

Prefixing Method for Correcting Repetition Errors

Lara Dolecek
 LIDS, EECS Department
 Massachusetts Institute of Technology
 Cambridge, MA 02139, USA
 Email: dolecek@mit.edu

Venkat Anantharam
 EECS Department
 University of California
 Berkeley, CA 94720, USA
 Email: ananth@eecs.berkeley.edu

Abstract—We develop a prefixing method for correcting any prescribed number r of repetition errors in an arbitrary binary block code. The proposed method constructs a prefix for each codeword such that the resulting strings are all of the same length and despite any r repetitions in the concatenation of the prefix and the codeword, the original codeword can be recovered. Further, the prefix length scales logarithmically with the blocklength of the original code, so the added redundancy is asymptotically negligible.

I. INTRODUCTION

In traditional communication systems the input message \mathbf{x} is encoded using a substitution error correcting code into a coded message $\mathbf{c} = C(\mathbf{x})$. The coded message \mathbf{c} is modulated and then transmitted over a channel, which typically introduces additive noise. The resulting waveform $s(t)$ seen at the receiver is then sampled at certain locations determined by the timing recovery process. This sampled sequence is the input to the decoder which then produces the estimate of \mathbf{c} (or \mathbf{x}). In analyzing the performance of codes, it is traditionally assumed that the timing recovery process is sufficiently accurate that there is one well positioned sample per symbol interval. Ensuring this is increasingly problematic: as data rates increase and the power constraints on chip designs become more stringent, timing recovery becomes increasingly expensive in terms of power consumption and allocated chip area.

To circumvent this cost, particularly in high speed systems, it could be worthwhile to implement a poorer timing recovery scheme, while oversampling the received waveform to attempt to ensure that no information is lost. As a result, the waveform $s(t)$ instead of being sampled at the proper instances $kT_s + \tau_k$ might be sampled at instances roughly T apart, for $T < T_s$, where T_s denotes the symbol interval. In the idealized infinite SNR limit of a PAM system, this situation can be viewed as if some symbols are sampled more than once. As a result, instead of creating n samples from $s(t)$, where n is the codeword length, $n + r$ samples are produced, where $r \geq 0$ denotes the total number of repetitions¹. Motivated by this scenario, in this paper we present a general method for improving the immunity to r repetition errors of an arbitrary binary block code, while incurring asymptotically negligible additional redundancy.

In Section II we briefly present a variant of our construction, first presented in [3], of subsets of binary strings immune

to repetitions. Section III contains several number-theoretic results. Based on these results, in Section IV we show how to transform a given binary block code so each codeword results in a string of fixed length of the type described in Section II. This is done by attaching a prefix to each codeword in the original code. As a result, despite any r repetitions in the concatenation of the prefix and the codeword, the original codeword can be recovered. The length of the prefix scales logarithmically with the codeword length, so the additional redundancy is asymptotically negligible. Section V briefly discusses suitable decoding algorithms for this scheme.

II. REPETITION ERROR CORRECTING SETS OF BINARY STRINGS

Let T_n be the $n \times n - 1$ binary matrix, satisfying

$$T_n(i, j) = \begin{cases} 1, & \text{if } i = j, j + 1 \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Given a binary string \mathbf{c} of length n let $\tilde{\mathbf{c}} = \mathbf{c}T_n$. A repetition in \mathbf{c} in position t corresponds to the insertion of ‘0’ in position $t - 1$ in $\tilde{\mathbf{c}}$. Both \mathbf{c} and its bitwise complement map into the same string under T_n . Suppose one could construct a subset of binary strings of length $n - 1$ immune to r insertions of ‘0’ in the sense that no two strings in this set give rise to the same string after r insertions of ‘0’. Taking the inverse image of such a set under T_n , one gets a collection of strings of length n immune to r repetitions. The following construction of subsets of binary strings immune to r ‘0’-insertions generalizes one that was presented in [3], [4].

Fix $m \geq 1$, $r \geq 1$, and p a prime. For $1 \leq w \leq m$, $\mathbf{a} = (a_1, \dots, a_r)$ an integer vector, and $\mathbf{f} = (f_1, \dots, f_{w+1})$ an integer vector such that $f_i \bmod p \neq f_j \bmod p$ for $i \neq j$, let the set $\tilde{S}(m, w, \mathbf{a}, \mathbf{f}, p)$ be defined as:

$$\begin{aligned} \tilde{S}(m, w, \mathbf{a}, \mathbf{f}, p) = \{ \mathbf{s} = (s_1, s_2, \dots, s_m) \in \{0, 1\}^m : \\ & v_0 = 0, v_{w+1} = m + 1, \\ & v_i \text{ is the position of the } i^{\text{th}} \text{ 1 in } \mathbf{s} \text{ for } 1 \leq i \leq w, \\ & b_i = v_i - v_{i-1} - 1 \text{ for } 1 \leq i \leq w + 1, \\ & \sum_{i=1}^m s_i = w, \\ & \sum_{i=1}^{w+1} f_i b_i \equiv a_1 \pmod{p}, \\ & \sum_{i=1}^{w+1} (f_i)^2 b_i \equiv a_2 \pmod{p}, \\ & \vdots \\ & \sum_{i=1}^{w+1} (f_i)^r b_i \equiv a_r \pmod{p} \}. \end{aligned} \quad (2)$$

¹We think of r as being fixed, but in practice one could think of it as an overestimate on the number of repetitions.

The strings \mathbf{s} in this set all have length m and weight w . Here b_i denotes the size of the i th “bin” of zeros in \mathbf{s} and f_i is the weight assigned to that bin.

The set $\tilde{S}(m, 0, \mathbf{0}, \mathbf{0}, p)$, by convention, contains just the all-zeros string. Let $\mathbf{a}_0 = \mathbf{0}$. Given p_1, \dots, p_m , $\mathbf{a}_1, \dots, \mathbf{a}_m$, and $\mathbf{f}_1, \dots, \mathbf{f}_m$, let $\tilde{S}(m, (\mathbf{a}_1, \mathbf{f}_1, p_1), (\mathbf{a}_2, \mathbf{f}_2, p_2), \dots, (\mathbf{a}_m, \mathbf{f}_m, p_m))$ be defined as $\bigcup_{k=0}^m \tilde{S}(m, k, \mathbf{a}_k, \mathbf{f}_k, p_k)$. We have:

Lemma 1: If each p_k is a prime and $p_k > \max(r, k)$, the set $\tilde{S}(m, (\mathbf{a}_1, \mathbf{f}_1, p_1), (\mathbf{a}_2, \mathbf{f}_2, p_2), \dots, (\mathbf{a}_m, \mathbf{f}_m, p_m))$ is r -insertions of zeros correcting.

Proof: It suffices to show that each non-empty set $\tilde{S}(m, k, \mathbf{a}_k, \mathbf{f}_k, p_k)$ is r -insertions of zeros correcting. The proof follows verbatim from Lemma 3 in [3] with the substitutions of f_i for i . ■

As in [4] one can also show that the cardinality of this set is within a constant of the best known upper bound for such sets [6].

III. SOME NUMBER THEORETIC CONSTRUCTIONS

Given $r \geq 1$ let P be a prime number such that $\text{lcm}(2, 3, \dots, r) \mid (P-1)$. Then, in the residue set $\text{mod } P$, there are $\frac{P-1}{i}$ elements that are i th power residues and each has i distinct roots, [1]. For convenience, let $G = \lfloor \log_2(P) \rfloor$.

For each i , $1 \leq i \leq r$, we will construct, for P large enough, a subset V_i of the i th power residues $\text{mod } P$, of size that is logarithmic in P , such that all other residues can be expressed as a sum of a subset of elements of V_i . The set of the i th roots of the elements of the set V_i will be denoted F_i and will be made disjoint across all i . Thus, F_i will also have size logarithmic in P . The elements of $M = \bigcup_{i=1}^r F_i \cup \{0\}$ will be reserved for the weights f_i of the bins of zeros of the prefix part in the transformed version of the concatenation of the prefix and the codeword, where the transform is that given by (1). Note that M also has size that is logarithmic in P , and since each bin in the prefix will have at most one zero, the length of the prefix is also logarithmic in P . The sets V_i will serve to satisfy the i th congruency constraint of the type given in (2) for the transformed domain version of the string consisting of the concatenation of the prefix and the codeword, as explained in Section IV. The special weight 0 in the set M will serve to ensure that for each codeword the concatenation of it with the prefix results in a string of fixed length.

We also prove that if n is large enough P can be chosen in an interval whose end points depend linearly on n , so the constructed prefix will have length logarithmic in n .

Let $[x]_P$ indicate the residue mod P congruent to x .

Lemma 2: For an integer P , each residue $u \text{ mod } P$ can be expressed as a sum of a subset of elements of the set $T_{z,P} = \{[z]_P, [2z]_P, [2^2z]_P, \dots, [2^G z]_P\}$ where $G = \lfloor \log_2 P \rfloor$, and z is an arbitrary non zero residue mod P .

Proof: To see this for $T_{1,P}$, consider the binary expansion of u . The proof for $T_{z,P}$ follows by multiplying throughout by z . ■

Lemma 3: Suppose P is a prime number such that $i \mid (P-1)$. 1). Suppose the equation $x^i \equiv a \text{ mod } P$ has a solution,

$1 \leq a \leq P-1$. Then the equation $x^i \equiv a \text{ mod } P$ has i distinct solutions, and we may call them x_1 through x_i . The sum $\sum_{k=1}^i x_k^j \equiv 0 \text{ mod } P$ for $1 \leq j \leq i-1$.

Proof: This is a standard fact in number theory [1]. ■

For a prime number P for which $i \mid (P-1)$, and $2 \leq i < P-1$, let $Q_i(P)$ be the set of distinct i th power residues mod P . We also state the following convenient result.

Lemma 4: For a prime P such that $i \mid (P-1)$ and $i \geq 2$, each residue $u \text{ mod } P$ can be expressed as a sum of two distinct elements of $Q_i(P)$ in at least $P/(2i^2) - \sqrt{P}/2 - 3$ ways.

Proof: The result follows from Theorem II in [5] which states that over $GF(P)$ the equation $x^i + y^i = a$, where $x, y, a \in GF(P)$ and nonzero, and $0 < i < P-1$, has at least

$$\frac{(P-1)^2}{P} - P^{-1/2} (1 + (i-1)P^{1/2})^2 \quad (3)$$

solutions. Rearrange the terms in (3) to conclude that the equation $x^i + y^i = a$ has at least

$$P - (i-1)^2 \sqrt{P} - 2(i-1) - 2 + \frac{1}{P} - \frac{1}{\sqrt{P}} \quad (4)$$

solutions. Noting that i distinct values of x result in the same x^i , accounting for the symmetry of x and y , and omitting the case $x^i = y^i$ we obtain a lower bound on the number of ways a residue can be expressed as a sum of two distinct i th power residues to be $P/(2i^2) - \sqrt{P}/2 - 3$. ■

We now continue with the introduction of some convenient notation. For $x_{i,1}$ an i th power residue define the set $A_{i,1}(x_{i,1})$ to be

$$A_{i,1}(x_{i,1}) = \{[2^{ik} x_{i,1}]_P \mid 0 \leq k \leq \lfloor \frac{G}{i} \rfloor\}. \quad (5)$$

Let $x_{i,2}$ and $x_{i,3}$ be distinct i th power residues such that $x_{i,2} + x_{i,3} \equiv 2x_{i,1} \text{ mod } P$. Likewise, for each $2^l x_{i,1}$ for $1 \leq l \leq i-1$ let $x_{i,2l}$ and $x_{i,2l+1}$ be distinct i th power residues such that $x_{i,2l} + x_{i,2l+1} \equiv 2^l x_{i,1} \text{ mod } P$. These residues generate the sets $A_{i,2l}(x_{i,2l})$ and $A_{i,2l+1}(x_{i,2l+1})$ where

$$A_{i,2l}(x_{i,2l}) = \{[2^{ik} x_{i,2l}]_P \mid 0 \leq k \leq \lfloor \frac{G-l}{i} \rfloor\} \text{ and} \quad (6)$$

$$A_{i,2l+1}(x_{i,2l+1}) = \{[2^{ik} x_{i,2l+1}]_P \mid 0 \leq k \leq \lfloor \frac{G-l}{i} \rfloor\}. \quad (7)$$

By introducing sets $A_{i,j}(x_{i,j})$ we have effectively decomposed all residues of the type $[2^{ik+l} x_{i,1}]_P$, $0 \leq ik+l \leq G$, $1 \leq l \leq i-1$, into a sum of two i th power residues, namely $[2^{ik} x_{i,2l}]_P$ and $[2^{ik} x_{i,2l+1}]_P$. For each set $A_{i,j}(x_{i,j})$, $1 \leq j \leq 2i-1$, we let $B_{i,j}(x_{i,j})$ be the set of all i th power roots of elements of $A_{i,j}(x_{i,j})$,

$$B_{i,j}(x_{i,j}) = \left\{ [2^k y_{i,j}^{(t)}]_P \mid (y_{i,j}^{(t)})^i \equiv x_{i,j} \text{ mod } P, \right. \\ \left. 1 \leq t \leq i, 0 \leq k \leq \lfloor \frac{G-\lfloor \frac{j}{2} \rfloor}{i} \rfloor \right\}. \quad (8)$$

First note that all elements in $A_{i,j}(x_{i,j})$ are i th power residues by construction. Moreover, they are all distinct since $2^{ij_1} \neq 2^{ij_2} \text{ mod } P$ for $1 \leq j_1, j_2 \leq \lfloor \frac{G-\lfloor \frac{j}{2} \rfloor}{i} \rfloor$ for $j_1 \neq j_2$ implies $x_{i,j} 2^{2^{ij_1}} \neq x_{i,j} 2^{2^{ij_2}} \text{ mod } P$. Thus, $|A_{i,j}(x_{i,j})| = \lfloor \frac{G-\lfloor \frac{j}{2} \rfloor}{i} \rfloor + 1$

and since the i th power roots of distinct i th power residues are themselves distinct, $|B_{i,j}(x_{i,j})| = i \left(\lfloor \frac{G - \lfloor \frac{j}{2} \rfloor}{i} \rfloor + 1 \right)$.

Lemma 5: Suppose P is a prime number such that $i|(P-1)$. Let $x_{i,1}$ be an i th power residue. Suppose $x_{i,j}$ for $2 \leq j \leq 2i-1$ are i th power residues such that $2^k x_{i,1} \equiv x_{i,2k} + x_{i,2k+1} \pmod{P}$ for $1 \leq k \leq (i-1)$. Let $A_{i,j}(x_{i,j}) = \{[2^{il}x_{i,j}]_P | 0 \leq l \leq \lfloor \frac{G - \lfloor \frac{j}{2} \rfloor}{i} \rfloor\}$ for $1 \leq j \leq 2i-1$ and $G = \lfloor \log_2 P \rfloor$. If the sets $A_{i,j}(x_{i,j})$ are disjoint for $1 \leq j \leq 2i-1$, each residue $u \pmod{P}$ can be expressed as a sum of a subset of elements of the set $L_{z,P} = \bigcup_{j=1}^{2i-1} A_{i,j}(x_{i,j})$ where z denotes $x_{i,1}$.

Proof: The statement follows from Lemma 2 by observing that, with z denoting $x_{i,1}$, the elements $[2^k z]_P$ in the set $T_{z,P}$ for which $i \nmid k$ are each decomposed into a sum of two component elements such that all component elements are distinct from one another and distinct from $[2^k z]_P$ for which $i|k$. ■

For $i \geq 2$, let $W_i(u)$ denote the number of ways a residue $u \pmod{P}$ can be expressed as a sum of two distinct non zero i th power residues \pmod{P} . A universal lower bound on $W_i(u)$ that holds for all residues u will be referred to as W_i . One such bound was given in Lemma 4. A condition on W_1 is not needed as there is no need to decompose residues into a sum of two other residues for the $i=1$ level.

Lemma 6: For a given integer r , suppose a prime number P satisfies $\text{lcm}(2, 3, \dots, r)|(P-1)$. Let $G = \lfloor \log_2 P \rfloor$. If $P-1 > (G+r)(G+r-1)(r-1)^2$ and $W_i > 2i(G+i)(G+i-1)$, for each i in the range $2 \leq i \leq r$, there exist subsets $A_{i,j}(x_{i,j})$ of the type given in (5), (6) and (7) and $B_{i,j}(x_{i,j})$ of the type given in (8) such that for fixed i subsets $A_{i,j}(x_{i,j})$ for $1 \leq j \leq 2i-1$ are disjoint, and for $1 \leq i \leq r$, $1 \leq j \leq 2i-1$ all subsets $B_{i,j}(x_{i,j})$ are disjoint.

Proof: We inductively build the sets $A_{i,j}(x_{i,j})$ and $B_{i,j}(x_{i,j})$ for $1 \leq i \leq r$ and $1 \leq j \leq 2i-1$, starting with the level $i=1$. Let $x_{1,1}$ be an arbitrary residue \pmod{P} , and let $A_{1,1}(x_{1,1}) = \{[2^k x_{1,1}]_P | 0 \leq k \leq G\}$. Let $z_1 = x_{1,1}$ and $y_{1,1}^{(1)} = x_{1,1}$ so that $B_{1,1}(z_1)$ is simply $A_{1,1}(x_{1,1})$. All elements in $B_{1,1}(z_1)$ are distinct and $|B_{1,1}(z_1)| = (G+1)$.

Suppose the disjoint sets $B_{k,j}(x_{k,j})$ for $1 \leq k < i$, $1 \leq j \leq 2k-1$ are already successfully constructed. Consider $x_{i,1}$ an i th power residue \pmod{P} with the property that the set $B_{i,1}(x_{i,1})$ is disjoint from all of $B_{k,j}(x_{k,j})$ for $1 \leq k < i$, $1 \leq j \leq 2k-1$.

These constraints on disjointness prevent no more than $(\frac{G+i}{i})(\frac{G+k}{k})$ choices for $x_{i,1}$ for each $y_{k,j}^{(t)}$ where $1 \leq k \leq i-1$, $1 \leq j \leq 2k-1$, and $1 \leq t \leq k$, since $|B_{i,1}(x_{i,1})| = \lfloor \frac{G}{i} \rfloor + 1 \leq \frac{G+i}{i}$, and $|B_{k,j}(x_{k,j})| = \lfloor \frac{G - \lfloor \frac{j}{2} \rfloor}{k} \rfloor + 1 \leq \frac{G+k}{k}$. Summing over all choices it follows that at most

$$\left(\frac{G+i}{i}\right) \sum_{k=1}^{i-1} (2k-1)k \left(\frac{G+k}{k}\right) \leq (G+i) \left(\frac{G+i-1}{i}\right) (i-1)^2 \quad (9)$$

i th power residues cannot be chosen for $x_{i,1}$. Since there are $\frac{P-1}{i}$ i th power residues, it is sufficient that

$$P-1 > (G+i)(G+i-1)(i-1)^2 \quad (10)$$

for such $x_{i,1}$ to exist. Note that since the expression on the right hand side of the inequality (10) is an increasing function

of positive i , each subsequent level poses a lower bound on P that subsumes all previous ones. It is thus sufficient to have $P-1 > (G+r)(G+r-1)(r-1)^2$, as given in the statement of the Lemma.

Consider $x_{i,2}$ and $x_{i,3}$ as distinct i th power residues \pmod{P} that satisfy $x_{i,2} + x_{i,3} \equiv 2x_{i,1} \pmod{P}$ for a previously chosen $x_{i,1}$. We require that $x_{i,2}$ and $x_{i,3}$ give rise to sets $B_{i,2}(x_{i,2})$ and $B_{i,3}(x_{i,3})$ that are disjoint and that are disjoint from each of $B_{k,j}(x_{k,j})$ for $1 \leq k < i$, $1 \leq j \leq 2k-1$ and from $B_{i,1}(x_{i,1})$. By construction, if the sets $B_{i,1}(x_{i,1})$, $B_{i,2}(x_{i,2})$, and $B_{i,3}(x_{i,3})$ are disjoint, then so are sets $A_{i,1}(x_{i,1})$, $A_{i,2}(x_{i,2})$, and $A_{i,3}(x_{i,3})$. Since $|B_{i,2}(x_{i,2})| = |B_{i,3}(x_{i,3})| = \lfloor \frac{G-1}{i} \rfloor + 1 \leq \frac{G+i-1}{i}$, and $|B_{k,j}(x_{k,j})| = \lfloor \frac{G - \lfloor \frac{j}{2} \rfloor}{k} \rfloor + 1 \leq \frac{G+k}{k}$, constraints based on the previously encountered $y_{j,k}^{(t)}$ for $1 \leq k < i$, $1 \leq j \leq 2k-1$, $1 \leq t \leq k$ prevent at most $(\frac{G+i-1}{i})(\frac{G+j}{j})$ choices for each of $x_{i,2}$ and $x_{i,3}$, for each $y_{j,k}^{(t)}$. Combined with the restriction on the disjointness with $B_{i,1}(x_{i,1})$ and the requirement that $B_{i,2}(x_{i,2})$ and $B_{i,3}(x_{i,3})$ be nonintersecting, it follows that

$$W_i > 2 \left(\frac{G+i-1}{i}\right) \left[\sum_{k=1}^{i-1} (2k-1)k \left(\frac{G+k}{k}\right) + \left(\frac{G+i}{i}\right) \right] + \left(\frac{G+i-1}{i}\right)^2 \quad (11)$$

is sufficient for the pair $(x_{i,2}, x_{i,3})$ to exist.

Likewise, for $x_{i,2l}$ and $x_{i,2l+1}$ to be distinct i th power residues \pmod{P} that satisfy $x_{i,2l} + x_{i,2l+1} \equiv 2^l x_{i,1} \pmod{P}$, that give rise to disjoint sets $B_{i,2l}(x_{i,2l})$ and $B_{i,2l+1}(x_{i,2l+1})$ and that are also disjoint from all previously constructed set $B_{k,j}(x_{k,j})$, it suffices that

$$W_i > 2 \left(\frac{G+i-1}{i}\right) \left[\sum_{k=1}^{i-1} (2k-1)k \left(\frac{G+k}{k}\right) + (2l-1) \left(\frac{G+i}{i}\right) \right] + \left(\frac{G+i-1}{i}\right)^2 \quad (12)$$

for the pair $(x_{i,2l}, x_{i,2l+1})$ to exist. Since at each level i we construct $i-1$ pairs $x_{i,2l}$ and $x_{i,2l+1}$, and since the right hand side of (12) is an increasing function of l , it is sufficient to upper bound the expression in (12) for $l=i-1$,

$$W_i > 2 \left(\frac{G+i-1}{i}\right) \left[(i-1)^2 (G+i) + \frac{2i-3}{i} (G+i) \right] + \left(\frac{G+i-1}{i}\right)^2. \quad (13)$$

Some simplification of (13) yields

$$W_i > 2i(G+i)(G+i-1) \quad (14)$$

as a sufficient condition for the disjoint sets $B_{i,j}(x_{i,j})$ to exist that are also disjoint from all sets $B_{k,l}(x_{k,l})$ for $k < i$. Thus, with the appropriate lower bounds on P and W_i 's, it is possible to construct disjoint sets $B_{i,j}(x_{i,j})$. ■

When the conditions of Lemma 6 hold, note that all residues \pmod{P} can be expressed as a sum of a subset of elements of $V_i := \bigcup_{j=1}^{2i-1} A_{i,j}(x_{i,j})$ by Lemma 5 for each i , $1 \leq i \leq r$. Also note that $|V_i|$ scales as $\log_2(P)$, since $|A_{i,j}(x_{i,j})| = \lfloor \frac{G - \lfloor \frac{j}{2} \rfloor}{i} \rfloor + 1$. For $F_i := \bigcup_{j=1}^{2i-1} B_{i,j}(x_{i,j})$, $|F_i|$ also scales as $\log_2(P)$, since $|B_{i,j}(x_{i,j})| = i \left(\lfloor \frac{G - \lfloor \frac{j}{2} \rfloor}{i} \rfloor + 1 \right)$.

We now discuss how large prime P needs to be so that the conditions of Lemma 6 hold. Namely we require

$$P-1 > (r-1)^2 (G+r)(G+r-1) \quad (15)$$

and

$$W_i > 2i(G+i)(G+i-1) \text{ for } 2 \leq i \leq r. \quad (16)$$

Using Lemma 4 it follows that it is sufficient that

$$P > 4r^3(G+r)(G+r-1) + r^2\sqrt{P} + 6r^2, \text{ for } r \geq 2 \quad (17)$$

for (16) to hold. Moreover, if (17) holds, it implies (15). For $r = 1$, the requirement is $P > 1$. Since the elements of $M := \bigcup_{i=1}^r F_i \cup \{0\}$ are to be reserved for the indices of bins of zeros of the prefix in the transformed domain we also require that $P - n > |M|$, since the total number of bins of zeros to be used is at most n (from the original string) + $|M|$ (from the prefix), and each bin receives a distinct index. Since $F_i = \bigcup_{j=1}^{2i-1} B_{i,j}(x_{i,j})$ and $|B_{i,j}(x_{i,j})| = i \left(\lfloor \frac{G-i}{i} \rfloor + 1 \right)$, whereby $i \left(\frac{G-i}{i} \right) \leq |B_{i,j}(x_{i,j})| \leq i \left(\frac{G+i}{i} \right)$, it follows that

$$\begin{aligned} |M| &\leq \sum_{i=1}^r (2i-1)(G+i) + 1 \leq r^2(G+r) + 1 \text{ and} \\ |M| &\geq \sum_{i=1}^r (2i-1)(G-i) + 1 \geq r^2(G-r) + 1. \end{aligned} \quad (18)$$

The top inequality in (18) along with the constraint $P - n > |M|$ yields a sufficient requirement on how large P needs to be in terms of n ,

$$P > n + r^2(\log_2(P) + r) + 1. \quad (19)$$

We thus seek a prime P which satisfies (17) and (19), satisfies $\text{lcm}(2, 3, \dots, r) | (P-1)$ and lies in the interval that linearly depends on n . This can be done using known estimates on the prime counting function for arithmetic progressions [7], [8]. The details are suppressed for lack of space. For the details, see [9].

IV. PREFIXING ALGORITHM

We now show how to injectively transform a given collection \mathcal{C} of binary strings of length n (i.e. a code) into another collection $T(\mathcal{C})$ of binary strings of equal length, such that the collection $T(\mathcal{C})$ is guaranteed to be immune to the prescribed number r of repetition errors. The procedure takes an element \mathbf{c} of \mathcal{C} and produces a string $\mathbf{t}_c = [\mathbf{p}_c \mathbf{c}]$, such that \mathbf{t}_c after the transformation (1) (of appropriate length) into $\tilde{\mathbf{t}}_c$ satisfies the conditions of the form given by (2). We will choose \mathbf{p}_c so that in the concatenation $[\mathbf{p}_c \mathbf{c}]$, the last bit of \mathbf{p}_c is the complement of the first bit of \mathbf{c} ; this prevents interference between the prefix from the original codeword when going through the transformation (1). Let P be a prime number for which $\text{lcm}(2, 3, \dots, r) | (P-1)$, which lies in an interval that scales linearly with n , and satisfies the conditions of Lemma 6. Sufficient conditions for the existence of such P were discussed in the previous section. Recall that $M = \bigcup_{i=1}^r F_i \cup \{0\}$ denotes the set of indices of bins of zeros reserved for the prefix, where $F_i = \bigcup_{j=1}^{2i-1} B_{i,j}(x_{i,j})$, and $B_{i,j}(x_{i,j})$ are given in (8), and are constructed such that all sets $B_{i,j}(x_{i,j})$ for $1 \leq i \leq r$, $1 \leq j \leq 2i-1$ are nonintersecting. Let $L = |M|$ and let $L+N$ be the total

number of bins of zeros of $\tilde{\mathbf{t}}_c$. Let

$$\begin{aligned} a'_1 &\equiv \sum_{i=L+1}^{L+N} b_i f_i \text{ mod } P, \\ a'_2 &\equiv \sum_{i=L+1}^{L+N} b_i f_i^2 \text{ mod } P \\ &\vdots \\ a'_r &\equiv \sum_{i=L+1}^{L+N} b_i f_i^r \text{ mod } P \end{aligned} \quad (20)$$

where b_i is the size of the i th bin of zeros in $\tilde{\mathbf{t}}_s$, and f_i in (20) are chosen in the increasing order from the set $R_P \setminus M$ for R_P the set of all residues $\text{mod } P$. By the choice of P , the set $R_P \setminus M$ is large enough to accommodate such f_i 's. We may think of a'_1 through a'_r as the contribution of the (transformed) original string to the overall congruency value, since the i th bin of zeros in $\tilde{\mathbf{t}}_c$ for $L+1 \leq i \leq L+N$ is the j th bin of zeros in $\tilde{\mathbf{c}}$ (where $\tilde{\mathbf{c}} = \mathbf{c}T_n$) for $j = i - L$, since no run spans both \mathbf{p}_c and \mathbf{c} .

We now show that it is always possible to achieve

$$\begin{aligned} a_1 &\equiv \sum_{i=1}^{L+N} b_i f_i \text{ mod } P, \\ a_2 &\equiv \sum_{i=1}^{L+N} b_i f_i^2 \text{ mod } P, \\ &\vdots \\ a_r &\equiv \sum_{i=1}^{L+N} b_i f_i^r \text{ mod } P, \end{aligned} \quad (21)$$

for arbitrary but fixed values a_1 through a_r irrespective of the values a'_1 through a'_r , where b_i is either 0 or 1 for $1 \leq i \leq L-1$, and where $f_L = 0$. By Lemma 2 and letting $p_k = P$ for all weights k , this set of congruential constraints is sufficient to ensure the immunity to r insertions of zeros, and in fact to recover the original codeword despite any r insertions of zero in $\tilde{\mathbf{t}}_c$.

The encoding procedure is recursive and proceeds as follows. Let l be the l th level of recursion for $l = 1$ to $l = r$. The l th level ensures that the l th congruency constraint in (21) is satisfied without altering the previous $l-1$ levels. At each level l , starting with $l = 1$ and while $l \leq r$:

- 1) Select a subset T_l of $F_l = \bigcup_{j=1}^{2l-1} B_{l,j}(x_{l,j})$ such that $\sum_{k \in T_l} k^l \equiv a_l - a'_l - \sum_{i=1}^{l-1} d_{l,i} \text{ mod } P$, and such that if an element $y, y^l \equiv z \text{ mod } P$ of $B_{l,j}(x_{l,j})$ is selected, then so are all other $l-1$ l th roots of z (which are also elements of $B_{l,j}(x_{l,j})$ by construction). For $l = 1$, $\sum_{k \in T_1} k \equiv a_1 - a'_1 \text{ mod } P$.
- 2) Let $d_{l,j} \equiv \sum_{k \in T_l} k^j \text{ mod } P$ for $l+1 \leq j \leq r$.
- 3) For each i , $1 \leq i \leq |F_l|$, for which $f_i \in T_l$ we set $b_i = 1$, and for each i , for which $f_i \notin T_l$ we set $b_i = 0$.
- 4) Proceed to level $l+1$.

After the level r is completed, let $b_L = \sum_{l=1}^r (|F_l| - |T_l|)$. The purpose of this bin with weight zero is to ensure that the overall string \mathbf{t}_c has the same length irrespective of the structure of the starting string \mathbf{c} .

The existence of $T_l, T_l \subseteq F_l$ in Step 1) follows from Lemmas in Section III. In particular, recall that each residue $\text{mod } P$ can be expressed as a sum of a subset L_l of $\bigcup_{j=1}^{2l-1} A_{l,j}(x_{l,j})$, by Lemma 5. We then let T_l consist of all l th power roots of elements in L_l . By construction, T_l is the union of appropriate subsets of sets $B_{l,j}(x_{l,j})$, whose l th powers are

precisely the elements of L_l , and these subsets are disjoint by construction.

Recall that the sets $B_{l,j}(x_{l,j})$ are constructed such that if an l th power root of a residue y belongs to $B_{l,j}(x_{l,j})$ then all l power roots of y also belong to $B_{l,j}(x_{l,j})$. Then, by Lemma 3 the contribution to each congruency sum for levels 1 through $l - 1$ of the elements of \tilde{F}_l is zero. Hence, once the target congruency value is reached for a particular level, it will not be altered by establishing congruencies at subsequent levels. As a result, each string \tilde{t}_c satisfies the congruency constraints given in (2).

As for the actual string that is transmitted, it is the unique one of the two preimages of \tilde{t}_c under the transformation of type (1) which equals the codeword in the last n bits. The set of all such strings constructed from \mathcal{C} in this manner is the set $T(\mathcal{C})$. This set of strings is immune to r repetitions, and the additional redundancy introduced by the prefix is asymptotically negligible.

V. DECODING ALGORITHM

In the infinite SNR regime that motivated the repetitions model the construction allows one to recover the original codeword despite any r repetitions, since one can just solve for the bin locations of the corresponding insertions of zeros in the transformed domain version of the concatenation of the prefix and the codeword. In practice, however, a codeword together with its prefix would be transmitted over a noisy channel, so the effects of poor timing recovery would be seen on the corrupted extended string rather than directly on the extended string. Since the proposed approach will first put the prefix in place and then put the extended string through the noisy channel, immunity to repetitions in the transmitted concatenated string does not a priori guarantee good repetition immunity for repetitions in the post-noise string. This is to be contrasted with the situation in the construction for providing immunity to repetition in array-based codes that we presented in [3]: there the idea was to *thin* the original code, so that the transmitted strings are codewords in the original code, and we presented a modified message passing algorithm for decoding the thinned code over a channel allowing for repetitions that showed good performance. The construction in [3] intimately exploited algebraic properties specific to the array-based codes, which are not available for arbitrary binary block codes.

In the doctoral thesis [2] a modified message passing algorithm for decoding from the post-noise and post-repetition version of the transmitted string (the concatenation of the prefix and the codeword) was proposed for the current construction. This algorithm is also available in [9]. Preliminary simulation results with that algorithm are not satisfactory. This is to be expected, since the present construction is not very robust to noise in the prefix part.

Since the prefix is of negligible length in comparison with the length of the codeword, one can protect the prefix from noise and repetitions while still having asymptotically negligible additional redundancy, so that the prefix can be

viewed as providing side information about the run length structure of the original codeword at the decoder. We are currently working on developing message passing algorithms to recover the original codeword from its post-noise post-repetition version while using such side information provided by the prefix and expect to have results to report along these lines at the time of the conference.

VI. CONCLUSION

We presented a general method for providing immunity to any fixed number of repetition errors to any binary block code. This scheme relies on introducing a prefix for each original binary string such that the resulting strings belong to a set previously shown to be immune to repetition errors. The prefix length is only logarithmic in the size of the original code, and thus produces asymptotically negligible additional redundancy.

Ability to recover the original codeword despite any pattern of the designed number of repetitions is guaranteed in the high SNR regime which motivated the repetitions model. In practice a codeword together with its prefix would be transmitted over a noisy channel, so the effects of poor timing recovery would be seen on the corrupted extended string rather than directly on the extended string. In such a scenario, protecting the prefix involves asymptotically negligible additional cost and allows one to treat the prefix as providing side information at the decoder about the run length structure of the codeword. We are developing message passing based decoding algorithms to decode the post-noise post-repetition codewords while exploiting such side information.

ACKNOWLEDGMENT

The work of the both authors was supported by NSF grants CCF-0500234 and CCF-0635372, by Marvell Semiconductor, and by the University of California MICRO program. The work of the first author was also supported by a Dissertation Year Fellowship from the UCOP. The work of the second author was also supported by NSF grant CNS-0627161.

REFERENCES

- [1] T. M. Apostol, *Introduction to Analytic Number Theory*, Springer-Verlag, NY, 1976.
- [2] L. Dolecek, *Rethinking the Minimum Distance: Channels with Varying Sampling Rate and Iterative Decoding of LDPC Codes*, PhD dissertation, EECS Department, University of California, Berkeley, Dec. 2007.
- [3] L. Dolecek and V. Anantharam, "A synchronization technique for array-based LDPC Codes in channels with varying sampling rates," *ISIT 2006*, Seattle, Washington, U.S.A., July 2006, pp. 2057–2061.
- [4] L. Dolecek and V. Anantharam, "On subsets of binary strings immune to multiple repetition errors," *ISIT 2007*, Nice, France, June 2007, pp. 1691–1695.
- [5] L. K. Hua and H. S. Vandiver, "Characters over certain types of rings with applications to the theory of equations in a finite field", *Proc. Nat. Acad. Sci. USA*, vol. 35, pp. 481–487, 1949.
- [6] V. I. Levenshtein, "Binary codes capable of correcting spurious insertions and deletions of ones", *Problems of Information Transmission*, vol. 1(1), pp. 8–17, Jan. 1965.
- [7] O. Ramare and R. Rumely, "Primes in arithmetic progressions", *Mathematics of Computation*, vol. 65, No. 213, pp. 397–425, Jan. 1996.
- [8] I. Soprounov, "A short proof of the prime number theorem for arithmetic progressions", available online at <http://www.math.umass.edu/~isoprou/pdf/primes.pdf>
- [9] L. Dolecek and V. Anantharam, "On a prefixing method for multiple repetition error correction", available online at <http://web.mit.edu/dolecek/www/workingPapers/>