 1$ the coefficients of the right hand power series above are all positive. Furthermore, we have

$$\sum_{k=1}^{N-1}\binom{\alpha}{k}(-1)^{k+1} = 1 - \frac{N}{\alpha}\binom{\alpha}{N}(-1)^{N+1}, \tag{5}$$

and

$$\sum_{k=1}^{N-1} \binom{\alpha}{k} \frac{(-1)^{k+1}}{k+1} = \frac{\alpha - \binom{\alpha}{N}(-1)^{N+1}}{\alpha+1}, \tag{6}$$

as can easily be proved by induction.

In the following, we will derive some estimates on the size of the binomial coefficients.

**Proposition 1.** *Let $\alpha$ be a positive real number less than $1/2$ and let $N \geq 2$ be an integer. There is a constant $c$ independent of $\alpha$ and $N$ such that*

$$\frac{c\alpha}{N^{\alpha+1}} \leq \binom{\alpha}{N}(-1)^{N+1} \leq \frac{\alpha}{N^{\alpha+1}}.$$

*Proof.* First note that $\binom{\alpha}{N}(-1)^{N+1}$ is positive. Taking its logarithm and expanding the series $-\ln(1-x) = \sum_i x^i/i$, we obtain

$$\ln\left(\binom{\alpha}{N}(-1)^{N+1}\right) = \ln(\alpha/N) - \alpha\sum_{k=1}^{N}\frac{1}{k} - \frac{\alpha^2}{2}\sum_{k=1}^{N}\frac{1}{k^2} - \cdots. \tag{7}$$

(Note that the series involved are absolutely convergent, so that we can rearrange the orders of the summations.) For an upper bound on this sum we replace $\sum_{k=1}^{N} 1/k^s$ by 1 for $s \geq 2$ and use the left hand side of the inequality

$$\ln(N) + \gamma < \sum_{k=1}^{N}\frac{1}{k} < \ln(N) + \gamma + \frac{1}{2N},$$

where $\gamma$ is Euler's constant [5, pp. 480–481]. For the lower bound we use the right hand side of the above inequality and replace $\sum_{k=1}^{N} 1/k^s$ for $s \geq 2$ by 2. This gives, after exponentiation,

$$\frac{\alpha}{N^{\alpha+1}}\mathrm{e}^{\alpha(2-\gamma-1/2N)}(1-\alpha)^2 \leq \binom{\alpha}{N}(-1)^{N+1} \leq \frac{\alpha}{N^{\alpha+1}}\mathrm{e}^{\alpha(1-\gamma)}(1-\alpha).$$

Noting that $0 < \alpha \leq 1/2$, we obtain our assertion. □

## 5 Right Regular Sequences

A closer look at the proof of Theorem 1 reveals the following: one of the reasons we might obtain a lower than optimal upper bound for the largest $\delta$ satisfying (1) is that the inequality (3) is not sharp. In fact, that inequality is sharp iff all the $a_i$ except for one are zero, i.e., iff $\rho(x)$ has only one nonzero coefficient, i.e., iff the graph has only one degree on the right hand side. We call such graphs *right regular* in the following. We will study in this section right regular degree distributions and design the left hand side in such a way as to obtain asymptotically quasi-optimal sequences. We remark that right regular sequences

were previously studied in an unpublished manuscript [9]. Our approach differs however from the one given in that paper.

From a theoretical point of view, codes obtained from these sequences should perform better than the heavy tail/Poisson distribution, since they allow for a larger loss-fraction for a given rate and a given average left degree. Furthermore, the analysis given in [6] suggests that the actual performance of the code is related to how accurately the neighborhood of a message node is described by a tree given by the degree distributions. For instance, the performance of regular graphs is much more sharply concentrated around the value predicted by the theoretical analysis, than the performance of irregular graphs. Hence, if the graph is right regular, one should expect a smaller variance in the actual performance than for corresponding irregular codes.

For integers $a \geq 2$ and $N \geq 2$ we define

$$\rho_a(x) := x^{a-1}, \qquad \lambda_{\alpha,N}(x) := \alpha \frac{\sum_{k=1}^{N-1} \binom{\alpha}{k} (-1)^{k+1} x^k}{\alpha - N\binom{\alpha}{N}(-1)^{N+1}}, \tag{8}$$

where here and in the following we set $\alpha := 1/(a-1)$. Note that $\rho_a(1) = 1$, that $\lambda_{a,N}(1) = 1$ by (5), and that $\lambda_{a,N}$ has positive coefficients. Hence, $(\rho_a, \lambda_{a,N})$ indeed defines a degree distribution.

Let $\nu < 1$ be a positive constant. The sequence we are interested in is given by the following degree distributions:

$$\rho_a(x), \quad \lambda_{a,\lfloor \nu^{-1/\alpha} \rfloor}(x). \tag{9}$$

First we consider the rate of the codes defined by these distributions.

**Proposition 2.** *The rate $R = R_{a,\mu}$ of the code defined by the distributions in (9) satisfies*

$$\frac{1-\nu}{1-c\nu\nu^{1/\alpha}} \leq 1 - R \leq \frac{1-c\nu}{1-\nu\nu^{1/\alpha}},$$

*where $c$ is the constant from Proposition 1.*

*Proof.* In the following, we denote by $N$ the quantity $\lfloor \nu^{-1/\alpha} \rfloor$ and by $\lambda(x)$ the function $\lambda_{a,N}(x)$. We need to compute the integral of $\lambda(x)$. We have

$$\int_0^1 \lambda(x)\mathrm{d}x = \frac{\alpha}{\alpha+1} \frac{\alpha - \binom{\alpha}{N}(-1)^{N+1}}{\alpha - N\binom{\alpha}{N}(-1)^{N+1}}.$$

Further, $\int_0^1 \rho_a(x)\mathrm{d}x = 1/a = \alpha/(\alpha+1)$. Hence, the rate of the code is

$$1 - \frac{\alpha - N\binom{\alpha}{N}(-1)^{N+1}}{\alpha - \binom{\alpha}{N}(-1)^{N+1}} := 1 - r_{\alpha,N}. \tag{10}$$

Next, we estimate $r_{\alpha,N}$ using Proposition 1 again:

$$\frac{1-\nu}{1-c\nu\nu^{1/\alpha}} \leq r_{\alpha,N} \leq \frac{1-c\nu}{1-\nu\nu^{1/\alpha}}.$$

This gives the desired assertion on the rate. □

**Theorem 2.** *The sequence of degree distributions given in (9) is asymptotically quasi-optimal.*

*Proof.* Let us compute the maximum value of $\delta$ such that $\delta\lambda(1-\rho(1-x)) < x$ for $x \in (0,\delta)$, where $(\lambda,\rho)$ is the pair given in (9). We have

$$\begin{aligned}\delta\lambda(1-\rho(1-x)) &< \frac{\delta\alpha}{\alpha - N\binom{\alpha}{N}(-1)^{N+1}}\left(1-\rho(1-x)^{\alpha}\right)\\ &= \frac{\delta\alpha}{\alpha - N\binom{\alpha}{N}(-1)^{N+1}}x.\end{aligned}$$

So, for $\delta\lambda(1-\rho(1-x)) < x$ we need

$$\delta \leq \frac{\alpha - N\binom{\alpha}{N}(-1)^{N+1}}{\alpha}. \tag{11}$$

This is, by the way, the same upper bound as the one obtained from Lemma 2. In the following we assume that $\delta$ is equal to the above upper bound, and compute $\delta/(1-R)$, $R$ being the rate of the code as computed in (10):

$$\frac{\delta}{1-R} = \frac{\alpha - \binom{\alpha}{N}(-1)^{N+1}}{\alpha} \geq 1 - \frac{1}{N^{\alpha+1}}.$$

(The inequality is obtained using the upper bound for $\binom{\alpha}{N}(-1)^{N+1}$ given in Proposition 1.) Ignoring diophantine constraints, we assume in the following that $N = \nu^{-1/\alpha}$. This gives

$$\frac{\delta}{1-R} \geq 1 - \nu^{(\alpha+1)/\alpha} = 1 - \nu^{a_R},$$

where $a_R = a = (\alpha+1)/\alpha$ is the average right degree of the code. This proves the assertion. □

In practical situations, one is interested in designing a code of a given fixed rate. Since the parameter $\alpha$ in the definition of the sequence in (9) is the inverse of an integer, its range is discrete. However, we can come arbitrarily close to our desired rate by making $\alpha$ smaller, i.e., by allowing high degrees on the RHS of the graph. Some examples for different target rates are given in Table 2. The value of $N$ has been hand-optimized in these examples to come as close to the desired rate as possible. The last column in that table corresponds to the value $\hat{\delta}$ defined in Remark 1 following Corollary 1. It gives a theoretical upper bound on the best value of $\delta$. As can be observed from the table, the maximum value of $\delta$ converges very quickly to the maximum possible value. Also, a comparison between these codes and the heavy tail/Poisson distribution in Table 1 reveals that the new codes are better in terms of the trade-off between $\delta$ and $a_R$. However, the new codes need larger degrees on the left than the heavy tail/Poisson distribution.

To obtain codes that have a *fixed* rate, one can modify $\alpha$ slightly. Such examples are given in Table 3. Both parameters ($\alpha$ and $N$) of these codes are hand-optimized to give the best results.

**Table 2.** Right regular sequences for rates $R$ close to 2/3, 1/2, and 1/3

| $N$ | $a_R$ | $1-R$ | $\delta/(1-R)$ | $\delta$ | $\hat{\delta}$ | $\delta/\hat{\delta}$ |
|---|---|---|---|---|---|---|
| 2 | 6 | 0.33333 | 0.60000 | 0.20000 | 0.29099 | 0.68731 |
| 3 | 7 | 0.31677 | 0.74537 | 0.23611 | 0.28714 | 0.82230 |
| 6 | 8 | 0.32886 | 0.88166 | 0.28994 | 0.31243 | 0.92801 |
| 11 | 9 | 0.33645 | 0.93777 | 0.31551 | 0.32690 | 0.96514 |
| 17 | 10 | 0.33357 | 0.96001 | 0.32024 | 0.32724 | 0.97860 |
| 27 | 11 | 0.33392 | 0.97502 | 0.32558 | 0.32984 | 0.98711 |
| 42 | 12 | 0.33381 | 0.98401 | 0.32847 | 0.33113 | 0.99197 |
| 64 | 13 | 0.33312 | 0.98953 | 0.32963 | 0.33134 | 0.99484 |
| 13 | 6 | 0.50090 | 0.96007 | 0.48090 | 0.49232 | 0.97679 |
| 29 | 7 | 0.50164 | 0.98251 | 0.49287 | 0.49759 | 0.99052 |
| 60 | 8 | 0.49965 | 0.99159 | 0.49545 | 0.49762 | 0.99563 |
| 12. | 9 | 0.49985 | 0.99598 | 0.49784 | 0.49885 | 0.99797 |
| 257 | 10 | 0.50000 | 0.99805 | 0.49903 | 0.49951 | 0.99904 |
| 523 | 11 | 0.50002 | 0.99904 | 0.49954 | 0.49977 | 0.99953 |
| 1058 | 12 | 0.49999 | 0.99953 | 0.49975 | 0.49986 | 0.99977 |
| 111 | 6 | 0.66677 | 0.99698 | 0.66475 | 0.66584 | 0.99837 |
| 349 | 7 | 0.66667 | 0.99904 | 0.66603 | 0.66636 | 0.99950 |
| 1077 | 8 | 0.66663 | 0.99969 | 0.66642 | 0.66653 | 0.99984 |
| 3298 | 9 | 0.66669 | 0.99990 | 0.66662 | 0.66665 | 0.99995 |

## 6 Conclusions and Open Questions

In this paper we have analyzed the theoretical performance of a simple erasure recovery algorithm applied to Gallager codes by deriving upper bounds on the maximum fraction of tolerable losses given a parameter that describes the running time for encoding and decoding of the codes. We have shown that there is a trade-off between proximity to the optimal value of tolerable losses, i.e., one minus the rate of the code, and the average degree of nodes in the graph, in the sense that multiplying the average degree by a constant factor implies an exponential relative increase of the maximum tolerable loss fraction. Further, we have introduced a new sequence of graphs which are asymptotically close to optimal with respect to this criterion. These graphs are right regular and their node degree distribution on the left hand side is closely related to the power series expansion of $(1-x)^{\alpha}$, where $\alpha$ is the inverse of an integer. Previously, the only known such sequence was the heavy tail/Poisson sequence introduced in [8]. We have included examples which show that the new codes tolerate a higher fraction of losses if the average degree of the graph is fixed.

It would be very interesting to extend the analysis given in this paper to other decoding algorithms, e.g., to the simple decoding algorithm of Gallager for error correcting codes [4, 7]. Such an analysis would probably give clues for constructing infinite sequences of graphs that perform asymptotically optimally with respect to the decoding algorithm in question.

**Table 3.** Rate 1/2 codes, $\rho(x) = x^{a_R - 1}$, $\lambda_{\alpha,N}(x)$ as defined in (8) for arbitrary $\alpha$.

| $N$ | $\alpha$ | $a_R$ | $\delta/(1-R)$ | $\delta$ | $\hat{\delta}$ | $\delta/\hat{\delta}$ |
|---|---|---|---|---|---|---|
| 11 | 0.17662 | 6 | 0.95982 | 0.47991 | 0.49134 | 0.97673 |
| 27 | 0.16412 | 7 | 0.98190 | 0.49095 | 0.49586 | 0.99009 |
| 59 | 0.14225 | 8 | 0.99142 | 0.49571 | 0.49798 | 0.99543 |
| 124 | 0.12480 | 9 | 0.99596 | 0.49798 | 0.49901 | 0.99794 |
| 256 | 0.11112 | 10 | 0.99800 | 0.49900 | 0.49951 | 0.99898 |
| 521 | 0.09993 | 11 | 0.99896 | 0.49948 | 0.49975 | 0.99945 |
| 1057 | 0.09090 | 12 | 0.99950 | 0.49975 | 0.49988 | 0.99974 |

## Acknowledgements

I would like to thank Saejoon Kim and an anonymous referee for pointing out several typos in the original version of the paper.

## References

[1] N. Alon and M. Luby. A linear time erasure-resilient code with nearly optimal recovery. *IEEE Trans. Inform. Theory*, 42:1732–1736, 1996.

[2] R.E. Blahut. *Theory and Practice of Error Control Codes.* Addison Wesley, Reading, MA, 1983.

[3] P. Elias. Coding for two noisy channels. In *Information Theory, Third London Symposium*, pages 61–76, 1955.

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

[5] R.L. Graham, D.E. Knuth, and P. Patashnik. *Concrete Mathematics.* Addison-Wesley, 1994.

[6] M. Luby, M. Mitzenmacher, and M.A. Shokrollahi. Analysis of random processes via and-or tree evaluation. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 364–373, 1998.

[7] M. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D. Spielman. Analysis of low density codes and improved designs using irregular graphs. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 249–258, 1998.

[8] M. Luby, M. Mitzenmacher, M.A. Shokrollahi, D. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proceedings of the 29th annual ACM Symposium on Theory of Computing*, pages 150–159, 1997.

[9] M. Luby and M.A. Shokrollahi. Right regular erasure codes. Unpublished, 1998.

[10] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, 1988.