Consider the following generalization of LT codes: instead of each output symbol consisting of one check symbol for a given set of input symbols, we can produce a constant number r of checks for a set of input symbols and send them as one "supersymbol" over the channel. The modified LT encoding and decoding are as follows.

Encoding Start with k input symbols that you want to encode. To produce an output supersymbol, sample an integer d from a degree distribution Ω that has the property that $\Pr[d < r] = 0$. Pick d input symbols uniformly at random. We view the input symbols as elements of a finite field $\mathbb{F}_q$ of size larger than d_{max} (the largest index of a nonzero coefficient of Ω), so that there always exists a $[d + r, d, r + 1]_q$ systematic RS code C_d. Use C_d to produce r checks from the given d input symbols and concatenate them into one supersymbol that will be sent on an erasure channel.

Decoding Start decoding when you have gathered $n = (1 + \epsilon)k$ output supersymbols. Look for an output supersymbol of neighborhood size (Nsize) r or less. Recover the corresponding input symbols. Remove these input symbols and all their outgoing edges from the decoding graph. Repeat until decoding has succeeded or until there are no more supersymbols of Nsize r or less.

Note that the decoding condition "look for an output supersymbol of degree r or less" (where the degree of an output supersymbol is defined as the sum of the degrees of its r checks) is not good enough. Suppose for instance that an output supersymbol has degree higher than r, but all of its edges are connected to one input symbols. Then we could have recovered this input symbol and continued decoding, but using the degree criterion will not allow us to decode.

Why can we recover the d corresponding input symbols from a supersymbol of Nsize r or less? We are using an MDS code, so that any d of the $d+r$ symbols can be used as information symbols to recover the others. We already have the r checks, so that we only need to know $d - r$ of the corresponding input symbols to recover the others. In other words, when only r unknown input symbols connected to the supersymbol remain, we can decode.

During decoding, define the *ripple* to be the set of output supersymbols of Nsize strictly greater than 0 and at most r, and the *cloud* to be the set of output supersymbols of Nsize strictly greater than r. At each decoding step, we recover one or more input symbols. Define the decoder to be in *state* (c, r, u) if the cloud has size c, the ripple has size r, and there are still u indecoded input symbols. Define the *state generating function* of the decoder when u symbols are undecoded to be

$$P_u(x, y) = \sum_{c \geq 0, r \geq 1} p_{c,r,u} x^c y^{r-1}. \tag{1}$$

Can we write a recursion for $P_u(x, y)$ similar to that of KLS? Note that such a recursion would not relate P_{u-1} to P_u only, as the number of symbols that we recover at once is variable (but always $\leq r$, and this number is *independent of* d). Is it a good idea to "pretend" we're actually recovering the symbols one by one?

Ripple Probabilities

I initially thought that we need to keep track of states of the form $(u, i_1, \ldots, i_r, c)$, where u indicates the number of undecoded symbols, c denotes the size of the cloud, and i_l denotes the number of ripple elements whose neighborhood is of size l. But it is probably enough to average things using the degree distribution. Suppose we are in some state (c, r, u) of the decoding and we pick a supersymbol z for decoding. What is the probability that a random $z' \neq z$ leaves the ripple? Let E_R denote the event that a randomly picked supersymbol is in the ripple before the state transition and E_0 denote the event that a randomly picked supersymbol is of Nsize 0 after the transition. Then the probability we are looking for is

$$p_R(u) = \Pr[E_0 | E_R] = \frac{\Pr[E_0 \wedge E_R]}{\Pr[E_R]}.$$

Now

$$\Pr[E_R] = \sum_{i=1}^{r} \Pr[\text{a random supersymbol is of Nsize } i \text{ when } u \text{ symbols are undecoded}].$$

The probability that a random supersymbol z' of initial Nsize d is of Nsize i when u symbols are undecoded is the probability that exactly i of its neighbors belong to the u undecoded symbols. This is given by

$$\binom{d}{i} \frac{\begin{bmatrix} u \\ i \end{bmatrix} \begin{bmatrix} k-u \\ d-i \end{bmatrix}}{\begin{bmatrix} k \\ d \end{bmatrix}},$$

so that the probability that a random supersymbol has Nsize i when u symbols are undecoded is

$$\sum_d \Omega_d \binom{d}{i} \frac{\begin{bmatrix} u \\ i \end{bmatrix} \begin{bmatrix} k-u \\ d-i \end{bmatrix}}{\begin{bmatrix} k \\ d \end{bmatrix}},$$

and

$$\Pr[E_R] = \sum_{i=1}^{r} \frac{\sum_d \Omega_d \binom{d}{i} \begin{bmatrix} u \\ i \end{bmatrix} \begin{bmatrix} k-u \\ d-i \end{bmatrix}}{\begin{bmatrix} k \\ d \end{bmatrix}}. \tag{2}$$

Now on to the computation of $\Pr[E_0 \wedge E_R]$. Again, summing over a disjoint union of events, this probability is given by

$$\Pr[E_0 \wedge E_R] = \sum_{i=1}^{r} \Pr[\text{a random supersymbol is of Nsize } i \text{ when } u \text{ symbols are undecoded and of Nsize 0 after the transition}].$$

Suppose the supersymbol z chosen for decoding at this step is of Nsize j, for some $1 \le j \le r$. We start by computing the probability of the event $E_0 \wedge E_R$ conditioned on the event that the Nsize of z is j. The probability that a supersymbol z' of initial Nsize d is of Nsize i when u symbols are undecoded and of Nsize 0 after the transition is the probability that the i yet-undecoded neighbors of z' fall within the j yet-undecoded neighbors of z, and the remaining $d-i$ neighbors of z' fall within the $k-u$ decoded symbols. This is

$$\binom{d}{i} \frac{\begin{bmatrix} j \\ i \end{bmatrix} \begin{bmatrix} k-u \\ d-i \end{bmatrix}}{\begin{bmatrix} k \\ d \end{bmatrix}}.$$

Hence the probability that a random supersymbol z' is of Nsize i when u symbols are undecoded and of Nsize 0 after the transition is given by

$$\sum_d \Omega_d \binom{d}{i} \frac{\begin{bmatrix} j \\ i \end{bmatrix} \begin{bmatrix} k-u \\ d-i \end{bmatrix}}{\begin{bmatrix} k \\ d \end{bmatrix}},$$

so that

$$\Pr[E_0 \wedge E_R | j] = \sum_{i=1}^{r} \sum_d \Omega_d \binom{d}{i} \frac{\begin{bmatrix} j \\ i \end{bmatrix} \begin{bmatrix} k-u \\ d-i \end{bmatrix}}{\begin{bmatrix} k \\ d \end{bmatrix}}.$$

Note that the probability that the symbol z picked for decoding has Nsize j for some $j \leq r$ is given by

$$\sum_{d'} \Omega_{d'} \binom{d'}{j} \frac{\begin{bmatrix} u \\ j \end{bmatrix} \begin{bmatrix} k-u \\ d'-j \end{bmatrix}}{\begin{bmatrix} k \\ d' \end{bmatrix}}.$$

Hence the probability that we're looking for is

$$\Pr[E_0 \wedge E_R] = \sum_{j=1}^{r} \sum_{d'} \Omega_{d'} \binom{d'}{j} \frac{\begin{bmatrix} u \\ j \end{bmatrix} \begin{bmatrix} k-u \\ d'-j \end{bmatrix}}{\begin{bmatrix} k \\ d' \end{bmatrix}} \cdot \Pr[E_0 \wedge E_R | j]$$

$$\sum_{i,j,=1}^{r} \sum_{d,d'} \Omega_d \Omega_{d'} \binom{d}{i} \binom{d'}{j} \frac{\begin{bmatrix} u \\ j \end{bmatrix} \begin{bmatrix} k-u \\ d'-j \end{bmatrix}}{\begin{bmatrix} k \\ d' \end{bmatrix}} \frac{\begin{bmatrix} j \\ i \end{bmatrix} \begin{bmatrix} k-u \\ d-i \end{bmatrix}}{\begin{bmatrix} k \\ d \end{bmatrix}},$$

which gives us the desired expression for $p_R(u)$.

PROBLEM: we should be able to derive the value $p_R(u) = 1/u$ that we obtain in the original LT setting. This amounts to setting $i, j = 1$, but MAYBE also

to conditioning on the fact that the symbol picked for decoding is of Nsize 1. If we just let $i, j = 1$, we get

$$p_R(u) = \sum_{d'} \Omega_{d'} d' \frac{\begin{bmatrix} k-u \\ d'-1 \end{bmatrix}}{\begin{bmatrix} k \\ d' \end{bmatrix}},$$

where the summation over d' is due to the fact that we're averaging over the possible Nsizes of the symbol picked for decoding. Is it reasonable to assume that we can also condition on $\text{Nsize}(z) = 1$? In that case we get exactly $p_R(u) = 1/u$.

Case $r = 2$

Ripple Suppose we are in state (c, r, u) and we pick a random supersymbol z for decoding. What is the probability that a ripple element becomes useless after the decoding step? It is given by

$$p_R(u) = \Pr[E_0 | E_R] = \frac{\Pr[E_0 \wedge E_R]}{\Pr[E_R]}$$

where E_0 is the event that a random supersymbol is of degree 0 after the decoding step and E_R is the probability that a random supersymbol is in the ripple before decoding.

$\Pr[E_R]$ is the probability that a random supersymbol is of reduced degree 1 or 2 at this decoding step. But a random supersymbol has reduced degree i when u symbols are undecoded with probability

$$\sum_d \Omega_d \binom{d}{i} \frac{\begin{bmatrix} u \\ i \end{bmatrix} \begin{bmatrix} k-u \\ d-i \end{bmatrix}}{\begin{bmatrix} k \\ d \end{bmatrix}},$$

so

$$\Pr[E_R] = \sum_{i=1,2} \sum_d \Omega_d \frac{\left[\begin{array}{c} u \\ i \end{array}\right] \left[\begin{array}{c} k-u \\ d-i \end{array}\right]}{\left[\begin{array}{c} k \\ d \end{array}\right]}$$

$$\sum_d \Omega_d du \frac{\left[\begin{array}{c} k-u \\ d-1 \end{array}\right]}{\left[\begin{array}{c} k \\ d \end{array}\right]} + \sum_d \Omega_d \frac{d(d-1)}{2} \frac{u(u-1)\left[\begin{array}{c} k-u \\ d-2 \end{array}\right]}{\left[\begin{array}{c} k \\ d \end{array}\right]}.$$

Similarly, $\Pr[E_0 \wedge E_R]$ is the probability that a random supersymbol is of reduced degree 1 before the transition and 0 after, or of reduced degree 2 before the transition and 0 after. But a random supersymbol is of reduced degree i before the transition and 0 after with probability