

A Relationship Between Linear Complexity and k -Error Linear Complexity

Kaoru Kurosawa, *Member, IEEE*, Fumio Sato, Takahiro Sakata, and Wataru Kishimoto

Abstract—The k -error linear complexity of a periodic sequence of period N is defined as the smallest linear complexity that can be obtained by changing k or fewer bits of the sequence per period. This correspondence shows a relationship between the linear complexity and the minimum value k for which the k -error linear complexity is strictly less than the linear complexity.

Index Terms—Cryptography, linear complexity, stream cipher.

I. INTRODUCTION

Linear complexity is an important cryptographic criterion of stream ciphers [5]. The linear complexity of a sequence (Y) , denoted by $LC(Y)$, is defined as the length of the shortest linear feedback shift register (LFSR) that generates (Y) . In a stream cipher, a keystream sequence (K) must have large linear complexity L because Berlekamp-Massey algorithm [3] can efficiently find the LFSR that generates (K) after examining at most $2L$ consecutive bits of (K) .

However, a high linear complexity does not necessarily ensure that a sequence is cryptographically secure. For example, the sequence

$$(K) = (\underbrace{0, \dots, 0, 1, 0, \dots, 0, 1, \dots}_{N}, \dots)$$

has the maximum possible linear complexity N , but it is obviously cryptographically weak. After changing every N th bit of the original sequence, its linear complexity decreases to zero. This shows that sequences with high linear complexity may be well approximated by sequences with very low linear complexity.

From this observation, Ding, Xiao, and Shan introduced the notion of k -error linear complexity [1]. (Actually, they called it sphere complexity.) The k -error linear complexity of a periodic sequence (Y) of period N , denoted by $LC_k(Y)$, is defined as the smallest linear complexity that can be obtained by changing k or fewer bits of the sequence per period. They showed some bounds on k -error linear complexity for several sequences [1]. Independently, Stamp and Martin showed a polynomial time algorithm which computes k -error linear complexity $LC_k(Y)$ for period $N = 2^n$ [6].

This correspondence shows a relationship between the linear complexity and the minimum value k for which the k -error linear complexity is strictly less than the linear complexity. For a periodic sequence (Y) of period N , we define $minerror(Y)$ as the minimum value of k such that

$$LC_k(Y) < LC(Y).$$

Manuscript received November 16, 1998; revised August 7, 1999.

K. Kurosawa, F. Sato, and T. Sakata are with the Department of Electrical and Electronic Engineering, Faculty of Engineering, Tokyo Institute of Technology 2-12-1 O-okayama, Meguro-ku, Tokyo 152, Japan.

W. Kishimoto is with the Department of Information and Image Sciences, Faculty of Engineering, Chiba University 1-33 Yayoi-cho, Inage-ku, Chiba-shi, Chiba 263-8522, Japan.

Communicated by D. Stinson, Associate Editor for Complexity and Cryptography.

Publisher Item Identifier S 0018-9448(00)01681-3.

In other words, $minerror(Y)$ is the smallest Hamming weight of an error vector E^N such that

$$LC(Y + E) < LC(Y)$$

where E^N denotes one period of (E) .

We first prove that

$$minerror(Y) = 2^{W_H(N - LC(Y))}$$

for binary sequences (Y) with period $N = 2^n$, where $W_H(N - LC(Y))$ denotes the Hamming weight of the binary representation of $N - LC(Y)$. The condition $N = 2^n$ implies that $1 - x^N = (1 - x)^N$ over $GF(2)$, which makes our analysis easy. Massey, Costello, and Justesen [4] derived a property of the weight of $(1 - x)^s f(x)$ and applied this property to error-correcting code constructions. Our analysis is based on this property.

Next, for $k = minerror(Y)$, we show a necessary and sufficient condition for $LC_k(Y) = 0, 1, 2$ and $LC_k(Y) > 2$ in terms of the Hamming weight of (Y) . We also present an upper bound on $LC_k(Y)$.

The above results are easily generalized to nonbinary sequences (Y) over $GF(p^r)$ with period $N = p^n$.

Finally, we completely determine the k -error linear complexities $LC_k(Y)$ of binary m -sequences (Y) with the linear complexity $LC(Y) = prime$.

II. LINEAR COMPLEXITY AND k -ERROR LINEAR COMPLEXITY

For a sequence $Y^N = (y_0, y_1, \dots, y_{N-1})$ of length N over $GF(q)$, let

$$y^N(x) \triangleq y_0 + y_1x + \dots + y_{N-1}x^{N-1}.$$

Let (Y) denote the periodic sequence obtained by appending the copies of Y^N . For (Y) , let

$$\begin{aligned} y(x) &\triangleq y^N(x) + y^N(x)x^N + y^N(x)x^{2N} + \dots \\ &= y^N(x)/(1 - x^N). \end{aligned}$$

For example, for a binary sequence

$$Y^{21} = (111110101001100010000)$$

we have

$$\begin{aligned} y^{21}(x) &= 1 + x + x^2 + x^3 + x^4 + x^6 + x^8 + x^{11} + x^{12} + x^{16} \\ y(x) &= (1 + x + x^2 + x^3 + x^4 + x^6 + x^8 + x^{11} \\ &\quad + x^{12} + x^{16})/(1 - x^{21}) \\ &= 1/(1 + x + x^5). \end{aligned}$$

Then it is known that

$$LC(Y) = \deg(1 + x + x^5) = 5.$$

Generally, the following proposition holds.

Proposition 2.1: (See, for example, [1].) Let

$$g(x) \triangleq \gcd(1 - x^N, y^N(x)).$$

Then

$$LC(Y) = N - \deg g(x).$$

Lemma 2.1: For a binary sequence (Y) ;

- (1) $LC(Y) = 1$ if and only if $(Y) = (1, 1, 1, \dots)$.
- (2) $LC(Y) = 2$ if and only if $(Y) = (1, 0, 1, 0, \dots)$ or $(0, 1, 0, 1, \dots)$.

Proof: Note that

$$\begin{aligned} 1/(1-x) &= 1 + x + x^2 + \dots, \\ 1/(1-x^2) &= 1 + x^2 + x^4 + \dots, \\ x/(1-x^2) &= x + x^3 + x^5 + \dots. \end{aligned} \quad \text{Q.E.D.}$$

Next, for an error sequence

$$E^N = (e_0, e_1, \dots, e_{N-1})$$

let $W_H(E^N)$ denote the Hamming weight of E^N . Let $(Y + E)$ denote the periodic sequence obtained by appending the copies of $Y^N + E^N$. The k -error linear complexity of (Y) is defined by [1], [6]

$$LC_k(Y) \triangleq \min_{W_H(E^N) \leq k} LC(Y + E).$$

For example, $LC(Y) = N$ for a binary sequence $Y^N = (0, \dots, 0, 1)$. However, $LC_1(Y) = 0$ because

$$Y^N + (0, \dots, 0, 1) = (0, \dots, 0).$$

This shows that a high linear complexity does not ensure that a sequence is cryptographically secure. It also shows the importance of k -error linear complexity.

For $E^N = (e_0, e_1, \dots, e_{N-1})$, let

$$e^N(x) \triangleq e_0 + e_1x + \dots + e_{N-1}x^{N-1}.$$

We say that E^N is an error vector and $e^N(x)$ is an error polynomial.

III. POLYNOMIAL HAMMING WEIGHT [4]

For a nonnegative integer s , let $W_H(s)$ denote the Hamming weight of the binary representation of s . For

$$e(x) = 1 + x^{a_1} + \dots + x^{a_{q-1}}$$

define

$$W(e(x)) \triangleq q.$$

Massey, Costello, and Justesen proved the following propositions for polynomials over $GF(2)$.

Proposition 3.1 [4]:

$$W((1-x)^s) = 2^{W_H(s)}. \quad (1)$$

Proposition 3.2 [4]: For any polynomial $f(x)$ such that $f(1) = 1$

$$W((1-x)^s f(x)) \geq 2^{W_H(s)}. \quad (2)$$

IV. MINIMUM k OF k -ERROR LINEAR COMPLEXITY

We define $minerror(Y)$ as the minimum value of k such that

$$LC_k(Y) < LC(Y).$$

In other words, $minerror(Y)$ is the smallest Hamming weight of an error vector E^N such that

$$LC(Y + E) < LC(Y).$$

In this section, we prove that

$$minerror(Y) = 2^{W_H(N-LC(Y))}$$

for binary sequences (Y) with period $N = 2^n$. Since $N = 2^n$, we have

$$\begin{aligned} \gcd(1-x^N, y^N(x)) &= \gcd((1-x)^{2^n}, y^N(x)) \\ &= (1-x)^s \end{aligned}$$

for some s . That is,

$$y^N(x) = (1-x)^s y_1(x) \quad (3)$$

for some $y_1(x)$ such that $y_1(1) = 1 (\neq 0)$. Then from Proposition 2.1

$$LC(Y) = N - s. \quad (4)$$

Next, for an error vector E^N , suppose that

$$\gcd((1-x)^{2^n}, y^N(x) + e^N(x)) = (1-x)^z \quad (5)$$

for some z . Then

$$LC(Y + E) = N - z$$

from Proposition 2.1. Therefore,

$$\begin{aligned} LC_k(Y) &= \min_{W_H(E^N)} LC(Y + E) \\ &= N - \max_{W(e^N(x)) \leq k} z. \end{aligned} \quad (6)$$

From (4) and (6), we see that

$$LC_k(Y) < LC(Y)$$

if and only if there exists $e^N(x)$ such that $z > s$.

Lemma 4.1: $LC(Y + E) < LC(Y)$ if and only if

$$e^N(x) = (1-x)^s e_1(x) \quad (7)$$

for some $e_1(x)$ such that $e_1(1) = 1 (\neq 0)$.

Proof: We prove that $z > s$ if and only if (7) holds, where s and z are defined by (3) and (5), respectively. Suppose that

$$e^N(x) = (1-x)^c e_1(x)$$

for some c , where $e_1(1) = 1 (\neq 0)$. If $c > s$, then

$$y^N(x) + e^N(x) = (1-x)^s (y_1(x) + (1-x)^{c-s} e_1(x)).$$

Therefore,

$$\gcd((1-x)^{2^n}, y^N(x) + e^N(x)) = (1-x)^s$$

and $z = s$. Similarly, if $c < s$, then we have $z = c < s$. Finally, suppose that $c = s$. Then

$$y^N(x) + e^N(x) = (1-x)^s(y_1(x) + e_1(x)).$$

Since $f(1) + e_1(1) = 1 + 1 = 0$ we have

$$y_1(x) + e_1(x) = (1-x)g(x)$$

for some $g(x)$. Therefore,

$$y^N(x) + e^N(x) = (1-x)^{s+1}g(x).$$

Hence

$$z \geq s + 1 > s.$$

Q.E.D.

Theorem 4.1: If $N = 2^n$, then

$$\text{minerror}(Y) = 2^{W_{\text{H}}(N - \text{LC}(Y))}.$$

Proof: First, from Lemma 4.1, we have

$$\text{minerror}(Y) = \min_{e_1(1)=1} W((1-x)^s e_1(x)).$$

Next, from Propositions 3.2 and 3.1, if $e_1(1) = 1$, then

$$W((1-x)^s e_1(x)) \geq W((1-x)^s) = 2^{W(s)}. \quad (8)$$

Finally,

$$s = N - \text{LC}(Y)$$

from (4).

Q.E.D.

Next we say that E^N (resp., $e^N(x)$) is a critical error vector (resp., polynomial) for (Y) if

$$W_{\text{H}}(E^N) = \text{minerror}(Y) \quad (9)$$

and

$$\text{LC}(Y + E) < \text{LC}(Y). \quad (10)$$

Note that for $k = \text{minerror}(Y)$

$$\text{LC}_k(Y) = \min \text{LC}(Y + E)$$

where the minimum is taken over the critical error vectors E^N . From the proof of Theorem 4.1, we have the following corollary.

Corollary 4.1: Suppose that $N = 2^n$. Let $s = N - \text{LC}(Y)$. Then $e^N(x)$ is a critical error polynomial for (Y) if and only if $e^N(x) = (1-x)^s e_1(x)$ for some $e_1(x)$ such that $e_1(1) = 1$ and

$$W((1-x)^s e_1(x)) = W((1-x)^s) = 2^{W_{\text{H}}(s)}.$$

We show an example. Let

$$Y^{16} = (1101010011100111).$$

Then

$$y^{16}(x) = 1 + x + x^3 + x^5 + x^8 + x^9 + x^{10} + x^{13} + x^{14} + x^{15}$$

$$\text{gcd}(1 - x^{16}, y^{16}(x)) = (1 - x)^5 \quad (s = 5)$$

$$\text{LC}(Y) = 16 - 5 = 11.$$

Now from Theorem 4.1, we see that

$$\text{minerror}(Y) = 2^{W_{\text{H}}(16-11)} = 2^{W_{\text{H}}(5)} = 2^2 = 4.$$

This means that

$$11 = \text{LC}(Y) = \text{LC}_1(Y) = \text{LC}_2(Y) = \text{LC}_3(Y) > \text{LC}_4(Y).$$

Further, $e^N(x)$ is a critical error polynomial for (Y) if and only if

$$e^N(x) = (1-x)^5 e_1(x)$$

for some $e_1(x)$ such that $e_1(1) = 1$ and

$$W((1-x)^5 e_1(x)) = W((1-x)^5) = 2^{W_{\text{H}}(5)} = 4.$$

V. $\text{LC}_k(Y)$ FOR $k = \text{minerror}(Y)$

Let $k = \text{minerror}(Y)$. In this section, we show a necessary and sufficient condition for $\text{LC}_k(Y) = 0, 1, 2$ and $\text{LC}_k(Y) > 2$ in terms of $W(y^N(x))$ for binary sequences (Y) with period $N = 2^n$. We also present an upper bound on $\text{LC}_k(Y)$.

Let $s = N - \text{LC}(Y)$. Remember that

$$y^N(x) = (1-x)^s y_1(x) \quad (11)$$

for some $y_1(x)$ such that $y_1(1) = 1$ from (3).

Lemma 5.1: Suppose that $N = 2^n$. Let

$$a^N(x) = (1 + x^{2^l} + x^{2 \cdot 2^l} + \cdots + x^{2^n - 2^l}) a_1(x)$$

where $\deg a_1(x) < 2^l$. If

$$N - 2^l > N - \text{LC}(Y) (= s)$$

then

$$W(y^N(x) + a^N(x)) \geq 2^{W_{\text{H}}(s)}.$$

Further, if the above equality holds, then

$$\text{LC}_k(Y) \leq \text{LC}(a^N(x)).$$

(Note that $a^N(x)$ represents a repeated sequence of $a_1(x)$ with period 2^l .)

Proof:

$$(1 - x^{2^l})(1 + x^{2^l} + \cdots + x^{2^n - 2^l}) = 1 - x^{2^n}.$$

Therefore,

$$1 + x^{2^l} + \cdots + x^{2^n - 2^l} = \frac{(1-x)^{2^n}}{(1-x)^{2^l}} = (1-x)^{2^n - 2^l}.$$

Let

$$f(x) \triangleq y^N(x) + a^N(x).$$

Then

$$\begin{aligned} f(x) &= (1-x)^s y_1(x) + (1-x)^{2^n - 2^l} a_1(x) \\ &= (1-x)^s \{y_1(x) + (1-x)^{2^n - 2^l - s} a_1(x)\}. \end{aligned}$$

Note that $2^n - 2^l - s > 0$ from our assumption. Hence

$$y_1(1) + (1-1)^{2^n - 2^l - s} a_1(1) = y_1(1) = 1.$$

Therefore, $W(f(x)) \geq 2^{W_{\text{H}}(s)}$ from Proposition 3.2.

Next, suppose that $W(f(x)) = 2^{W_{\text{H}}(s)}$. Then $f(x)$ is a critical error polynomial from Corollary 4.1. Therefore,

$$\text{LC}_k(Y) \leq \text{LC}(y^N(x) + f(x)) = \text{LC}(a^N(x)). \quad \text{Q.E.D.}$$

It is clear that $\text{LC}_k(Y) = 0$ if $\text{LC}(Y) = 1$. Therefore, we consider $\text{LC}_k(Y)$ for $\text{LC}(Y) \geq 2$.

Theorem 5.1: If $\text{LC}(Y) \geq 2$, then

$$2^{W_{\text{H}}(s)} \leq W(y^N(x)) \leq N - 2^{W_{\text{H}}(s)}. \quad (12)$$

Further

- (1) $\text{LC}_k(Y) = 0$ if and only if $W(y^N(x)) = 2^{W_{\text{H}}(s)}$.
- (2) $\text{LC}_k(Y) = 1$ if and only if $W(y^N(x)) = N - 2^{W_{\text{H}}(s)} > 2^{W_{\text{H}}(s)}$.
- (3) $\text{LC}_k(Y) \geq 2$ if and only if $\text{LC}(Y) \geq 3$ and

$$2^{W_{\text{H}}(s)} < W(y^N(x)) < N - 2^{W_{\text{H}}(s)}.$$

$\text{LC}_k(Y) = 2$ if and only if the following condition is satisfied as well. Let

$$b(x) \triangleq 1 + x^2 + x^4 + \cdots + x^{N-2}.$$

Then

$$W(y^N(x) + b(x)) = 2^{W_{\text{H}}(s)} \quad (13)$$

or

$$W(y^N(x) + x \cdot b(x)) = 2^{W_{\text{H}}(s)}. \quad (14)$$

Proof: Equation (12) will be proved in the proof of (1) and (2).

(1) In lemma 5.1, let $a_1(x) = 0$ and $l = 0$. Then we have

$$W(y^N(x)) \geq 2^{W_{\text{H}}(s)}.$$

If $W(y^N(x)) = 2^{W_{\text{H}}(s)}$ then

$$0 \leq \text{LC}_k(Y) \leq \text{LC}(a^N(x)) = 0$$

from Lemma 5.1. Hence, $\text{LC}_k(Y) = 0$.

Conversely, suppose that $\text{LC}_k(Y) = 0$. Then there exists a critical error polynomial $e^N(x)$ such that

$$y^N(x) + e^N(x) = 0.$$

Therefore,

$$W(y^N(x)) = W(e^N(x)) = 2^{W_{\text{H}}(s)}$$

from Corollary 4.1.

(2) In Lemma 5.1, let $a_1(x) = 1$ and $l = 0$. Then

$$a^N(x) = 1 + x + x^2 + \cdots + x^{N-1}.$$

Hence we have

$$N - W(y^N(x)) = W(y^N(x) + a^N(x)) \geq 2^{W_{\text{H}}(s)}.$$

Therefore,

$$W(y^N(x)) \leq N - 2^{W_{\text{H}}(s)}.$$

If the above equality holds, then

$$\text{LC}_k(Y) \leq \text{LC}(a^N(x)) = 1$$

from Lemmas 5.1 and 2.1. Suppose that $W(y^N(x)) > 2^{W_{\text{H}}(s)}$ as well. Then $\text{LC}_k(Y) \geq 1$ from (1). Therefore,

$$\text{LC}_k(Y) = 1$$

if $W(y^N(x)) = N - 2^{W_{\text{H}}(s)} > 2^{W_{\text{H}}(s)}$.

Conversely, suppose that $\text{LC}_k(Y) = 1$. Then from Lemma 2.1, there exists a critical error polynomial $e^N(x)$ such that

$$y^N(x) + e^N(x) = 1 + x + x^2 + \cdots + x^{N-1} (= a^N(x)).$$

Therefore,

$$\begin{aligned} N - W(y^N(x)) &= W(y^N(x) + a^N(x)) \\ &= W(e^N(x)) = 2^{W_{\text{H}}(s)} \end{aligned}$$

from Corollary 4.1.

(3) From (1) and (2), $\text{LC}_k(Y) \geq 2$ if

$$2^{W_{\text{H}}(s)} < W(Y^N(x)) < N - 2^{W_{\text{H}}(s)}. \quad (15)$$

Next in Lemma 5.1, let $l = 1$ and $a_1(x) = 1$ (that is, $a^N(x) = b(x)$) or let $l = 1$ and $a_1(x) = x$ (that is, $a^N(x) = x \cdot b(x)$). In any case, from Lemma 2.1, $\text{LC}(a^N(x)) = 2$. Therefore, we have

$$\text{LC}_k(Y) \leq \text{LC}(a^N(x)) = 2$$

if (13) or (14) holds. Therefore, $\text{LC}_k(Y) = 2$ if (13)–(15) hold.

Conversely, suppose that $\text{LC}_k(Y) = 2$. Then from Lemma 2.1, there exists a critical error polynomial $e^N(x)$ such that

$$y^N(x) + e^N(x) = b(x) \text{ or } x \cdot b(x).$$

Therefore,

$$W(y^N(x) + b(x)) = W(e^N(x)) = 2^{W_{\text{H}}(s)}$$

or

$$W(y^N(x) + x \cdot b(x)) = W(e^N(x)) = 2^{W_{\text{H}}(s)}$$

from Corollary 4.1.

Q.E.D.

Next we show an upper bound on $\text{LC}_k(Y)$. From Corollary 4.1, $(1-x)^s x^i$ is a critical error polynomial for $0 \leq i < N - s = \text{LC}(Y)$. In this case,

$$y^N(x) + e^N(x) = (1-x)^s (y_1(x) + x^i).$$

Now let

$$L_i \triangleq \text{gcd}(1 - x^N, y_1(x) + x^i).$$

That is,

$$y_1(x) + x^i = (1-x)^{L_i} g_i(x)$$

for some $g_i(x)$ such that $g_i(1) = 1$.

Theorem 5.2:

$$\text{LC}_k(Y) \leq \text{LC}(Y) - \max_{0 \leq i < \text{LC}(Y)} L_i.$$

Proof: From Corollary 4.1, $(1-x)^s x^i$ is a critical error polynomial for (Y) . Then from the definition of k -error linear complexity and Proposition 2.1

$$\begin{aligned} \text{LC}_k(Y) &\leq N - \max_{0 \leq i < \text{LC}(Y)} \text{gcd}(1 - x^N, y^N(x) + (1-x)^s x^i) \\ &= N - \max_{0 \leq i < \text{LC}(Y)} \text{gcd}(1 - x^N, (1-x)^s (y_1(x) + x^i)) \\ &= N - \max_{0 \leq i < \text{LC}(Y)} \text{gcd}(1 - x^N, (1-x)^{s+L} g_i(x)) \\ &= N - (s + \max_{0 \leq i < \text{LC}(Y)} L_i) \\ &= \text{LC}(Y) - \max_{0 \leq i < \text{LC}(Y)} L_i. \end{aligned} \quad \text{Q.E.D.}$$

VI. EXTENSION TO NONBINARY SEQUENCES

In this section, we generalize our results to nonbinary sequences (Y) over $\text{GF}(p^r)$ with period $N = p^n$.

For a polynomial $e(x)$ over $\text{GF}(p^r)$, let $W(e(x))$ denote the Hamming weight of $e(x)$, i.e., the number of nonzero coefficients. Massey, Costello, and Justesen extended Propositions 3.1 and 3.2 as follows [4].

Proposition 6.1 [4]: If $f(1) \neq 0$, then

$$W((1-x)^N f(x)) \geq W((1-x)^N).$$

Definition 6.1: For a nonnegative integer i which is written as

$$(i_{m-1}, \dots, i_1, i_0)$$

in the radix p form, define

$$\text{Prod}(i) \triangleq \prod_{j=0}^{m-1} (i_j + 1).$$

Proposition 6.2 [4]:

$$W((1-x)^N) = \text{Prod}(N).$$

Then Theorem 4.1 is generalized as follows.

Theorem 6.1: $\text{minerror}(Y) = \text{Prod}(N - \text{LC}(Y))$.

Next let $s = N - \text{LC}(Y)$. Then similarly to (3), $y^N(x)$ is written as

$$y^N(x) = (1-x)^s y_1(x)$$

for some $y_1(x)$ such that $y_1(1) \neq 1$.

Theorem 6.2: $e^N(x)$ is a critical error polynomial for (Y) if and only if $e^N(x) = (1-x)^s e_1(x)$ for some $e_1(x)$ such that $e_1(1) = -y_1(1)$ and

$$W((1-x)^s e_1(x)) = W((1-x)^s) = \text{Prod}(N).$$

Theorems 5.1 and 5.2 are generalized as follows.

Theorem 6.3: If $\text{LC}(Y) \geq 2$, then

$$W(y^N(x)) \geq \text{Prod}(N).$$

Further, $\text{LC}_k(Y) = 0$ if and only if $W(y^N(x)) = \text{Prod}(N)$.

Theorem 6.4: Let

$$L_i \triangleq \gcd(1 - x^N, y_1(x) - y_1(1)x^i).$$

Then

$$\text{LC}_k(Y) \leq \text{LC}(Y) - \max_{0 \leq i < \text{LC}(Y)} L_i.$$

VII. k -ERROR LINEAR COMPLEXITY OF BINARY m -SEQUENCES

In this section, we completely determine k -error linear complexities $\text{LC}_k(Y)$ of binary m -sequences (Y) with the linear complexity $\text{LC}(Y) = \text{prime}$.

Proposition 7.1 [2]: $x^{p^m} - x = \text{product of all monic polynomials, irreducible over } \text{GF}(p), \text{ whose degree divides } m.$

Corollary 7.1: Let p be a prime and let $N = 2^p - 1$. Then

$$x^N - 1 = (x-1) \prod_i f_i(x)$$

where $f_i(x)$ is an irreducible polynomial over $\text{GF}(2)$ such that $\deg f_i(x) = p$.

Theorem 7.1: Let (Y) be a binary m -sequence with $\text{LC}(Y) = p = \text{prime}$. Then

$$\text{LC}_k(Y) = \begin{cases} \text{LC}(Y) = p, & \text{for } 0 \leq k \leq 2^{p-1} - 2 \\ 1, & \text{for } k = 2^{p-1} - 1 \\ 0, & \text{for } 2^{p-1} \leq k \leq 2^p - 1. \end{cases}$$

Proof: The m -sequence (Y) has a period $N = 2^p - 1$. Let Y^N denote one period of (Y) and $W_H(Y^N)$ denote the Hamming weight of Y^N . Then it is known that [2]

$$W_H(Y^n) = 2^{p-1}.$$

Therefore,

$$\text{LC}_k(Y) = 1$$

if and only if

$$k = N - W_H(Y^n) = 2^{p-1} - 1 \quad (16)$$

from Lemma 2.1. This implies that

$$\text{LC}_k(Y) = 0, \quad \text{if } k \geq W_H(Y^N) = 2^{p-1}.$$

Next for $k \leq 2^{p-1} - 2$, suppose that $\text{LC}_k(Y)$ is given by $e^N(x)$. Then Corollary 7.1 implies that

$$(y^N(x) + e^N(x))/(1-x^N) = g(x)/f(x) \text{ or } 1/(1-x) \text{ or } 1$$

for some $g(x)$ and $f(x)$ such that $\deg g(x) < \deg f(x) = p$ because $\text{LC}_k(Y) \leq \text{LC}(Y) = p$. In other words,

$$\text{LC}_k(Y) = p \text{ or } 1 \text{ or } 0.$$

However, from (16), we see that $\text{LC}_k(Y) = p$ if $0 \leq k \leq 2^{p-1} - 2$. Q.E.D.

VIII. CONCLUDING REMARKS

Stamp and Martin showed an efficient algorithm which computes the k -error linear complexity $\text{LC}_k(Y)$ of binary sequences with period $N = 2^n$ for any k [6]. We can use this algorithm to obtain $\text{minerror}(Y)$, which has been introduced in this correspondence.

Our contribution is that we have shown an *explicit* expression of $\text{minerror}(Y)$. It has been given to sequences (Y) over $\text{GF}(p^r)$ with period $N = p^n$. We have also derived some bounds on $\text{LC}_k(Y)$ for $k = \text{minerror}(Y)$. Finally, we have completely determined $\text{LC}_k(Y)$ of binary m -sequences with $\text{LC}(Y) = \text{prime}$.

It will be a further work to find $\text{minerror}(Y)$ for arbitrary period N . It will also be interesting to find

- (1) tighter bounds on $\text{LC}_k(Y)$ and
- (2) the *second-minerror* (Y) , that is, the minimum value of k' such that

$$\text{LC}_{k'}(Y) < \text{LC}_k(Y)$$

where $k = \text{minerror}(Y)$.

REFERENCES

- [1] C. Ding, G. Xiao, and W. Shan, "The stability theory of stream ciphers," in *Springer Verlag, Lecture Notes in Computer Science*. New York: Springer-Verlag, 1991, p. 561.
- [2] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.
- [3] J. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122-127, Jan. 1969.
- [4] J. Massey, D. Costello, and J. Justesen, "Polynomial weights and code constructions," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 101-110, Jan. 1973.
- [5] R. A. Rueppel, *Analysis and Design of Stream Ciphers*. New York: Springer-Verlag, 1986.
- [6] M. Stamp and F. Martin, "An algorithm for the k -error linear complexity of binary sequences with period 2^n ," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1398-1401, July 1993.

About Priority Encoding Transmission

Stéphane Boucheron, *Member, IEEE*, and
 Mohammad Reza Salamatian, *Student Member, IEEE*

Abstract—Recently, Albanese *et al.* introduced priority encoding transmission (PET) for sending hierarchically organized messages over lossy packet-based computer networks [1]. In a PET system, each symbol in the message is assigned a priority which determines the minimal number of codeword symbols that is required to recover that symbol. This note revisits the PET approach using tools from network information theory. We first outline that priority encoding transmission is intimately related with the broadcast erasure channel with degraded message set. Using the information spectrum approach, we provide an informational characterization of the capacity region of general broadcast channels with degraded message set. We show that the PET inequality has an information-theoretical counterpart: The inequality defining the capacity region of the broadcast erasure channel with degraded message sets. Hence the PET approach which consists in time-sharing and interleaving classical erasure-resilient codes achieves the capacity region of this channel. Moreover, we show that the PET approach may achieve the sphere packing exponents. Finally, we observe that on some simple nonstationary broadcast channels, time-sharing may be outperformed. The impact of memory on the optimality of the PET approach remains elusive.

Index Terms—Broadcast channels, coding exponents, erasure-resilient codes, information spectrum, priority encoding transmission.

I. INTRODUCTION

The quality of packet voice and image on the Internet has been mediocre due, in part, to congestion-induced packet losses. From the end-user viewpoint, the Internet can actually be modeled as an erasure channel acting over the large input alphabet formed by IP packets. On an erasure channel, each input symbol is either faithfully transmitted or erased, independently from its value. The output alphabet contains the input alphabet plus a special symbol denoting erasure. Although retransmission upon request, automatic repeat request (ARQ) has traditionally been the way to turn computer networks into reliable channels, the delay requirement of multimedia applications eliminates the possibility of retransmission and renews the interest for forward error correcting (FEC) [2]. Potential users of FEC also have to take into account that standard multimedia compression techniques [4] introduce a hierarchical structure in the information source.

The priority encoding transmission (PET) approach has been motivated by the search for robust multicasting of digital video sequences conforming to the MPEG standard [1]. In a first approximation, the different sorts of frames constituting a GOP (group of pictures in the MPEG methodology [4]) are assumed to form independent sources (this would be true if compression were perfect). When using PET, the source information is protected in such a way that even receivers undergoing high loss rates can reconstruct essential parts of the source flow (for example, I-frames), while receivers undergoing lower loss rates can reconstruct most of the flow.

This note outlines the connection between the pragmatically motivated PET approach and the *broadcast channel with degraded*

message set described in multiuser information theory [5]. In such a broadcast channel, one transmitter tries to send k independent messages m_0, \dots, m_{k-1} (corresponding to different priority levels) to k receivers (enjoying different reception conditions). The messages are multiplexed by a channel encoder into a sequence of input symbols x , $x = f(m_0 \dots m_{k-1})$. The i th receiver ($0 \leq i < k$) gets a corrupted version y_i of the input x , and tries to reconstruct messages $m_0 \dots m_i$. At least from an intuitive viewpoint, the task faced by priority encoding transmission and coping with broadcast channels are similar. To elaborate further, let us recall more precisely the PET code description.

A PET code over alphabet \mathcal{X} , with message length m , code length n , and nondecreasing priority function β is a pair of mappings f (encoder) from \mathcal{X}^m to \mathcal{X}^n and ϕ (decoder) from $(\mathcal{X} \cup \{\mathbf{e}\})^n$ to $\mathcal{X}^m \cup \{\text{reject}\}$ such that if w' is obtained from $f(w)$ by erasing at most $n - \beta(i)$ symbols, then $\phi(w')$ coincides with w at least on the first i symbols. The values in the range of a priority function are called the priority levels. In the sequel, the i th level of the priority function is denoted by β^i for $0 \leq i < k$.

The rate (resp., normalized rate) of a code represents the fraction of information bits (resp., symbols) per symbol. Formally, the rate \mathbf{R} (resp., normalized rate $\tilde{\mathbf{R}}$) of a code of length n , with M codewords over alphabet \mathcal{X} is $\frac{1}{n} \log M$ (resp., $\frac{1}{n} \log_{|\mathcal{X}|} M$), all logarithms being given in base 2. Following [1], a tuple of normalized rates $(\tilde{\mathbf{R}}_0, \dots, \tilde{\mathbf{R}}_{k-1})$ is achieved by a PET system if and only if there are exactly $n\tilde{\mathbf{R}}_i$ symbols from the message that are protected at level i . Though PET was not presented in a Shannon-theoretical perspective, the following relevant inequality is proved in [1, Theorems 3.3, 5.4]. For any PET system

$$\sum_{0=i}^{k-1} \frac{n\tilde{\mathbf{R}}_i}{\beta^i} \leq 1. \quad (1)$$

Inequality (1) puts combinatorial limits on finite sets of fixed-length words. When restricted, for example, to the case of one priority level, it coincides with the Singleton bound: $\tilde{\mathbf{R}}_0 \leq \beta^0/n$. Note that the latter bound cannot be achieved by codes of arbitrary length (cf., for example, [1, Theorem 6.1]). On a given alphabet, arbitrarily long codes satisfying the PET inequality do not exist and thus cannot be used to achieve arbitrarily reliable transmission on lossy broadcast channels.

Moreover, the PET approach advocates a very simple method to design broadcast codes. Packets are divided into slots, each slot size defining an alphabet. For each priority level, apply a good point-to-point erasure-resilient code over the small alphabet corresponding to the slot size and then interleave the resulting codewords in the packets. This is a version of the engineering approach called time-sharing in [6]. This approach is known to be nonoptimal on many broadcast channels. Assessing the PET approach from an information-theoretical viewpoint amounts to first checking whether the capacity region of broadcast erasure channels with degraded message sets can be exhausted using priority encoding transmission. In the affirmative, the second natural question is: Does error probability decline as fast as possible when PET is used?

We first use the *information spectrum approach* [7] to provide an informational characterization of the capacity region of general broadcast channels with degraded message set (DMS) in Section II. This characterization illustrates the relevance to general broadcast channels with memory of superposition codes as introduced in [6]. Then we show in Section III that the PET approach achieves the capacity region of memoryless broadcast erasure channels with DMS and, moreover, that it may achieve the sphere-packing exponent of this channel. The last sec-

Manuscript received October 21, 1997; revised September 22, 1999. This research was supported in part by CNET under Grant 96 1-B 212, and by Esprit Working Group RAND II.

The authors are with the LRI, CNRS UMR 8623, Bât 490, Université Paris-Sud, 91405 Orsay, France (e-mail: {bouchero; salamat}@lri.fr).

Communicated by F. R. Kschischang, Associate Editor for Coding Theory. Publisher Item Identifier S 0018-9448(00)01689-8.

tion shows that on some simple nonstationary broadcast erasure channels, time-sharing may be outperformed.

II. BROADCAST CHANNEL WITH DEGRADED MESSAGE SET

A. General Definitions

In the sequel \mathbf{X} , \mathbf{Y} denote families of input processes and their corresponding output through a channel \mathbf{W} . For each n , X^n denotes a random variable over \mathcal{X}^n and Y^n is distributed over \mathcal{Y}^n according to $W^n(\cdot|X^n)$. $X^n(i)$, $Y^n(i)$ denotes the i th element of X^n , Y^n .

The entropy of a random variable X , $H(X)$ is defined by

$$H(X) = - \sum_i \Pr\{X = i\} \log \Pr\{X = i\}.$$

The mutual information between X and Y is defined by

$$I(X; Y) \triangleq - \sum_{x, y} \Pr\{X = x, Y = y\} \times \log \frac{\Pr\{X = x, Y = y\}}{\Pr\{X = x\} \Pr\{Y = y\}}.$$

The *information spectrum approach* has been proposed recently to handle systems with general dependencies [7]. It relies on the notion of *liminf in probability*: $p - \liminf_{n \rightarrow \infty} X_n$ is defined as the supremum of α such that $\limsup_{n \rightarrow \infty} \Pr\{X_n < \alpha\} = 0$. Given two random processes (\mathbf{X}, \mathbf{Y}) , the mutual *information spectrum-inf* is defined by

$$\underline{I}(\mathbf{X}; \mathbf{Y}) \triangleq p - \liminf_{n \rightarrow \infty} \frac{1}{n} \times \log \frac{\Pr\{X^n = x^n, Y^n = y^n\}}{\Pr\{X^n = x^n\} \times \Pr\{Y^n = y^n\}}.$$

B. Broadcast Channels

To avoid confusion, sources (corresponding to requested priority levels) and component channels (corresponding to different levels of transmission reliability) will be indexed using boldface indices $\mathbf{i}, \dots, \mathbf{k}$.

Definition 1: A k -ary broadcast channel \mathbf{W} consists of a sequence of joint probability transitions $W^n(Y_0^n \cdots Y_{k-1}^n | X^n)$ from \mathcal{X}^n toward $\mathcal{Y}^{n \times k}$. The marginal probability transitions $W_i^n(Y_i^n | X^n)$ are called the component channels.

A *degraded message set* can be transmitted at rate $(\mathbf{R}_0, \dots, \mathbf{R}_{k-1})$ over \mathbf{W} with error probability ϵ if and only if there exists a family of (broadcast) codes

$$f^n(m_0, \dots, m_{k-1}) \mapsto x$$

and

$$\phi_i^n(y_i) = (\hat{m}_0, \dots, \hat{m}_i)$$

such that for almost all block length n , $\liminf (\log |M_i|/n) \geq \mathbf{R}_i$ and for all $\mathbf{i} < \mathbf{k}$, the error probability experienced by the i th receiver satisfies

$$\limsup \mathbf{e}_i(f^n, \phi_i^n) \triangleq \max_{m_0, \dots, m_{k-1}} W[\phi_i^n(y_i) \neq (m_0 \cdots m_i) | f^n(m_0, \dots, m_{k-1})] \leq \epsilon.$$

The rate-tuple (\mathbf{R}_i) is then said to be ϵ -achievable over \mathbf{W} . A broadcast code with block length n , rates (\mathbf{R}_i) , and error rate smaller than ϵ for all receivers on channel \mathbf{W} is called an $(n, \mathbf{R}_0 \cdots \mathbf{R}_{k-1}, \epsilon)$ -code over \mathbf{W} . A tuple of rates is achievable if it is ϵ -achievable for all $\epsilon > 0$.

Remarks:

- 1) From the definition, it is immediate that the set of achievable (resp., ϵ -achievable) rates is closed.
- 2) For broadcast channels, achievable rates do not depend on whether we consider average (over codewords) or worst case error probability (cf. [5], where no assumption on channel

memory is made). In the sequel, we will adopt the most convenient viewpoint depending on the situation.

The memoryless broadcast channel with degraded message set has received a single letter characterization [5, Theorems III.4.1 and 3], [8]. The information spectrum approach allows to give an informational (though not computational) characterization of operationally defined achievable rates over general broadcast channels.

Let us now define the class of families of input processes \mathcal{S} as $\mathbf{U}_0, \dots, \mathbf{U}_{k-2}, \mathbf{U}_{k-1}$ with $\mathbf{U}_{k-1} \triangleq \mathbf{X}$ and corresponding output families $\mathbf{Y}_{i=0}^{k-1}$ as follows. For every block length n , U_i^n is independent from $U_{i+2}^n \cdots X^n, Y^n$ conditionally on U_{i+1}^n , and Y_i^n is distributed according to $W_i^n(\cdot|X^n)$. Let us insist on the fact that the variables U_i^n ($i < k-1$) do not necessarily live in an n -dimensional product space. Such families of input processes are general since no consistency constraint between processes with different indices n and m is imposed. The structure of input families is inspired from the superposition codes construction [6].

$\mathbf{R}_W(\mathbf{U}_{i < k})$ is the set of tuple of rates $(\mathbf{R}_0 \cdots \mathbf{R}_{k-1})$ satisfying

$$\begin{aligned} 0 \leq \mathbf{R}_i &\leq \underline{I}(\mathbf{U}_i; \mathbf{Y}_i | \mathbf{U}_{i-1}, \dots, 0), & \text{for } \mathbf{i} < \mathbf{k} \\ 0 \leq \sum_{j \leq i} \mathbf{R}_j &\leq \underline{I}(\mathbf{U}_i; \mathbf{Y}_i), & \text{for } \mathbf{i} < \mathbf{k}. \end{aligned}$$

Proposition 1: The set of achievable rates $(\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_{k-1})$ over a broadcast channel \mathbf{W} with degraded message set is the closure of

$$\bigcup_{\mathbf{U}_0, \dots, \mathbf{U}_{k-1} \in \mathcal{S}} \mathbf{R}_W(\mathbf{U}_{i < k}).$$

Proposition 1 can be completed by the determination of the region of ϵ -achievable rate tuples and by the characterization of those broadcast channels that have the strong converse property.

$\mathbf{R}_W(\epsilon, \mathbf{U}_{i < k})$ is the set of tuple of rates $(\mathbf{R}_0 \cdots \mathbf{R}_{k-1})$ satisfying

$$\limsup_{n \rightarrow \infty} \Pr \left\{ \forall_i \frac{1}{n} \log \frac{\Pr\{Y_i^n, U_i^n | U_0^n, \dots, U_{i-1}^n\}}{\Pr\{Y_i^n | U_0^n, \dots, U_{i-1}^n\}} \leq \mathbf{R}_i, \right. \\ \left. \forall_i \frac{1}{n} \log \frac{\Pr\{Y_i^n, U_i^n\}}{\Pr\{Y_i^n\}} \leq \sum_{j \leq i} \mathbf{R}_j \right\} \leq \epsilon. \quad (2)$$

This set is closed.

Proposition 2: The set of ϵ -achievable rates $(\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_{k-1})$ over a broadcast channel \mathbf{W} with degraded message set is the closure of

$$\bigcup_{\mathbf{U}_0, \dots, \mathbf{U}_{k-1} \in \mathcal{S}} \mathbf{R}_W(\epsilon, \mathbf{U}_{i < k}).$$

As $\bigcap_{\epsilon > 0} \mathbf{R}_W(\epsilon, \mathbf{U}_{i < k})$ equals $\mathbf{R}_W(\mathbf{U}_{i < k})$, this could serve as a definition of $\mathbf{R}_W(\mathbf{U}_{i < k})$. Proposition 2 has an intuitive interpretation: for a tuple of rates to be ϵ -achievable it is essential that the probability that the amount of transmitted information is less than the corresponding rate, is smaller than ϵ . The statement of the two technical Lemmas 1 and 2 that are used to prove Propositions 1 and 2 reveals that this intuition is quantitative.

A broadcast channel has the *strong converse property* if for any tuple of rates $(\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_{k-1})$, either $(\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_{k-1})$ is ϵ -achievable for all ϵ or any sequence of codes with rates $(\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_{k-1})$ has error probability converging toward 1.

Let $\mathbf{R}_W^*(\mathbf{U}_{i < k})$ be the set of rate-tuples defined by

$$\liminf_{n \rightarrow \infty} \Pr \left\{ \forall i \frac{1}{n} \log \frac{\Pr \{Y_i^n, U_i^n | U_0^n, \dots, U_{i-1}^n\}}{\Pr \{Y_i^n | U_0^n, \dots, U_{i-1}^n\}} \leq R_i \right. \\ \left. \vee \frac{1}{n} \log \frac{\Pr \{Y_i^n, U_i^n\}}{\Pr \{Y_i^n\}} \leq \sum_{j \leq i} R_j \right\} < 1. \quad (3)$$

The broadcast channels that have the strong converse properties may be characterized by the following condition.

Proposition 3: A broadcast channel with degraded message set has the strong converse property if and only if

$$\text{closure}[\cup_{\mathbf{U}_i} \mathbf{R}_W^*(\mathbf{U}_{i < k})] = \text{closure}[\cup_{\mathbf{U}_i} \mathbf{R}_W(\mathbf{U}_{i < k})].$$

The proof of Propositions 1, 2, and 3 is an exercise in *information spectrum calculus*. It relies on an *information spectrum* version of Feinstein's lemma (for direct parts) and of its dual (for converse parts). To alleviate notations, we assume $k = 2$.

Lemma 1: Let \mathbf{U} and \mathbf{X} be any input sequence, then for any positive integer $(\mathbf{R}_0, \mathbf{R}_1)$, for any positive γ , there exists a broadcast $(n, \mathbf{R}_0, \mathbf{R}_1, \mathbf{e}_n)$ -code satisfying

$$\mathbf{e}_n \leq \Pr \left\{ \frac{1}{n} \log \frac{\sum_{x^n} \Pr_{X^n | U^n}(x^n | u^n) W_0^n(y_0^n | x^n)}{\Pr_{Y_0^n} \{y_0^n\}} \leq \mathbf{R}_0 + \gamma \right. \\ \left. \text{or } \frac{1}{n} \log \frac{W_1^n(y_1^n | x^n)}{\Pr_{Y_1^n | U^n} \{y_1^n | u^n\}} \leq \mathbf{R}_1 + \gamma \right. \\ \left. \text{or } \frac{1}{n} \log \frac{W_1^n(y_1^n | x^n)}{\Pr_{Y_1^n} \{y_1^n\}} \leq \mathbf{R}_0 + \mathbf{R}_1 + \gamma \right\} + 3e^{-n\gamma}.$$

Lemma 2: For every n , any $(n, \mathbf{R}_0, \mathbf{R}_1, \mathbf{e}_n)$ broadcast code satisfies

$$\mathbf{e}_n \geq \Pr \left\{ \frac{1}{n} \log \frac{\sum_{x^n} \Pr_{X^n | U^n}(x^n | u^n) W_0^n(y_0^n | x^n)}{\Pr_{Y_0^n} \{y_0^n\}} \leq \mathbf{R}_0 - \gamma \right. \\ \left. \text{or } \frac{1}{n} \log \frac{W_1^n(y_1^n | x^n)}{\Pr_{Y_1^n | U^n} \{y_1^n | u^n\}} \leq \mathbf{R}_1 - \gamma \right. \\ \left. \text{or } \frac{1}{n} \log \frac{W_1^n(y_1^n | x^n)}{\Pr_{Y_1^n} \{y_1^n\}} \leq \mathbf{R}_0 + \mathbf{R}_1 - \gamma \right\} - 3e^{-n\gamma}$$

where \mathbf{U}^n is uniformly distributed on a set of $M_0 = 2^{nR_0}$ disjoint sets (clouds) of codewords and $\Pr_{X^n | U^n}$ places probability 2^{-nR_1} over each codeword in a cloud.

A sketch of the proof of those two lemmas is given in the Appendix.

Remarks on the Proof of Propositions 1 and 2: The proof of the direct part is an application of Lemma 1 and of the definition of the mutual information spectrum-inf, as in [9, pp. 2782–2784]. The proof of the converse part is an application of Lemma 2 and of the definition of the mutual information spectrum-inf, as in [9, p. 2783–2784]. \square

Remarks on the Proof of Proposition 3: As

$$\text{closure} \cup_{\mathbf{U}_{i < k}} \mathbf{R}_W(\mathbf{U}_{i < k}) \subseteq \text{closure} \cup_{\mathbf{U}_{i < k}} \mathbf{R}_W^*(\mathbf{U}_{i < k})$$

always holds, the proof amounts to showing that the strong converse property is equivalent to

$$\text{closure} \cup_{\mathbf{U}_{i < k}} \mathbf{R}_W^*(\mathbf{U}_{i < k}) \subseteq \text{closure} \cup_{\mathbf{U}_{i < k}} \mathbf{R}_W(\mathbf{U}_{i < k}).$$

Proving that the strong converse property implies the inclusion is based on Lemma 1 and simple closure properties. Proving that the proper inclusion implies the strong converse property is based on Lemma 2 and again on simple closure properties. Indeed, if the last inclusion holds, the limiting probability that the amount of transmitted information is below a certain value is either 1 or 0. \square

Thus even though the superposition codes idea was first motivated by memoryless broadcast channels, the information spectrum approach shows that it remains fully relevant in face of memory. Indeed, for the broadcast channel, the characterization of the capacity region described in Lemma 1 is closer to the memoryless characterization than in the case of the multiple-access channel (cf. [9]). And it is natural to ask whether something is lost when using the PET method. In the next section, we show that almost nothing is lost in the memoryless case, while in the last section, a simple example shows it may be false in face of memory.

III. LOSSY BROADCAST CHANNELS

A. General Characterization

When specialized to lossy channels, the information spectrum approach provides a simple characterization of the capacity of single-user lossy channels. Let us first introduce another notation. The loss process (\mathbf{Z}) is defined as $Z^n(i) = 1$ if $Y^n(i)$ is a loss, 0 otherwise. The loss process completely defines the erasure channel. \mathbf{Z} is assumed to be independent from channel inputs. Let $X^n(z^n)$ denote the subsequence of random variables $X^n(i)$, such that $z^n(i) = 0$.

Proposition 4: The capacity of the erasure channel defined by the loss process \mathbf{Z} is

$$\hat{\mathbf{C}} = 1 - \text{p-lim sup} \frac{1}{n} \sum_{i \leq n} Z^n(i).$$

Remark: An erasure channel has the strong converse property if and only if $\frac{1}{n} \sum_{i \leq n} Z^n(i)$ converges in probability toward a fixed value. For stationary ergodic channels, the capacity only depends on the stationary loss probability, but dependence may dramatically change the reliability function of the channel. Let

$$\underline{H}(\mathbf{X}) \triangleq \text{p-lim inf} \frac{1}{n} \log \Pr \{X^n\}.$$

Proposition 1 can be specialized to lossy broadcast channels. $\mathbf{R}_W(\mathbf{U}_{i < k})$ is now the set of tuple of rates $(\mathbf{R}_0 \dots \mathbf{R}_{k-1})$ satisfying

$$0 \leq \mathbf{R}_i \leq \underline{I}(\mathbf{U}_i; \mathbf{X}(\mathbf{Z}_i) | \mathbf{U}_{i-1}, \dots, 0) \\ 0 \leq \mathbf{R}_{k-1} \leq \underline{H}(\mathbf{X}(\mathbf{Z}_{k-1}) | \mathbf{U}_{k-2}, \dots, 0) \\ 0 \leq \sum_{j \leq i} \mathbf{R}_j \leq \underline{I}(\mathbf{U}_i; \mathbf{X}(\mathbf{Z}_i)), \quad \text{for } i < k-1 \\ 0 \leq \sum_{i < k} \mathbf{R}_i \leq \underline{H}(\mathbf{X}(\mathbf{Z}_{k-1})). \quad (4)$$

Remark: If a broadcast erasure channel has exchangeable component loss processes, it is said to be exchangeable. Such broadcast channels provide simple examples of stationary broadcast channels failing to have to strong converse property.

B. Memoryless Broadcast Erasure Channels

1) *Capacity Region:* The structure of memoryless broadcast erasure channels is almost captured by Han's inequalities [10] or the following theorem due to Shearer [11]

Theorem 1: Let X^n be a collection of n random variables and Z^n be a collection of n Boolean random variables, such that for each i , $1 \leq i \leq n$, $\mathbb{E} Z_i = 1 - \hat{\mathbf{C}}$.

$$\mathbb{E} H(X^n(Z^n)) \geq \hat{\mathbf{C}} H(X^n). \quad (5)$$

This theorem was a key ingredient in the derivation of inequality (1), it enables here a direct derivation of the capacity region of the memoryless broadcast erasure channel without resorting to the single-letter characterization. It could as well have been used to derive the strong converse for memoryless lossy broadcast channels without resorting to the single-letter characterization, since it allows to use directly [5, Ch. III.3, Lemmas 4.2 and 4.3] in the proof of the strong converse for lossy broadcast channels [5, Theorem 4.3, Ch. III.4]. The following proposition was independently pointed out in [12] and in [13], its proof parallels the proof of [1, Inequality 1].

Proposition 5: A tuple of rates $(\mathbf{R}_{b,f_0}, \mathbf{R}_1, \dots, \mathbf{R}_{k-1})$ is achievable over a memoryless broadcast erasure channel with degraded message set if and only if

$$\sum_{i=1}^{k-1} \frac{\mathbf{R}_i}{\tilde{\mathbf{C}}_i} < 1. \quad (6)$$

Thus achievable rates may always be achieved by time-sharing.

Proof of Proposition 5: Since time-sharing is always feasible over stationary ergodic broadcast channels, only the converse part needs to be proved. We prove it when $k = 2$. Let $(\mathbf{R}_0, \mathbf{R}_1)$ be an achievable rate pair, then by [7, Proposition 1 and Theorem 8], there exists a family of input processes (\mathbf{U}, \mathbf{X}) such that for any $\gamma > 0$, there exists an n such that

$$\begin{aligned} \mathbf{R}_0 &\leq \frac{1}{n} (H[X^n(Z_0^n)] - H[X^n(Z_0^n)|U^n]) + \gamma \\ \mathbf{R}_1 &\leq \frac{1}{n} H[X^n(Z_1^n)|U^n] + \gamma. \end{aligned} \quad (7)$$

Now

$$H[X^n(Z_0^n)] \leq \tilde{\mathbf{C}}_0 \sum_{j \leq n} H[X^n(j)].$$

On the other hand, since losses on each channel are assumed to be independent, we may consider that the broadcast channel is stochastically or even physically degraded: any symbol lost on \mathbf{W}_1 is lost over \mathbf{W}_0 , other symbols are lost over \mathbf{W}_0 with probability $1 - (\tilde{\mathbf{C}}_0/\tilde{\mathbf{C}}_1)$. Then conditionally on Z_1^n , the sequence of random variables $X^n(Z_0^n)$ is a subsequence of $X^n(Z_1^n)$, a conditional use of Shearer's theorem implies

$$\mathbb{E} [H[X^n(Z_0^n)|U^n, Z_1^n = z_1^n]] \leq \frac{\tilde{\mathbf{C}}_0}{\tilde{\mathbf{C}}_1} H[X^n(z_1^n)|U^n].$$

Deconditioning with respect to z_1^n , we get

$$H([X^n(Z_0^n)|U^n]) \geq \frac{\tilde{\mathbf{C}}_0}{\tilde{\mathbf{C}}_1} H[X^n(Z_1^n)|U^n].$$

Substituting in (7) and adding the two inequalities, we get for arbitrary $\gamma > 0$

$$\frac{\mathbf{R}_0}{\tilde{\mathbf{C}}_0} + \frac{\mathbf{R}_1}{\tilde{\mathbf{C}}_1} \leq \log |\mathcal{X}| + 2\gamma. \quad \square$$

Remark: The essential ingredient in the proof is the conditional application of Shearer's theorem (or Han inequalities). It is still relevant to the analysis of exchangeable channels: The capacity region of exchangeable broadcast erasure channels is also described by (6), although those channels do not have the strong converse property.

2) *Error Exponents:* For good families of codes, error probabilities of memoryless broadcast channels are known to decline exponentially fast with block length n , provided the rate vector is in the capacity region of the channel [14], [15]. E is called an attainable error exponent if there exists a sequence of codes such that for any $\delta >$

0, for almost all block length n : $\max_i \mathbf{e}_i(n, \mathbf{R}_0, \dots, \mathbf{R}_{k-1}, W) < e^{-n(E-\delta)}$. The best error exponent is upper-bounded by the sphere-packing bound $E_{sp}(\cdot, W)$ and lower-bounded by the random coding bound $E_{rc}(\cdot, W)$, informational characterizations of those quantities are known [14], [15]. We are not aware of the existence of a general algorithm capable of computing the value of the exponents for broadcast channels but, fortunately, the broadcast erasure channel is simple enough so that those two quantities can be determined. A refined assessment of the PET approach consists in comparing the random coding exponent for PET codes with the sphere-packing exponent for memoryless broadcast erasure channels. Let us first recall the form of the sphere-packing and random-coding exponent for the single-user erasure channel. Gallager's approach [16] provides a closed form for single-user erasure channel exponents. Let

$$h(x, y) \triangleq x \log \frac{x}{y} + (1-x) \log \frac{1-x}{1-y}$$

denote the relative entropy between two Bernoulli random variables with parameters x and y , then

$$E_{sp}(\mathbf{R}, W) = h(\tilde{\mathbf{R}}, \tilde{\mathbf{C}}).$$

Let now the critical rate be

$$\tilde{\mathbf{R}}_{cr} \triangleq \tilde{\mathbf{C}} / (\tilde{\mathbf{C}} + |\mathcal{X}|(1 - \tilde{\mathbf{C}}))$$

then the random coding exponent satisfies

$$\begin{aligned} E_{rc}(\tilde{\mathbf{R}}, W) &= E_{sp}(\tilde{\mathbf{R}}, W), & \text{if } \tilde{\mathbf{R}}_{cr} \leq \tilde{\mathbf{R}} < \tilde{\mathbf{C}} \\ &= E_{sp}(\tilde{\mathbf{R}}_{cr}, W) + \tilde{\mathbf{R}}_{cr} - \mathbf{R}, & \text{if } \tilde{\mathbf{R}} \leq \tilde{\mathbf{R}}_{cr}. \end{aligned} \quad (8)$$

Remarks: For erasure channels, the random coding exponent only depends on $\tilde{\mathbf{R}}, \tilde{\mathbf{C}}$, and $|\mathcal{X}|$. It will be denoted $E_{rc}(\tilde{\mathbf{R}}, \tilde{\mathbf{C}}, \mathcal{X})$. For large alphabets, the sphere-packing exponent and the random-coding exponents coincide over a wide range of rates. Inspection of the derivation of the coding exponents of erasure channels (cf. [5]) shows that if we consider k erasure channels with capacities $\mathbf{C}_0 < \dots < \mathbf{C}_{k-1}$, it is always possible to find a family of codes that realizes simultaneously the k random coding exponents for the k channels.

The multiplexing strategy described here encompasses the partition refinement idea described in [1]. It is parameterized by the block length n , the bit length l of a packet ($|\mathcal{X}| = 2^l$), the number of priority levels k ($k \leq l$), the rates \mathbf{R}_i at which the different priority levels should be encoded for $0 \leq i < k$, and the capacities of the component channels $(W_i): (\mathbf{C}_i)$. We assume that

$$\sum_{i < k} (\mathbf{R}_i / \mathbf{C}_i) < 1.$$

In a PET code, for all $i < k$, l_i bits from each packet will be dedicated to encoding of message m_i at normalized rate $\tilde{\mathbf{R}}_i$. We will have to satisfy the constraints $\sum_i l_i = l$ and $l_i \tilde{\mathbf{R}}_i = l \tilde{\mathbf{R}}_i$.

The total bit length of a codeword will be nl , and nl_i bits will be dedicated to the encoding of the i th priority level. The PET code designer still has to trade l_i 's and $\tilde{\mathbf{R}}_i$'s. And one may wonder whether this is the best way to minimize the error probabilities experienced by the different receivers, i.e., whether spreading information from the i th priority level over the whole packet would not improve the error exponents. The following propositions constitute a partial answer to those questions. In the sequel, Λ denotes the set of vectors from \mathbb{R}^k such that $\sum_{i=0}^k \lambda_i = 1$ and $\lambda_i \geq 0$.

Proposition 6: For a broadcast erasure channel W , with capacities $(\mathbf{C}_i)_{i < k}$, and degraded message set, the sphere packing exponent is at most

$$\inf_{\lambda \in \Lambda} \max_i h \left(\frac{\tilde{\mathbf{R}}_i}{\lambda_i}, \tilde{\mathbf{C}}_i \right).$$

Proposition 7: For a broadcast erasure channel W , with degraded message set, the random coding exponent achieved by juxtaposing and interleaving single-user erasure codes is

$$\sup_{\lambda \in \Lambda} \min_{\mathbf{i}} E_{\text{rc}} \left(\frac{\tilde{\mathbf{R}}_{\mathbf{i}}}{\lambda_{\mathbf{i}}}, \tilde{\mathbf{C}}_{\mathbf{i}}, |\mathcal{X}|^{\lambda_{\mathbf{i}}} \right).$$

Remarks: The supremum in both propositions is attained for a tuple λ where all terms in the minimization (or maximization) are equal.

Notice that if all $\tilde{\mathbf{R}}_{\mathbf{i}}/\lambda_{\mathbf{i}}$ are larger than the critical rate of the erasure channel with normalized capacity $\tilde{\mathbf{C}}_{\mathbf{i}}$ and alphabet size $|\mathcal{X}|^{\lambda_{\mathbf{i}}}$, the upper bound stated in Proposition 7 is achieved. The combination of the two propositions seriously backs the following separation principle: First design good single-erasure codes with the required rates and alphabets, second, multiplex them in the simplest way using juxtaposition and interleaving.

Proof of Proposition 6: The arguments use the change of channel trick and a good guess of the twisted broadcast channel (cf. [5, Ch. II.5, p. 167] or [14]). We will consider the case of two sources that are to be transmitted at rates \mathbf{R}_0 and \mathbf{R}_1 over a broadcast erasure channel (W_0, W_1) . $(\mathbf{R}_0, \mathbf{R}_1)$ is assumed to be achievable. Now consider a twisted memoryless broadcast erasure channel $(W_0^{n'}, W_1^{n'})$ with pair of capacities

$$(\tilde{\mathbf{C}}_0', \tilde{\mathbf{C}}_1') = (1 - \delta)(\tilde{\mathbf{R}}_0/\lambda, \tilde{\mathbf{R}}_1/(1 - \lambda))$$

with $\lambda \in (0, 1)$ and $\delta > 0$. The pair $(\tilde{\mathbf{R}}_0, \tilde{\mathbf{R}}_1)$ is not achievable over this twisted channel. Hence as the strong converse property holds for memoryless broadcast channels [5], for any sequence of block broadcast codes, for n large enough $\max_{i \in \{0, 1\}} \mathbf{e}_i(f^n, \phi_i^n) > 1 - \delta/2$. This means that for some pair of messages (m_0, m_1) , the events $S_0 \triangleq \{y; \phi_0^n(y) \neq m_0\}$ and $S_1 \triangleq \{y; \phi_1^n(y) \neq \langle m_0, m_1 \rangle\}$ satisfy

$$\max_{\mathbf{i}} W_i^{n'}(S_{\mathbf{i}} | f^n(m_0, m_1)) > 1 - \delta/2.$$

Notice that on a fixed input, the distributions of outputs through channels W_i and W_i' have relative entropies $n \times h(\tilde{\mathbf{C}}_i, \tilde{\mathbf{C}}_i')$. Thus we have [5, Ch. II.5]

$$h[W_i^{n'}(S_{\mathbf{i}} | f^n(m_0, m_1)), W_i^n(S_{\mathbf{i}} | f^n(m_0, m_1))] \leq n \cdot h(\tilde{\mathbf{C}}_i', \tilde{\mathbf{C}}_i)$$

which implies:

$$\begin{aligned} \max_{\mathbf{i}} W_i^n(S_{\mathbf{i}} | f^n(m_0, m_1)) \\ \geq \min_{\mathbf{i}} \exp \left(- \frac{n h(\tilde{\mathbf{C}}_i', \tilde{\mathbf{C}}_i) - h(\delta/2)}{1 - \delta/2} \right). \end{aligned}$$

As δ may be arbitrarily small, by continuity of h , this in turn implies that the sphere-packing exponent for the memoryless broadcast erasure channel is smaller than $\max_{\mathbf{i}} (h(\tilde{\mathbf{R}}_{\mathbf{i}}/\lambda_{\mathbf{i}}, \tilde{\mathbf{C}}_{\mathbf{i}}))$ where $\lambda_0 = \lambda$ and $\lambda_1 = 1 - \lambda$. \square

Proof of Proposition 7: Let λ_j denote l_j/l for $j \in \{0, 1\}$. Let multiplexed single-user codes realize simultaneously the random-coding exponents for the relevant erasure channels, for all i

$$\begin{aligned} \mathbf{e}_i &\leq \sum_{j \leq i} e^{-n E_{\text{rc}}(\tilde{\mathbf{R}}_j', \tilde{\mathbf{C}}_j, |\mathcal{X}|^{\lambda_j})} \\ &\leq (i + 1) \max_{j \leq i} e^{-n E_{\text{rc}}(\tilde{\mathbf{R}}_j', \tilde{\mathbf{C}}_j, |\mathcal{X}|^{\lambda_j})}. \end{aligned} \quad (10)$$

Thus by monotonicity of the random-coding exponent with respect to capacity

$$\begin{aligned} \max_{\mathbf{i} < \mathbf{k}} \mathbf{e}_i &\leq \mathbf{k} \max_{j \leq \mathbf{i} < \mathbf{k}} e^{-n E_{\text{rc}}(\tilde{\mathbf{R}}_j', \tilde{\mathbf{C}}_j, |\mathcal{X}|^{\lambda_j})} \\ &\leq \mathbf{k} \max_{\mathbf{i} < \mathbf{k}} e^{-n E_{\text{rc}}(\tilde{\mathbf{R}}_{\mathbf{i}}', \tilde{\mathbf{C}}_{\mathbf{i}}, |\mathcal{X}|^{\lambda_{\mathbf{i}}})}. \end{aligned} \quad (11)$$

We get

$$E_{\text{rc}}(\mathbf{R}_0 \cdots \mathbf{R}_{k-1}, W) = \sup_{\lambda \in \Lambda} \min_{\mathbf{i}} E_{\text{rc}}(\tilde{\mathbf{R}}_{\mathbf{i}}/\lambda_{\mathbf{i}}, \tilde{\mathbf{C}}_{\mathbf{i}}, |\mathcal{X}|^{\lambda_{\mathbf{i}}}).$$

IV. GENERAL BROADCAST ERASURE CHANNELS

The engineering solution, i.e., time sharing and interleaving, provides a good method to design broadcast codes over memoryless broadcast erasure channels. It would be nice if it were also the case over erasure channels with memory. Unfortunately, it is possible to design asymptotically memoryless broadcast erasure channels for which time sharing does not exhaust the capacity region. Let (W_0, W_1) be defined in the following way. For every sequence of six symbols $x_{6i} \cdots x_{6i+5}$ for $i = 0 \cdots \infty$ either the first three symbols $x_{6i} \cdots x_{6i+2}$, or last three symbols $x_{6i+3} \cdots x_{6i+5}$ are erased by the powerful component channel W_1 , and four symbols, either $x_{6i} \cdots x_{6i+3}$, $x_{6i}, x_{6i+1}, x_{6i+4}, x_{6i+5}$, or $x_{6i+2} \cdots x_{6i+5}$ are erased. Blocks $x_{6i} \cdots x_{6i+5}$ and $x_{6j} \cdots x_{6j+5}$ are handled independently by both component channels. This broadcast channel is not stationary, although it is block-stationary and loss probability is shift-invariant over both component channels. It does not have long memory: Channel memory vanishes after six steps. Component channels have capacities $\tilde{\mathbf{C}}_0 = 1/3$ and $\tilde{\mathbf{C}}_1 = 1/2$, nevertheless $(\tilde{\mathbf{R}}_0, \tilde{\mathbf{R}}_1) = (1/6, 1/3)$ is an achievable rate pair: The channel alphabet is assumed to be provided with a group structure, one symbol of common information m_0 and two symbols of private information (m_1, m_2) are encoded in six symbols $x_0 \cdots x_5$, in the following way: $x_0 = x_5 = m_1, x_2 = m_2, x_1 = x_4 = m_0 \oplus m_1, x_3 = m_0 \oplus m_2$. The common information can always be recovered from the \mathbf{W}_0 output, the private information can always be recovered from \mathbf{W}_1 output. And $\frac{\tilde{\mathbf{R}}_0}{\tilde{\mathbf{C}}_0} + \frac{\tilde{\mathbf{R}}_1}{\tilde{\mathbf{C}}_1} = 7/6 > 1$. Notice that because losses are not exchangeable, for the input sequence defined by this encoding, the conditional entropy rates per variable $H[\mathbf{X}(\mathbf{Z}_0)|U]$ is $1/2$ while $H[\mathbf{X}(\mathbf{Z}_1)|U]$ is $2/3$.

Hence the class of broadcast erasure channels over which time sharing does not exhaust the capacity region, has still to be determined.

APPENDIX

Sketch of Proof of Lemma 1: Let (U, \mathbf{X}) be a family of input processes. $M_0 = 2^{\lfloor n \mathbf{R}_0 \rfloor}$ elements u_1, \dots, u_{M_0} are generated according to U^n , then for each u_i , $M_1 = 2^{\lfloor n \mathbf{R}_1 \rfloor}$ codewords $x_{i,j}$ are drawn according to $\text{Pr}_{X^n|U^n}$. The message (i, j) is encoded by $x_{i,j}$. To describe decoding let us define as in [9] the following information-spectrum typical sets.

$$\begin{aligned} T_1^n &= \left\{ (u, y_0): \frac{1}{n} \log \frac{\sum \text{Pr}_{X^n|U^n}(x^n|u) W_0^n(y_0|x^n)}{\text{Pr}_{Y_0^n}\{y_0\}} \right. \\ &\quad \left. > \mathbf{R}_0 + \gamma \right\} \\ T_2^n &= \left\{ (x, y_1): \frac{1}{n} \log \frac{W_1^n(y_1|x)}{\text{Pr}_{Y_1^n}\{y_1\}} > \mathbf{R}_0 + \mathbf{R}_1 + \gamma \right\} \\ T_3^n &= \left\{ (u, x, y_1): \frac{1}{n} \log \frac{W_1^n(y_1|x)}{\text{Pr}_{Y_1^n|U^n}\{y_1|u\}} > \mathbf{R}_1 + \gamma \right\}. \end{aligned}$$

Let F_i and $E_{i,j}$ denote the following sets:

$$\begin{aligned} F_i &= \{(u_i, y_0) \in T_1^n\} \\ E_{i,j} &= \{(u_i, x_{i,j}, y_1) \in T_2^n \cap T_3^n\}. \end{aligned}$$

On receiving y_0 over W_0^n the decoder ϕ_0 reproduces i if and only if there exists a unique i such that $(u_i, y_0) \in F_i$. On receiving an output y_1 over W_1^n the decoder ϕ_1 reproduces (i, j) if and only if there exists a unique (i, j) such that $(u_i, x_{i,j}, y_1) \in E_{i,j}$.

Averaging the decoding error probability for W_0^n over the random codes, using the exchangeability of messages, and the union bound, we get for the average error probability on channel W_0

$$\mathbf{e}_0 \leq \Pr\{u_1, x_{1,1}, y_0: (u_1, y_0) \notin T_1^n\} + M_0 \times \sum_{u_1, x_{1,1}, y_0} \Pr\{u_1, y_0\} \sum_{u': (u', y_0) \in T_1^n} \Pr\{u'\}. \quad (12)$$

But by definition of T_1^n for any $(u', y_0) \in T_1^n$

$$\sum_{x'} \Pr\{x'|u'\} W_0^n(y_0|x') > M_0 \times e^{-\gamma n} \times \Pr\{y_0\}$$

hence for any y_0

$$\sum_{u': (u', y_0) \in T_1^n} \Pr\{u'\} \leq \frac{e^{-\gamma n}}{M_0}. \quad (13)$$

Plugging (13) into (12) gives

$$\mathbf{e}_0 \leq \Pr\{T_1^{n,c}\} + e^{-\gamma n}. \quad (14)$$

A similar argument is developed for the decoding error probability over W_1^n .

$$\begin{aligned} \mathbf{e}_1 &\leq \Pr\{T_2^{n,c} \text{ or } T_3^{n,c}\} \\ &+ M_1 M_0 \sum_{u_1, x_{1,1}, y_1} \Pr\{u_1, x_{1,1}, y_1\} \\ &\times \sum_{u', x', (x', y_1) \in T_2^n} \Pr\{u', x'\} \\ &+ M_1 \sum_{u_1, x_{1,1}, y_1} \Pr\{u_1, x_{1,1}, y_1\} \\ &\times \sum_{x', (u_1, x', y_1) \in T_3^n} \Pr\{x'|u_1\}. \end{aligned} \quad (15)$$

But by definition of T_2^n and T_3^n for any y_1 for any u', x' , if

$$(x', y_1) \in T_2^n : W_1^n(y_1|x') > e^{-\gamma n} M_0 M_1 \Pr\{y_1\}$$

and if

$$(u_1, x', y_1) \in T_3^n, W_1^n(y_1|x') > e^{-\gamma n} M_1 \Pr\{y_1|u_1\}$$

hence

$$\sum_{u', x' (x', y_1) \in T_2^n} \Pr\{x', u'\} \leq \frac{e^{-\gamma n}}{M_0 M_1} \quad (16)$$

and

$$\sum_{x' (u_1, x', y_0) \in T_3^n} \Pr\{x'|u_1\} \leq \frac{e^{-\gamma n}}{M_1}. \quad (17)$$

Plugging (16) and (17) into (15)

$$\mathbf{e}_1 \leq \Pr\{T_2^{n,c} \text{ or } T_3^{n,c}\} + 2e^{-\gamma n}. \quad (18)$$

This terminates the proof of Lemma 1. \square

Proof of Lemma 2: Let the broadcast code and input process be defined as in the statement of Lemma 2. Then let

$$\begin{aligned} L_n^1 &\triangleq \{(u, y_0): \Pr\{y_0, u\} \leq e^{-\gamma n} \Pr\{y_0\}\} \\ L_n^2 &\triangleq \{(x, y_1): \Pr\{x, y_1\} \leq e^{-\gamma n} \Pr\{y_1\}\} \\ L_n^3 &\triangleq \{(u, x, y_1): \Pr\{x, y_1|u\} \leq e^{-\gamma n} \Pr\{y_1|u\}\}. \end{aligned}$$

Notice that $L_n^1 \cup L_n^2 \cup L_n^3$ is the event described in the lemma's statement. By the union bound

$$\begin{aligned} \Pr\{L_n^1 \cup L_n^2 \cup L_n^3\} &\leq \mathbf{e}_n \\ &+ \sum_{y_0} \sum_{i=1}^{M_0} \Pr\{(u_i, y_0) \in L_n^1 \text{ and } \phi_0^n(y_0) = i\} \\ &+ \sum_{y_1} \sum_{i=1, j=1}^{M_0, M_1} \Pr\{(u_i, x_j, y_1) \in L_n^2 \text{ and } \phi_1^n(y_1) = (i, j)\} \\ &+ \sum_{y_1} \sum_{i=1, j=1}^{M_0, M_1} \Pr\{(u_i, x_j, y_1) \in L_n^3 \text{ and } \phi_1^n(y_1) = (i, j)\}. \end{aligned}$$

Now, using first the definition of L_n^1 , then the fact that each y_0 belongs to at most one decoding set

$$\begin{aligned} &\sum_{y_0} \sum_{i=1}^{M_0} \Pr\{(u_i, y_0) \in L_n^1 \text{ and } \phi_0^n(y_0) = i\} \\ &= \sum_{i=1}^{M_0} \sum_{y_0: \phi_0^n(y_0)=i} \Pr\{(u_i, y_0) \in L_n^1\} \\ &\leq \sum_{i=1}^{M_0} \sum_{y_0: \phi_0^n(y_0)=i} \Pr\{y_0\} e^{-\gamma n} \\ &\leq \sum_{y_0} \Pr\{y_0\} e^{-\gamma n}. \end{aligned}$$

Similar arguments are applied to the second and third summand, completing the proof of the lemma. \square

ACKNOWLEDGMENT

The authors wish to thank an anonymous reviewer for suggesting a mistake in a previous version of the correspondence, and another reviewer for pointing out [13].

REFERENCES

- [1] A. Albanese, J. Bloemer, J. Edmonds, and M. Luby, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1737–1744, Nov. 1996.
- [2] E. Biersack, "Performance evaluation of forward error correction in an ATM environment," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 631–640, 1993.
- [3] V. Paxson, "Measurements and analysis of end-to-end internet traffic," Ph.D. dissertation, Univ. Calif. Berkeley, Feb. 1997.
- [4] J. D. Gibson, T. Berger, and D. Lindbergh, *Digital Compression for Multimedia: Principles and Standards*. San Francisco, CA: Morgan Kaufmann, 1998.
- [5] I. Csiszár and J. Körner, "Information theory: coding theorems for discrete memoryless channels," in *Probability and Mathematical Statistics*. New York: Academic, 1981.
- [6] T. Cover, "Broadcast channels," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 2–14, 1972.
- [7] S. Verdù and T. S. Han, "A general formula for channel capacity," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1147–1157, 1994.
- [8] J. Körner and K. Marton, "General broadcast channel with degraded message set," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 60–64, 1977.
- [9] T. S. Han, "An information spectrum approach to capacity theorems for the general multiple-access channel," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2773–2795, Nov. 1998.
- [10] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [11] F. R. K. Chung, R. L. Graham, P. Frankl, and J. B. Shearer, "Some intersection theorems for ordered sets and graphs," *J. Comb. Theory, Ser. A*, vol. 43, pp. 23–37, 1986.
- [12] S. Boucheron and K. Salamatian, "Codage à protections inégales et diffusion," in *Actes du 16ème GRETSI*, 1997, pp. 547–550.

- [13] R. Urbanke and A. D. Wyner, "Packetizing for the erasure broadcast channel with an Internet application," in *Int. Conf. Combinatorics, Information Theory and Statistics*, 1997, p. 93.
- [14] J. Körner and A. Sgarro, "Universally attainable error exponents for broadcast channels with degraded message sets," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 670–679, Nov. 1980.
- [15] G. Poltyrev, "Random coding bounds for some broadcast channels," *Probl. Pered. Inform.*, vol. 19, no. 1, pp. 9–20, 1983.
- [16] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1967.

Sequential Decoding for the Exponential Server Timing Channel

Rajesh Sundaresan, *Student Member, IEEE*, and
Sergio Verdú, *Fellow, IEEE*

Abstract—We show the existence of a good tree code with a sequential decoder for the exponential server timing channel. The expected number of computations before moving one step ahead is upper-bounded by a finite number. The rate of information transfer for this code is $\mu/(2e)$ nats per second, i.e., one half of the capacity. The cutoff rate for the exponential server queue is therefore at least $\mu/(2e)$ nats per second.

Index Terms—Computation, decoding metric, sequential decoder, single-server queue, timing channel, tree codes.

I. INTRODUCTION

Sequential decoding of convolutional codes and tree codes ([1]–[5], etc.) is a useful decoding technique wherein the average number of computations performed is linear in block length as compared to an exponential number of computations for the maximum-likelihood decoder. A vast majority of the literature on sequential decoding deals with memoryless channels. A few papers, (for example, [6], [7]) extend the sequential decoding technique to a class of channels with memory, namely, finite-state channels. In this work we show that the sequential decoding technique can be used on timing channels (for example, [8] and [9]). Interestingly, this timing channel is a channel with memory and cannot be described within the class of finite-state channels.

Specifically, we want to transmit information reliably through a single-server queue [8], [9], at rates below *half* the capacity, but with manageable decoding complexity. In [8]–[10], a decoding technique for block codes was described where the number of computations is exponential in n , the number of packets. By imposing a tree structure on the codes and using the sequential decoding technique, we save on computations at the expense of the rate at which information is reliably transmitted. This work is perhaps a first step in the direction of finding good codes for communication over timing channels.

There are many versions of the sequential decoding technique. The basic idea behind the Fano algorithm [3] is to move forward in the de-

coding tree so long as we seem to be (based on a metric) on the right track. Once the metric falls below a certain threshold, we backtrack and explore other paths, possibly changing the value of the threshold to account for the changed circumstances. The stack algorithm [4], [5], extends the node with the highest metric at each stage, until the end of the tree is reached. There is a relation between the number of computations in both these algorithms.

We are interested in finding bounds on the average number of computations before proceeding one step forward in the correct path. The difficulty with analyzing the performance of the sequential decoding technique for communication systems with memory is the following. When comparing two paths that are the same up to a certain node, the choice of one or the other depends on the branches common to both paths in a way that is typically difficult to handle. For memoryless channels, however, the metric that determines this choice can be selected so that the choice does not depend on the common branches.

We can also get over this difficulty for timing channels. We show that the first m branches can be summed up by one quantity that lends itself to a simple analysis. Our proof is based on the proof in [2] for multiple-access channels, restricted to single-user channels. Burke's output theorem for an $M/M/1$ queue plays an important role in determining a suitable metric. The main contributions of this work are the choice of this metric, and a simple analytical artifice (used earlier in [8] in a different context) that shows how the elegant technique in [2] can be modified to prove the existence of a good tree code for this system with memory.

Section II introduces the problem in the appropriate notation and states the result. Section III contains the proof. We conclude with a brief discussion in Section IV.

II. TREE CODES FOR SINGLE-SERVER QUEUE

Before describing the tree code and our result, we briefly describe the channel. The queue is initially empty. The encoder inputs a certain (nonzero) number of packets at time $t = 0$. The last packet input at time $t = 0$ is called the *zeroth* packet. Let y_0 be the time at which the zeroth packet exits the queue after service. The quantity y_0 is therefore the amount of unfinished work at time $t = 0$. Depending on the message to be transmitted, the encoder then sends the first packet at time x_1 seconds, the second packet at time x_2 after the first packet, and so on. Thus the interarrival times of packets are x_1, x_2, \dots . The receiver observes the interdeparture times, y_1, y_2, \dots , following the departure of the zeroth packet. Let $\mathcal{R}_+ = [0, \infty)$. Let $e_\mu(s) = \mu e^{-\mu s}$, $s \in \mathcal{R}_+$. The conditional probability density of the output $y^n = (y_1, \dots, y_n)$ given x^n and y_0 is

$$f_\mu(y^n | x^n, y_0) = \prod_{i=1}^n e_\mu(y_i - w_i) \quad (1)$$

where

$$w_i = \max \left\{ 0, \sum_{j=1}^i x_j - \sum_{j=0}^{i-1} y_j \right\} \quad (2)$$

is the server's idling time before serving the i th packet.

We now describe the tree code. We follow the notation in [2] with a few modifications. At each instant of time t , the source generates a letter $u_t \in \{0, 1, \dots, M-1\}$, and the sequence $\mathbf{u} = (u_1, u_2, \dots)$ is encoded by a tree code \mathbf{g} . The tree \mathbf{g} is such that M edges leave each node of the code tree. Each edge is labeled by an N -tuple of nonnegative real numbers. The root node is labeled by the number of

Manuscript received December 21, 1998; revised September 23, 1999. This work was supported in part by the National Science Foundation under Grant NCR-9523805 002.

The authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA.

Communicated by T. E. Fuja, Associate Editor At Large.

Publisher Item Identifier S 0018-9448(00)01690-4.

packets input at time $t = 0$ including the zeroth packet. We denote by $u^t = (u_1, u_2, \dots, u_t)$ the path leading from the root node to the t th level. The code corresponding to the source sequence u^t is given by $x^{Nt}(u^t) \in \mathcal{R}_+^{Nt}$, where

$$x^{Nt}(u^t) = (x_1(u^1), \dots, x_N(u^1), x_{N+1}(u^2), \dots, x_{Nt}(u^t))$$

is the sequence of interarrival times of the Nt packets for message sequence u^t . Furthermore, we denote the entire codeword corresponding to the source sequence \mathbf{u} by

$$\mathbf{x}(\mathbf{u}) = (x_1(u^1), \dots, x_N(u^1), x_{N+1}(u^2), \dots).$$

The source sequence from m to l is defined to be

$$u_m^l = (u_m, u_{m+1}, \dots, u_l).$$

Similarly, we define

$$x_{Nl}^{Nl}(u^l) = (x_{Nl}(u^{m+1}), \dots, x_{Nl}(u^l)).$$

The set of all paths in \mathbf{g} that diverge from \mathbf{u} at the m th level is called the m th incorrect subtree for the path \mathbf{u} , i.e.,

$$\mathcal{U}_m(\mathbf{u}) = \{\hat{\mathbf{u}} = (u_1, \dots, u_{m-1}, \hat{u}_m, \hat{u}_{m+1}, \dots) : \hat{u}_m \neq u_m\}.$$

Let \mathbf{g} be a tree code. We characterize the source as follows. The source sequence $\mathbf{U} = (U_1, U_2, \dots)$ is an independent and identically distributed (i.i.d.) sequence of random variables where each source letter U_t is uniformly distributed on the set $\{0, 1, \dots, M-1\}$. The tree code \mathbf{g} then transmits information at a rate R nats per unit time, where

$$R = \lim_{t \rightarrow \infty} \frac{\log M^t}{E \left[\sum_{i=0}^{Nt} Y_i \right]} \quad (3)$$

if the limit exists. All logarithms in this work are taken to be natural logarithms. The quantity $E \left[\sum_{i=0}^{Nt} Y_i \right]$ is the average time to receive the Nt packets, when the tree code is \mathbf{g} . This rate can also be written as

$$R = \left(\frac{\log M}{N} \right) / \left(\lim_{t \rightarrow \infty} E \left[\frac{1}{Nt} \sum_{i=0}^{Nt} Y_i \right] \right).$$

The quantity $r = (\log M)/N$ depends only on the structure of the tree, and is a measure of the number nats of information transmitted per packet.

We now define the metric. This metric depends on the quantity r . Fix $0 < \lambda < \mu/2 = \mu'$. We take

$$\Gamma(u^t | y_0, y^{Nt}) = M(x^{Nt}(u^t) | y_0, y^{Nt}) \quad (4)$$

where

$$M(x^n | y_0, y^n) \triangleq \log \left(\frac{f_{\mu'}(y^n | x^n, y_0)}{\prod_{i=1}^n e_{\lambda}(y_i)} \right) - nr(1 + \varepsilon). \quad (5)$$

The bias term $nr(1 + \varepsilon)$ in (5) is to make a fair comparison between paths of different lengths. $M(\cdot, \cdot)$ in (5) is similar to the metric in [2]. Note the dependence on the quantity μ' rather than μ . This is because the quantity $\sqrt{f_{\mu'}(\cdot, \cdot)}$ which determines the metric [2, eq. (4.4)] normalizes to $f_{\mu'}(\cdot, \cdot)$.

The function M in (5) is further related to [2, eq. (4.4)] due to the following special case of Burke's output theorem [11]. Let $\lambda < \mu'$. Let the number of packets Q_0 at time $t = 0$, excluding the zeroth packet, be distributed according to

$$\Pr\{Q_0 = k\} = (1 - \lambda/\mu') (1 - \lambda/\mu')^k, \quad k \in \mathcal{Z}_+.$$

In addition to these packets, the zeroth packet is sent. Thus the zeroth packet sees the queue in steady state upon arrival. Let the arrivals thereafter form a Poisson process of rate λ . The zeroth packet departs the queue at time Y_0 , whose probability density is $e_{\mu'-\lambda}$. Furthermore, at the moment of its departure, the queue is in equilibrium. The output starting from time Y_0 is then a Poisson process of rate λ [11 Fact 2.8.2, p. 60]. In other words,

$$E[f_{\mu'}(y^n | X^n, Y_0)] = \prod_{i=1}^n e_{\lambda}(y_i) \quad (6)$$

where the expectation is with respect to X^n and Y_0 . X^n is a random vector of i.i.d. exponential random variables with mean $1/\lambda$ seconds, Y_0 is independent of X^n , and is exponentially distributed with mean $1/(\mu' - \lambda)$. The right-hand side of (6) is the normalizing denominator within the log function in (5).

The decoder follows the stack algorithm. From a stack containing some paths in \mathbf{g} , the decoder selects a path with the largest metric, extends it to the next level in M possible ways, and stores the M new paths in the stack. A sorting is done as soon as the new paths are added. The stack algorithm terminates for a tree code with finite depth as soon as the last level of the tree reaches the stack top. As mentioned in [1], we shall consider only infinite trees because the average complexity of sequential decoding is most cleanly formalized and conservatively estimated in the framework of infinite trees. For finite trees, we also need to evaluate the probability of error, which occurs when the last level of the tree to reach the stack top is not the correct message sequence. The proof in Section III applies to finite tree codes with simple modifications.

Using (1), the first term in the right-hand side of (5) can be expanded as

$$\log \left(\frac{f_{\mu'}(y^n | x^n, y_0)}{\prod_{i=1}^n e_{\lambda}(y_i)} \right) = \begin{cases} n \log \frac{\mu'}{\lambda} - (\mu' - \lambda) \sum_{i=1}^n y_i + \mu' \sum_{i=1}^n w_i, & \text{if } y_i \geq w_i, \text{ for } i = 1, \dots, n \\ -\infty, & \text{otherwise} \end{cases} \quad (7)$$

where w_i is the idling time defined in (2).

We now make the following important observation. Suppose we compare two paths of lengths j and l , respectively, that are identical for the first $m-1$ nodes and diverge at the m th node. The past up to the first $m-1$ nodes can be summarized by one quantity

$$\tilde{y}_{m-1} = \sum_{i=0}^{N(m-1)} y_i - \sum_{i=1}^{N(m-1)} x_i.$$

This quantity \tilde{y}_{m-1} is the amount of unfinished work at the instant when the $N(m-1)$ st packet arrives. To decide which of the two paths is placed higher on the stack, we can simply treat the $(m-1)$ st node as the root node with \tilde{y}_{m-1} playing the role of y_0 . The terms in (7) common to both paths are the same up to the $(m-1)$ st node. Furthermore, the w_i 's for branches from node m and beyond are unchanged with \tilde{y}_{m-1} in place of y_0 . This is because, for $k > N(m-1)$, we can rewrite (2) as

$$w_k = \max \left\{ 0, \sum_{i=N(m-1)+1}^k x_i - \sum_{i=N(m-1)+1}^{k-1} y_i - \tilde{y}_{m-1} \right\}.$$

Thus the path metric depends on the common nodes only through the unfinished work at the instant of the arrival of the last common packet. This observation is summarized by

$$\Gamma(u^j|y_0, y^{Nj}) = \Gamma(u^{m-1}|y_0, y^{N(m-1)}) + \Gamma(u_m^j|\tilde{y}_{m-1}, y_{N(m-1)+1}^{Nj}) \quad (8)$$

if $j \geq m$. Of course, Γ for the root node is taken to be 0. Comparing $\Gamma(u^j|y_0, y^{Nj})$ and $\Gamma(u^l|y_0, y^{Nl})$, where $l, j \geq m$, and when the two source sequences have identical initial $m-1$ branches, is therefore equivalent to comparing

$$\Gamma(u_m^j|\tilde{y}_{m-1}, y_{N(m-1)+1}^{Nj}) \quad \text{and} \quad \Gamma(u_m^l|\tilde{y}_{m-1}, y_{N(m-1)+1}^{Nl}).$$

Let $C_m(\mathbf{g}, \mathbf{u}, \mathbf{y})$ denote the number of nodes in $\mathcal{U}_m(\mathbf{u})$ that reach the top of the stack for a given tree code \mathbf{g} and a received sequence \mathbf{y} . This is precisely the number of computations made in the m th incorrect subtree. Let

$$C_m(\mathbf{g}) = E[C_m(\mathbf{g}, \mathbf{U}, \mathbf{Y})]$$

be the average number of computations (averaged over the source sequence and output of the channel). The random variables over which the expectation is taken are indicated in upper case letters. For each $L \geq 1$, let

$$D_L(\mathbf{g}) \triangleq \frac{C_1(\mathbf{g}) + \cdots + C_L(\mathbf{g})}{L}.$$

$D_L(\mathbf{g})$ is, therefore, a measure of the average number of computations required to move one step ahead on the correct path [1].

Theorem 1: For every $\delta > 0$, there exists a tree code \mathbf{g} and a constant $A < \infty$ such that the rate of information transfer is R nats per second where $R(1 + \delta) > \mu/(2e)$, and $D_L(\mathbf{g}) \leq A$ for every $L \geq 1$.

III. PROOF

A. Main Steps

Our proof technique to show the existence of a good tree code with sequential decoding is the well-known random coding technique. A tree is characterized by the number of packets at time $t = 0$, and the labels for all the branches. A suitable distribution on these quantities induces a distribution on the set of infinite trees (using extension theorems in probability theory). We state some bounds over this ensemble of trees and thence argue the existence of a good tree. We then prove the stated bounds in the following subsection.

Choose $\varepsilon > 0$ so that $(1 + \delta) > (1 + \varepsilon)^3$. Fix $\lambda = e^{-1}\mu' = \mu/(2e)$. Fix M and N so that $r = (\log M)/N$ satisfies

$$r(1 + \varepsilon) < \log(\mu'/\lambda) = 1 < r(1 + \varepsilon)^2.$$

Each realization \mathbf{g} is a tree of infinite depth having M branches per node, the root node is labeled by a positive integer $Q_0 + 1$, and every branch of the tree is labeled by an N -tuple in \mathcal{R}_+^N . $Q_0 + 1$ is the number of arrivals (including the zeroth packet) at time $t = 0$. The distribution \mathbf{G} on the set of infinite trees is described as follows. Q_0 is selected independent of the other branch labelings according to the distribution

$$\Pr\{Q_0 = k\} = (1 - \lambda/\mu)(\lambda/\mu)^k, \quad \text{for } k \in \mathcal{Z}_+.$$

Furthermore, each N -tuple is i.i.d., and such that each component of the N -tuple is independent and has density e_λ . This induces a distribution \mathbf{G} on the set of infinite trees.

Let

$$T_L(\mathbf{g}, u^L, y^{NL}) = \frac{1}{NL} \sum_{i=1}^{NL} y_i$$

denote the average time for a packet to exit, given the input message is u^L , and the output stream is y^{NL} . Let $T_L(\mathbf{g}) = ET_L(\mathbf{g}, U^L, Y^{NL})$. Consider the random variable $T_L(\mathbf{G})$. The queue is in equilibrium at time $t = 0$, and the arrivals thereafter are Poisson with rate λ . By Burke's output theorem, the departures are also Poisson with rate λ . Hence, for every $L \geq 1$

$$ET_L(\mathbf{G}) = 1/\lambda \quad (9)$$

where the expectation in (9) is with respect to the distribution \mathbf{G} .

From the argument in Section I, while finding the expected number of computations in the m th incorrect subtree, the past up to $m-1$ nodes can be summarized by one quantity \tilde{y}_{m-1} . Equilibrium at $t = 0$ and Poisson arrivals thereafter ensures that the $N(m-1)$ st packet (the last common packet to the paths under consideration) sees the queue in equilibrium upon arrival. \tilde{Y}_{m-1} therefore has the same distribution as Y_0 . Consequently, the random variables $C_m(\mathbf{G})$, $m \geq 1$, are identically distributed. Recall that

$$D_L(\mathbf{G}) = (C_1(\mathbf{G}) + \cdots + C_L(\mathbf{G}))/L.$$

In Section III-B we show the following result.

Proposition 1: If $r(1 + \varepsilon) < \log(\mu'/\lambda)$, there is a finite K such that $EC_1(\mathbf{G}) \leq K$.

Stationarity and the ergodic theorem [12, p. 374] imply that, as $L \rightarrow \infty$, both $T_L(\mathbf{G})$ and $D_L(\mathbf{G})$ converge almost surely to random variables $T'(\mathbf{G})$ and $D(\mathbf{G})$, respectively, such that $ET'(\mathbf{G}) = 1/\lambda$, and $ED(\mathbf{G}) = EC_1(\mathbf{G}) \leq K$. Furthermore, because $Y_0(\mathbf{G})$ has a finite expectation, dominated convergence theorem implies that

$$E\left[\lim_{L \rightarrow \infty} \frac{1}{NL} Y_0(\mathbf{G})\right] = 0.$$

Hence, with

$$T(\mathbf{G}) = T'(\mathbf{G}) + \lim_{L \rightarrow \infty} (Y_0(\mathbf{G}))/L$$

we get $ET(\mathbf{G}) = 1/\lambda$.

From Chebyshev's inequality and the union bound on probabilities, we obtain

$$P\left\{\left\{T(\mathbf{G}) > \frac{1 + \varepsilon}{\lambda}\right\} \cup \left\{D(\mathbf{G}) > \frac{2K(1 + \varepsilon)}{\varepsilon}\right\}\right\} \leq \frac{1 + \varepsilon/2}{1 + \varepsilon}$$

which implies that

$$P\left\{\left\{T(\mathbf{G}) \leq \frac{1 + \varepsilon}{\lambda}\right\} \cap \left\{D(\mathbf{G}) \leq \frac{2K(1 + \varepsilon)}{\varepsilon}\right\}\right\} \geq \frac{\varepsilon/2}{1 + \varepsilon} > 0.$$

Hence, there exists a tree code \mathbf{g} such that $T(\mathbf{g}) \leq (1 + \varepsilon)/\lambda$ and $D(\mathbf{g}) \leq 2K(1 + \varepsilon)/\varepsilon$.

Following the argument in [1], we then get

$$\limsup D_L(\mathbf{g}) \leq 2K(1 + \varepsilon)/\varepsilon$$

and, therefore, $\sup\{D_L(\mathbf{g})A : L \geq 1\} < A$ for some finite A . Moreover, because $r(1 + \varepsilon)^2 > 1$, we get

$$R(1 + \varepsilon)^3 = r(1 + \varepsilon)^3/T(\mathbf{g}) \geq \lambda r(1 + \varepsilon)^2 > \mu/(2e).$$

This concludes the proof of the Theorem. \square

B. Expected Number of Computations Over the Tree Ensemble

In this subsection, we prove Proposition 1. Fix the first incorrect subtree $\mathcal{U}_1(\mathbf{u})$. The number of computations in this subtree is upper-bounded by (cf. [2, eq. (3.1)])

$$C_1(\mathbf{g}, \mathbf{u}, \mathbf{y}) \leq \sum_{l \geq 0} \sum_{j \geq 1} \sum_{\hat{u}^j \in \mathcal{U}_1(\mathbf{u})} \cdot \exp \left\{ \Gamma(\hat{u}^j | y_0, y^{Nj}) - \Gamma(u^l | y_0, y^{Nl}) \right\}.$$

Our aim is to find the expected value of this upper bound over the code ensemble and the output. Clearly, this average value does not depend on the source sequence due to symmetry.

We now look at a \hat{u}^j in the first incorrect subtree. The distribution of \mathbf{G} is such that the choice of $x_1(\hat{u}^1), \dots, x_{Nj}(\hat{u}^j)$, is independent of the choice of $\mathbf{x}(\mathbf{u})$. Consequently, taking the expectation with respect to the choice of $x_1(\hat{u}^1), \dots, x_{Nj}(\hat{u}^j)$, and denoting that expectation by $\hat{E}[\cdot]$ as in [2], we get

$$\hat{E}C_1(\mathbf{G}, \mathbf{u}, \mathbf{y}) \leq \sum_{l \geq 0} \sum_{j \geq 1} \exp \left\{ -\Gamma(u^l | y_0, y^{Nl}) \right\} \cdot \sum_{\hat{u}^j \in \mathcal{U}_1(\mathbf{u})} \hat{E} \left[\exp \left\{ \Gamma(\hat{u}^j | y_0, y^{Nj}) \right\} \right]. \quad (10)$$

The last summation in (10) can be upper-bounded as follows. This would have been straightforward if it were not for the memory represented by y_0 .

Lemma 1:

$$\sum_{\hat{u}^j \in \mathcal{U}_1(\mathbf{u})} \hat{E} \left[\exp \left\{ \Gamma(\hat{u}^j | y_0, y^{Nj}) \right\} \right] \leq e^{-Nj r \varepsilon} \cdot \frac{e^{(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')}.$$

Proof: There are $\exp\{jNr\}$ nodes at depth j in the set $\mathcal{U}_1(\mathbf{u})$. The left-hand side is, therefore, equal to

$$e^{jNr} \cdot e^{-jNr(1+\varepsilon)} \cdot \hat{E} \left[\frac{f_{\mu'}(y^{Nj} | X^{Nj}, y_0)}{\prod_{i=1}^{Nj} e_{\lambda}(y_i)} \right] \quad (11)$$

where the expectation $\hat{E}[\cdot]$ is with respect to X^{Nj} , which represents the branch labelings for a generic path in the first incorrect subtree.

We now introduce an auxiliary random variable Z which denotes the number of packets in the system when the zeroth packet departs after service. The conditional distribution of Z given $Y_0 = y_0$ is

$$P'_{Z|Y_0}(z|y_0) = \frac{(\lambda y_0)^z e^{-\lambda y_0}}{z!}$$

for $z \in \mathcal{Z}_+$. The marginal of Z when the service times are independent and have density $e_{\mu'}$ is given by

$$P'_Z(z) = \left(1 - \frac{\lambda}{\mu'}\right) \left(\frac{\lambda}{\mu'}\right)^z$$

for $z \in \mathcal{Z}_+$. The prime indicates that the service times have density $e_{\mu'}$. Observe that

$$\begin{aligned} P'_{Z|Y_0}(z|y_0) &= P'_Z(z) \frac{e^{(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')} \frac{(\mu' y_0)^z e^{-\mu' y_0}}{z!} \\ &\leq P'_Z(z) \frac{e^{(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')} \end{aligned} \quad (12)$$

where (12) follows from $(\mu' y_0)^z e^{-\mu' y_0} / z! \leq 1$ for every $z \in \mathcal{Z}_+$ and $y_0 \in \mathcal{R}_+$. Let

$$P'_{Y^{Nj}|X^{Nj}, Y_0}, P'_{Y^{Nj}|Y_0}, P'_{Y^{Nj}|Y_0, Z}, P'_{Y^{Nj}}$$

denote the conditional densities of Y^{Nj} given the indicated random variables. We then have the following sequence of inequalities:

$$\begin{aligned} &\hat{E} \left[f_{\mu'}(y^{Nj} | X^{Nj}, y_0) \right] \\ &= \hat{E} \left[P'_{Y^{Nj}|X^{Nj}, Y_0}(y^{Nj} | X^{Nj}, y_0) \right] \\ &= P'_{Y^{Nj}|Y_0}(y^{Nj} | y_0) \\ &= \sum_{z \in \mathcal{Z}_+} P'_{Z|Y_0}(z|y_0) P'_{Y^{Nj}|Y_0, Z}(y^{Nj} | y_0, z) \\ &\stackrel{a)}{=} \sum_{z \in \mathcal{Z}_+} P'_{Z|Y_0}(z|y_0) P'_{Y^{Nj}|Z}(y^{Nj} | z) \\ &\stackrel{b)}{\leq} \sum_{z \in \mathcal{Z}_+} P'_Z(z) P'_{Y^{Nj}|Z}(y^{Nj} | z) \frac{e^{(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')} \\ &\stackrel{c)}{=} P'_{Y^{Nj}}(y^{Nj}) \frac{e^{(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')}, \\ &= \frac{e^{(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')} \prod_{i=1}^{Nj} e_{\lambda}(y_i) \end{aligned} \quad (13)$$

where a) follows because Y_0 and Y^{Nj} are conditionally independent given Z , a consequence of the memoryless property of the interarrival times; b) follows from (12); in equality c), the dependence on y_0 has been successfully separated; equality (13) follows from (6).

Substitution of (13) in (11) yields the lemma. \square

We continue with the proof of Proposition 1. Observe that the random variables in the right-hand side of (10) are Y_0 and $(\mathbf{X}(\mathbf{u}), \mathbf{Y})$. Substitution of (5) and the result of Lemma 1 in (10), followed by the expectation operation with respect to Y_0 and $(\mathbf{X}(\mathbf{u}), \mathbf{Y})$, yields

$$\begin{aligned} EC_1(\mathbf{G}, \mathbf{u}, \mathbf{Y}) &\leq \sum_{l \geq 0} \sum_{j \geq 1} e^{-jNr\varepsilon} e^{-lNr(1+\varepsilon)} \\ &\cdot \int_{\mathcal{R}_+} dy_0 P_{Y_0}(y_0) \frac{e^{(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')} \\ &\cdot \left[\int_{\mathcal{R}_+^{Nl}} dy^{Nl} E \left[f_{\mu'}(y^{Nl} | X^{Nl}, y_0) \left(\frac{\prod_{i=1}^{Nl} e_{\lambda}(y_i)}{f_{\mu'}(y^{Nl} | X^{Nl}, y_0)} \right) \right] \right] \end{aligned} \quad (14)$$

where the expectation in the innermost integral in (14) is with respect to X^{Nl} . Observe that

$$\begin{aligned} E \left[\frac{f_{\mu'}(y^{Nl} | X^{Nl}, y_0)}{f_{\mu'}(y^{Nl} | X^{Nl}, y_0)} \right] &= E \left[f_{\mu'}(y^{Nl} | X^{Nl}, y_0) \right] \left(\frac{4}{\mu}\right)^{Nl} \\ &\leq \left(\prod_{i=1}^{Nl} e_{\lambda}(y_i) \right) \frac{e^{(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')} \left(\frac{4}{\mu}\right)^{Nl} \end{aligned} \quad (15)$$

where (15) follows from (13). Furthermore, because $2\mu' = \mu$ and $P_{Y_0}(y_0) = e_{\mu - \lambda}(y_0)$, we obtain

$$\int_{\mathcal{R}_+} dy_0 P_{Y_0}(y_0) \frac{e^{2(\mu' - \lambda)y_0}}{(1 - \lambda/\mu')^2} = \frac{(\mu/\lambda - 1)}{(1 - \lambda/\mu')^2} \quad (16)$$

and

$$\int_{\mathcal{R}_+^{Nl}} dy^{Nl} \left(\prod_{i=1}^{Nl} e_{\lambda}(y_i) \right)^2 = \left(\int_{\mathcal{R}_+} dy \lambda^2 e^{-2\lambda y} \right)^{Nl} = (\lambda/2)^{Nl}. \quad (17)$$

Substitution of (15)–(17) in (14) yields

$$EC_1(\mathbf{G}, \mathbf{u}, \mathbf{Y}) \leq \sum_{l \geq 0} \sum_{j \geq 1} e^{-jNr\varepsilon} \cdot \frac{(\mu/\lambda - 1)}{(1 - \lambda/\mu')^2} \cdot \exp \left\{ lN \left[r(1 + \varepsilon) - \log \left(\frac{\mu}{2\lambda} \right) \right] \right\}.$$

The summation over j is finite. The summation over l is finite because $r(1 + \varepsilon) < \log(\mu/(2\lambda))$. Consequently, $EC_1(\mathbf{G}) \leq K$, for some finite K . \square

IV. DISCUSSION

We have shown that for every $\delta > 0$, there is a tree code such that the rate of information transfer R , using the sequential decoding technique, satisfies $R(1 + \delta) > \mu/(2e)$ nats per second, and the average number of computations to move one step forward in the correct direction is upper-bounded by a finite number. The quantity $\mu/(2e)$ nats per second is one half of the capacity, and is a lower bound on the cutoff rate for sequential decoding. Some open questions remain. For example, we do not know the cutoff rate for this exponential server timing channel.

Although we have not dealt with discrete-time timing channels [9] in this work, analogous results follow straightforwardly. However, we do not know a closed-form expression for the rate achievable using sequential decoding with an analogous metric. For the geometric service time distribution $P(S = k) = \mu(1 - \mu)^{k-1}$, $k \geq 1$, the corresponding achievable rate in nats per slot is

$$\max_{\lambda \in [0, 1 - \sqrt{1 - \mu})} \lambda \left[\log \left(\frac{1 - \sqrt{1 - \mu}}{1 + \sqrt{1 - \mu}} \right) + \log \left(\frac{2 - \lambda}{\lambda} \right) \right].$$

Let λ^* be the maximizing λ . To remove the dependence on Y_0 as in the continuous-time case (cf. (13)), λ^* should satisfy

$$(1 - (\mu - \lambda^*)) \cdot (2 - \lambda^* + \sqrt{1 - \mu})^2 < 1.$$

Although we have not proved that this holds for all $\mu \in (0, 1)$, numerical evidence indicates that this is so.

In practice, we need trees with finite depth having extra terminating branches. These tail branches ensure that the last few source symbols can also be decoded correctly with high probability. While this causes a loss in rate, the loss is negligible if the number of additional branches is small in comparison to the block length of the code. In this case, we can easily show that the number of computations in each incorrect subtree is upper-bounded by a constant that is independent of the code length. Furthermore, the probability of error, when one of the other terminating leaves reaches the top of the stack, can be made small by choosing a sufficiently long tail [4]. We omit proofs for the rationale of these simple modifications.

If all terminating leaves have the same $\sum_{i=1}^{Nt} x_i$, where t is the maximum depth of the tree, then the state represented by \tilde{y}_t is the same for all terminating leaves, given a sequence of received interdeparture times. All states have therefore merged into a single one. Transmission can then begin afresh, with a decision up to depth t not affecting future decisions.

We finally remark that λ , the net throughput in packets per second, should be smaller than $\mu/2$ for the sequential decoding scheme to work with finite per-branch computational complexity. Therefore, in already

existing systems, information can be piggy-backed through timing in the above tree-code form only if the system is lightly loaded. Moreover, unlike convolutional codes, we need to store the labels for the entire tree at the decoder. Despite these drawbacks, this work is a positive step in the direction of finding good codes for communication over timing channels.

REFERENCES

- [1] E. Arikan, "Sequential decoding for multiple access channels," *IEEE Trans. Inform. Theory*, vol. 34, pp. 246–259, Mar. 1988.
- [2] V. B. Balakirsky, "An upper bound on the distribution of computation of a sequential decoder for multiple-access channels," *IEEE Trans. Inform. Theory*, vol. 42, pp. 399–408, Mar. 1996.
- [3] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
- [4] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, vol. 13, pp. 675–685, 1969.
- [5] K. Zigangirov, "Some sequential decoding procedures," *Probl. Pered. Inform.*, vol. 2, no. 4, pp. 13–25, 1966.
- [6] A. Lapidoth and J. Ziv, "Universal sequential decoding," in *Proc. 1998 IEEE Information Theory Workshop*, Killarney, Ireland, June 1998.
- [7] J. Ziv, "Universal decoding for finite-state channels," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 453–460, July 1985.
- [8] V. Anantharam and S. Verdú, "Bits through queues," *IEEE Trans. Inform. Theory*, vol. 42, pp. 4–18, Jan. 1996.
- [9] A. Bedekar and M. Azizoglu, "The information-theoretic capacity of discrete-time queues," *IEEE Trans. Inform. Theory*, vol. 44, pp. 446–461, Mar. 1998.
- [10] R. Sundaresan and S. Verdú, "Robust decoding for timing channels," *IEEE Trans. Inform. Theory*, vol. 46, pp. 405–419, Mar. 2000.
- [11] J. Walrand, *An Introduction to Queueing Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [12] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, 2nd ed. New York: Oxford Univ. Press, 1992.

Entropy Expressions for Multivariate Continuous Distributions

Georges A. Darbellay and Igor Vajda, *Senior Member, IEEE*

Abstract—Analytical formulas for the entropy and the mutual information of multivariate continuous probability distributions are presented.

Index Terms—Differential entropy, mutual information.

I. INTRODUCTION

The differential entropy of a random vector \mathbf{X} taking its values in \mathbb{R}^n with probability density function $p(\mathbf{x})$ is defined by

$$h(\mathbf{X}) = - \int_{\mathbb{R}^n} d\mathbf{x} p(\mathbf{x}) \ln p(\mathbf{x})$$

Manuscript received April 22, 1998; revised November 3, 1999. This work was supported by the Fonds National Suisse de la Recherche Scientifique under Grant 8220-040089 and by the Grant Academy of the Czech Republic under Grant 102/99/1137.

The authors are with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 182 08 Prague, Czech Republic (E-mail: {dbe}{vajda}@utia.cas.cz).

Communicated by S. Shamai, Associate Editor for Shannon Theory. Publisher Item Identifier S 0018-9448(00)01687-4.

provided that the integral exists. Analytical expressions for the entropy of univariate continuous distributions are known [12], [6]. For multivariate distributions very few formulas appear to be available. A list of such formulas did appear in [2], where, besides the well-known expressions for the normal and log-normal distributions, five other formulas are given. Amazingly enough, out of these five expressions, four are utterly wrong, namely, the formulas for the logistic, Pareto, exponential, and Weibull exponential distributions.

In this short communication we give a series of analytical expressions for the entropy and the mutual information of continuous multivariate distributions. Space constraints having forced us to be very concise, we refer to [8] for more details on the calculations, as well as for additional expressions. It is worth remarking that for the distributions considered in this correspondence the entropy and the mutual information are defined for all values of the distribution parameters. This is not always the case for the variances and the covariances, or even the means [8].

Several nonparametric estimators for the differential entropy have been developed, a review of which may be found in [4]. To our knowledge, the bias of these estimators on finite samples has never been tested on distributions other than the normal. This is clearly very restrictive and the entropy formulas below offer new possibilities, e.g., [7], [9]. It would be very interesting to study how some estimators which have been proved to be consistent perform on finite samples. Furthermore, from the formulas given hereafter, it is possible to construct parametric entropy estimators.

Two mathematical functions will often appear, the gamma function $\Gamma(z)$ and the digamma function $\Psi(z) = d[\ln \Gamma(z)]/dz = \Gamma'(z)/\Gamma(z)$. More information on them may be found in [1]. The determinant of a matrix will be denoted by $\det(\cdot)$.

II. MULTIVARIATE DIFFERENTIAL ENTROPIES

Unlike the entropy of discrete random variables, the entropy of continuous random variables, i.e., the differential entropy, is not invariant under an invertible transformation of the variables. It will thus come as no surprise that the following well-known lemma will indeed be very useful. Its proof may be found in, e.g., [10].

Lemma 1: Let \mathbf{X} and \mathbf{Y} be two vectors of random variables with values in \mathbb{R}^n , such that $\mathbf{Y} = f(\mathbf{X})$ where f is a one-to-one differentiable transformation from \mathbb{R}^n onto itself. Then

$$h(\mathbf{Y}) = h(\mathbf{X}) - \int_{\mathbb{R}^n} d\mathbf{x} p(\mathbf{x}) \ln |J(f(\mathbf{x}))| \quad (1)$$

where $p(\mathbf{x})$ is the probability density of \mathbf{X} and

$$J(\mathbf{y}) = \det \left(\frac{\partial f_i^{-1}(\mathbf{y})}{\partial y_j} \right)_{1 \leq i, j \leq n} \quad (2)$$

is the Jacobian of the inverse transformation f^{-1} (with the notation $f_i^{-1} \equiv (f^{-1})_i$).

For densities belonging to the exponential family [5], it is possible to simplify the calculation of the entropy by using the formula in Lemma 2. Such densities take the form

$$p(\mathbf{x}) = \frac{1}{C(\boldsymbol{\theta})} e^{\boldsymbol{\theta} \cdot T(\mathbf{x})} \quad (3)$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ is a vector of parameters, T is a transformation from \mathbb{R}^n onto itself, and

$$C(\boldsymbol{\theta}) = \int d\mathbf{x} e^{\boldsymbol{\theta} \cdot T(\mathbf{x})} \quad (4)$$

is a normalizing constant. The \cdot denotes the scalar product in \mathbb{R}^n .

Lemma 2: The entropy of a random variable \mathbf{X} , whose density $p(\mathbf{x})$ belongs to the exponential family as defined by (3), is given by

$$h(\mathbf{X}) = \ln C(\boldsymbol{\theta}) - \boldsymbol{\theta} \cdot \nabla_{\boldsymbol{\theta}} [\ln C(\boldsymbol{\theta})]. \quad (5)$$

Proof:

$$\begin{aligned} h(\mathbf{X}) &= - \int d\mathbf{x} p(\mathbf{x}) \ln p(\mathbf{x}) \\ &= \ln C(\boldsymbol{\theta}) - \boldsymbol{\theta} \cdot \int d\mathbf{x} p(\mathbf{x}) T(\mathbf{x}). \end{aligned}$$

But since

$$\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) = \int d\mathbf{x} T(\mathbf{x}) e^{\boldsymbol{\theta} \cdot T(\mathbf{x})}$$

we obtain

$$h(\mathbf{X}) = \ln C(\boldsymbol{\theta}) - \frac{\boldsymbol{\theta} \cdot \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta})}{C(\boldsymbol{\theta})}. \quad \square$$

We now present a list of differential entropies. In each case we give the density, the entropy formula, and a short explanation on how the formula was derived.

- An n -dimensional **Pareto distribution** of type IV has the density

$$\begin{aligned} p(\mathbf{x}) &= \prod_{i=1}^n \frac{\alpha + i - 1}{\gamma_i \theta_i} \left(\frac{x_i - \lambda_i}{\theta_i} \right)^{\frac{1}{\gamma_i} - 1} \\ &\times \left[1 + \sum_{j=1}^n \left(\frac{x_j - \lambda_j}{\theta_j} \right)^{\frac{1}{\gamma_j}} \right]^{-(\alpha+n)} \end{aligned} \quad (6)$$

where $x_i > \lambda_i$, $\alpha > 0$, $\gamma_i > 0$, $\theta_i > 0$ for $i = 1, \dots, n$. A Pareto distribution of type III is obtained by setting $\alpha = 1$. A Pareto distribution of type II is obtained by setting $\gamma_i = 1$, for $i = 1, \dots, n$. If one sets $\gamma_i = 1$ and $\lambda_i = \theta_i$ for $i = 1, \dots, n$, then one gets a Pareto distribution of type I. The joint density of any subset of the components of a Pareto random vector is again of the form (6) [3].

The entropy is

$$\begin{aligned} & - \sum_{i=1}^n \ln \left(\frac{\alpha + i - 1}{\theta_i} \right) + (\alpha + n) \sum_{i=1}^n \frac{1}{\alpha + i - 1} \\ & + \sum_{i=1}^n \ln \gamma_i - [\Psi(1) - \Psi(\alpha)] \left(n - \sum_{i=1}^n \gamma_i \right). \end{aligned} \quad (7)$$

It is obtained by direct integration for a Pareto random vector of type II, say \mathbf{X} , and for a Pareto random vector of type IV, say \mathbf{Y} , from Lemma 1 and the relations $Y_i = \theta_i(X_i - \lambda_i/\theta_i)^{\gamma_i} + \lambda_i$ for $i = 1, \dots, n$.

- The density of the n -dimensional **logistic distribution** is

$$p(\mathbf{y}) = \prod_{i=1}^n \frac{\alpha + i - 1}{\theta_i} e^{-\frac{y_i - \lambda_i}{\theta_i}} \left[1 + \sum_{j=1}^n e^{-\frac{y_j - \lambda_j}{\theta_j}} \right]^{-(\alpha+n)} \quad (8)$$

with $\mathbf{y} \in \mathbb{R}^n$, $\theta_i > 0$ for $i = 1, \dots, n$ and $\alpha > 0$. The joint density of any subset of the components of a logistic random vector \mathbf{Y} is again of the form (8) [11].

The entropy is

$$- \sum_{i=1}^n \ln \left(\frac{\alpha + i - 1}{\theta_i} \right) + (\alpha + n) \sum_{i=1}^n \frac{1}{\alpha + i - 1} + n[\Psi(\alpha) - \Psi(1)]. \quad (9)$$

This formula is derived from Lemma 1 and the fact that a logistic random vector \mathbf{Y} may be obtained by transforming a Pareto random vector \mathbf{X} of type II according to

$$(X_i - \lambda_i)/\theta_i = \exp(-(Y_i - \lambda_i)/\theta_i)$$

with $i = 1, \dots, n$.

- The n -dimensional **Burr distribution** has the density

$$p(\mathbf{y}) = \prod_{i=1}^n (\alpha + i - 1) d_i c_i y_i^{c_i - 1} \left[1 + \sum_{j=1}^n d_j y_j^{c_j} \right]^{-(\alpha + n)} \quad (10)$$

with $y_i > 0$, $c_i > 0$, $d_i > 0$ for $i = 1, \dots, n$ and $\alpha > 0$. The joint density of any subset of the components of a Burr random vector \mathbf{Y} is again of the form (10) [11].

The entropy is

$$-\sum_{i=1}^n \ln(\alpha + i - 1) + (\alpha + n) \sum_{i=1}^n \frac{1}{\alpha + i - 1} - \sum_{i=1}^n \ln\left(c_i d_i^{1/c_i}\right) + [\Psi(1) - \Psi(\alpha)] \sum_{i=1}^n \frac{c_i - 1}{c_i}. \quad (11)$$

This expression is obtained from Lemma 1 and the relations $(X_i - \lambda_i)/\theta_i = d_i Y_i^{c_i}$ for $i = 1, \dots, n$ and where \mathbf{X} is a Pareto II random vector.

- The n -dimensional **exponential distribution** has the density

$$p(\mathbf{y}) = \prod_{i=1}^n \frac{\alpha + i - 1}{\theta_i} e^{-\frac{y_i - \lambda_i}{\theta_i}} \left[\sum_{j=1}^n e^{-\frac{y_j - \lambda_j}{\theta_j}} - n + 1 \right]^{-(\alpha + n)} \quad (12)$$

with $y_i > \lambda_i$, $\theta_i > 0$ for $i = 1, \dots, n$ and $\alpha > 0$. The joint density of any subset of the components of an exponential random vector \mathbf{Y} is again of the form (10) [11].

The entropy is

$$-\sum_{i=1}^n \ln\left(\frac{\alpha + i - 1}{\theta_i}\right) + (\alpha + n) \sum_{i=1}^n \frac{1}{\alpha + i - 1} - \frac{n}{\alpha}. \quad (13)$$

It is obtained from Lemma 1 and the transformation

$$X_i/\theta_i = \exp((Y_i - \lambda_i)/\theta_i)$$

for $i = 1, \dots, n$ and where \mathbf{X} is a Pareto I random vector.

- The density of the n -dimensional **Weibull distribution** is

$$p(\mathbf{y}) = \prod_{i=1}^n \frac{\alpha + i - 1}{\theta_i} c_i \left(\frac{y_i - \lambda_i}{\theta_i}\right)^{c_i - 1} \times e^{-\left(\frac{y_i - \lambda_i}{\theta_i}\right)^{c_i}} \left[\sum_{j=1}^n e^{-\left(\frac{y_j - \lambda_j}{\theta_j}\right)^{c_j}} - n + 1 \right]^{-(\alpha + n)} \quad (14)$$

where $y_i > \lambda_i$, $c_i > 0$, $\theta_i > 0$ for $i = 1, \dots, n$ and $\alpha > 0$. The joint density of any subset of the components of a Weibull random vector \mathbf{Y} is again of the form (14) [11].

The entropy is

$$-\sum_{i=1}^n \ln\left(\frac{\alpha + i - 1}{\theta_i}\right) + (\alpha + n) \sum_{i=1}^n \frac{1}{\alpha + i - 1} - \frac{n}{\alpha} - \sum_{i=1}^n \ln c_i + [\Psi(1) + \ln \alpha] \sum_{i=1}^n \frac{c_i - 1}{c_i}. \quad (15)$$

This expression is obtained from Lemma 1 and the transformation of variables

$$(X_i - \lambda_i)/\theta_i = \exp([\ln(Y_i - \lambda_i)/\theta_i]^{c_i}) - 1$$

for $i = 1, \dots, n$, \mathbf{X} being a Pareto II random vector.

- The density of the n -dimensional **Weinman exponential distribution** is

$$p(\mathbf{x}) = \prod_{j=0}^{n-1} \frac{1}{\theta_j} e^{-\frac{x_j}{\theta_j}} e^{-\frac{x_{j+1} - x_j}{\theta_j}} \quad (16)$$

where the $x_j > 0$ are arranged in increasing order of magnitude, with $\theta_j > 0$ for $j = 0, \dots, n - 1$ [11].

The entropy is

$$\sum_{j=0}^{n-1} \ln \theta_j + n. \quad (17)$$

It is obtained through direct integration.

- The n -dimensional **Ordered Weinman exponential distribution** has the density

$$p(\mathbf{y}) = n! \prod_{j=0}^{n-1} \frac{1}{\theta_j} e^{-\frac{y_j}{\theta_j}} e^{-\frac{y_{j+1} - y_j}{\theta_j}} \quad (18)$$

where $y_n \geq y_{n-1} \geq \dots \geq y_1 > x_0 = 0$. Its entropy is

$$\sum_{j=0}^{n-1} \ln \theta_j + n - \ln(n!). \quad (19)$$

This formula is derived from Lemma 1 and expression (17).

- The two-dimensional **Gamma-exponential distribution** has the density

$$p(x_1, x_2) = \frac{\theta_1^{\theta_2} \theta_3}{\Gamma(\theta_2)} x_1^{\theta_2} e^{-\theta_1 x_1 - \theta_3 x_1 x_2}. \quad (20)$$

It is defined for $x_1, x_2 > 0$, with the parameters $\theta_1, \theta_2, \theta_3 > 0$.

The entropy is

$$1 + \theta_2 - \theta_2 \Psi(\theta_2) + \ln \Gamma(\theta_2) - \ln \theta_3. \quad (21)$$

This expression is derived from Lemma 2 since (20) may be written in the form (3).

III. MUTUAL INFORMATIONS

The mutual information between two random subvectors \mathbf{X}_a and \mathbf{X}_b of an n -dimensional vector $\mathbf{X} = (\mathbf{X}_a, \mathbf{X}_b)$ is the difference of entropies

$$I(\mathbf{X}_a, \mathbf{X}_b) = h(\mathbf{X}_a) + h(\mathbf{X}_b) - h(\mathbf{X}_a, \mathbf{X}_b). \quad (22)$$

We will denote the dimension of \mathbf{X}_a as n_a , and that of \mathbf{X}_b as n_b . Of course $n_a + n_b = n$. Some formulas for the mutual information are given below. The following lemma will be helpful in understand most of them. This lemma is a consequence of the data processing inequality (e.g. [6]).

Lemma 3: Let $f_a : \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_a}$ and $f_b : \mathbb{R}^{n_b} \rightarrow \mathbb{R}^{n_b}$ be invertible functions and $\mathbf{Y}_a = f_a(\mathbf{X}_a)$, $\mathbf{Y}_b = f_b(\mathbf{X}_b)$. Then

$$I(\mathbf{Y}_a, \mathbf{Y}_b) = I(\mathbf{X}_a, \mathbf{X}_b). \quad (23)$$

We now give some analytical expressions.

- For the **Pareto, logistic, Burr, exponential, and Weibull distributions** the mutual information is given by

$$I(\mathbf{X}_a, \mathbf{X}_b) = \ln \frac{\prod_{j=n_a+1}^n (\alpha + j - 1)}{\prod_{j=1}^{n_b} (\alpha + j - 1)} + (\alpha + n_a) \sum_{j=1}^{n_a} \frac{1}{\alpha + j - 1} + (\alpha + n_b) \sum_{j=n_a+1}^n \frac{1}{\alpha + j - n_a - 1} - (\alpha + n) \sum_{j=1}^n \frac{1}{\alpha + j - 1}. \quad (24)$$

The formula (24) is derived from Section II and (22). These five distributions being related by one-to-one transformations, it follows

from Lemma 3 that they have the same mutual information. Lemma 3 also implies that, besides the dimensions, the mutual information can only depend on one of the distribution parameters, namely α .

- For the two-dimensional **ordered Weinman exponential distribution** the mutual information is

$$I(X_1, X_2) = \begin{cases} \ln\left(\frac{1}{\theta_1}\left(\frac{\theta_0}{2} - \theta_1\right)\right) + \Psi\left(\frac{\theta_0}{\theta_0 - 2\theta_1}\right) + \Psi(1), & \text{if } \theta_1 < \frac{\theta_0}{2} \\ \Psi(1), & \text{if } \theta_0 = \frac{\theta_0}{2} \\ \ln\left(\frac{1}{\theta_1}\left(\theta_1 - \frac{\theta_0}{2}\right)\right) + \Psi\left(\frac{2\theta_1}{2\theta_1 - \theta_0}\right) + \Psi(1), & \text{if } \theta_1 > \frac{\theta_0}{2}. \end{cases} \quad (25)$$

It is obtained through direct integration.

- The mutual information of the **Gamma-exponential distribution** is

$$I(X_1, X_2) = \Psi(\theta_2) - \ln \theta_2 + \frac{1}{\theta_2}. \quad (26)$$

It is derived through direct integration. Note that it depends only on the parameter θ_2 . This is a consequence of Lemma 3.

REFERENCES

- [1] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, V. H. Deutscher, Ed. Frankfurt/Main, Germany: abridged version, 1984.
- [2] N. A. Ahmed and D. V. Gokhale, "Entropy expressions and their estimators for multivariate distributions," *IEEE Trans. Inform. Theory*, vol. 35, pp. 688–692, May 1989.
- [3] B. C. Arnold, *Pareto Distributions*. Burtonsville, MD: Int. Coop. Publishing House, 1983.
- [4] J. Beirlant, E. J. Dudewicz, L. Györfi, and E. C. van der Meulen, "Non-parametric entropy estimation: An overview," *Int. J. Math. Stat. Sci.*, vol. 6, pp. 17–39, June 1997.
- [5] L. D. Brown, *Fundamentals of Statistical Exponential Families*. Hayward, CA: Inst. Math. Statist., 1986, vol. 9. Lecture Notes.
- [6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [7] G. A. Darbellay, "Predictability: An information-theoretic perspective," in *Signal Analysis and Prediction*, A. Procházka, J. Uhlíř, P. J. W. Rayner, and N. G. Kingsbury, Eds. Boston, MA: Birkhäuser-Verlag, 1998, pp. 249–262.
- [8] G. A. Darbellay and I. Vajda. (1998, Feb.) Entropy Expressions for Multivariate Continuous Distributions. UTIA, Academy of Sciences, Prague, Czech Republic. [Online]. Available: <http://siprint.utia.cas.cz/darbellay>
- [9] —, "Estimation of the mutual information with data-dependent partitions," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1315–1320, May 1999.
- [10] S. Ihara, *Information Theory for Continuous Systems*. Singapore, Singapore: World Scientific, 1993.
- [11] N. L. Johnson and S. Kotz, *Distributions in Statistics: Continuous Multivariate Distributions*. New York: Wiley, 1972.
- [12] A. C. G. V. Lazo and P. N. Rathie, "On the entropy of continuous probability distributions," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 120–122, 1978.

Source Code with Cost as a Nonuniform Random Number Generator

Te Sun Han, *Fellow, IEEE*, and Osamu Uchida, *Student Member, IEEE*

Abstract—We show that an optimal source code with a cost function for code symbols can be regarded as a random number generator generating a random sequence (not necessarily a sequence of fair coin bits) as the target distribution in the sense that the normalized conditional divergence between the distribution of the generated codeword distribution and the target distribution vanishes as the block length tends to infinity.

Index Terms—Cost function, general source, normalized conditional divergence, random number generation, source code with cost.

I. INTRODUCTION

In the problem of random number generation, the purpose is in general to simulate the source Y with a prescribed distribution q (called the *target* distribution) by using the source X with a given probability p (called the *coin* distribution). von Neumann [1] has initially addressed this problem. He has considered the problem of simulating a fair random bit by repeatedly using a biased coin with an unknown distribution. Elias [2] has clarified that the optimal expected number of generated fair random bits per coin toss is equal asymptotically to the entropy rate of the source X . Moreover, Vembu and Verdú [3] have shown that the optimal rate at which we can generate fair random bits from a general source X with arbitrary accuracy in the sense of some vanishing distance (e.g., the variational distance, the d-bar distance, and the normalized divergence) between the distribution of the generated codeword process and the uniform distribution is equal to $\liminf_{n \rightarrow \infty} (1/n) H(X^n)$. On the other hand, it was conjectured for a long time on the basis of the folklore that an output sequence from an optimal source code is a *uniform* random sequence, because any incompressible sequence seemingly looks like a uniform random sequence. Visweswariah *et al.* [4] and Han [5] have independently made clear that this folklore is in fact true, that is, they have shown that an optimal variable-length source code can be regarded as a variable-length random number generator in the sense that the normalized divergence distance between the distribution of the generated codeword process and the *uniform* distribution actually vanishes as the block length tends to infinity.

On the other hand, as is well known, if we impose *unequal costs* on code symbols, it is no longer optimal to use the code which minimizes the average codeword length. It is instead required to use the codes which minimize the average codeword cost. Several studies have been made on the source coding problem in this interesting setting. Karp [6] has given an algorithm for constructing minimum-redundancy prefix codes with unequal cost symbols. Iwata *et al.* [7] have proposed a universal lossless coding algorithm for minimizing the average codeword cost for stationary sources based on the Lempel-Ziv (LZ78) code. Hereafter, we shall call the code constructed in the case with unequal cost symbols the *source code with cost*. Naturally, there would exist a bias in the frequency of code symbols generated by an optimal source code

Manuscript received July 23, 1999; revised October 18, 1999.

The authors are with the Graduate School of Information Systems, University of Electro-Communications, Chofugaoka 1-5-1, Chofu, Tokyo 182-8585, Japan. Communicated by N. Merhav, Associate Editor for Source Coding. Publisher Item Identifier S 0018-9448(00)01673-4.

with cost. Can we then consider the optimal variable-length source code with cost as a variable-length *nonuniform* random number generator? The purpose of this correspondence is to demonstrate that the answer to this question is “yes.”

II. VARIABLE-LENGTH SOURCE CODING WITH COST

In order to state our problem in a more formal manner, let \mathcal{X} be a *countably infinite* source alphabet and \mathcal{Y} be a *finite* code alphabet, respectively. In the sequel all the logarithms are taken to the base $K \equiv |\mathcal{Y}|$, where $|\mathcal{Y}|$ denotes the cardinality of \mathcal{Y} . We denote the set of all nonnull finite-length sequences taken from \mathcal{Y} by \mathcal{Y}^* . In this correspondence, we consider quite general sources as follows. Let us define a *general source* as an infinite sequence

$$\mathbf{X} = \{X^n = (X_1^{(n)}, \dots, X_n^{(n)})\}_{n=1}^{\infty}$$

of n -dimensional random variables X^n where each component random variable $X_i^{(n)}$ ($1 \leq i \leq n$) takes values in \mathcal{X} . It should be noted here that each component of X^n may change depending on block length n . This implies that the sequence \mathbf{X} is quite general in the sense that it may not satisfy even the consistency condition as usual processes. The class of sources thus defined covers a very wide range of source including all nonstationary and/or nonergodic sources.

We define the *cost function* $c: \mathcal{Y}^* \rightarrow \mathbf{R}^+ \equiv (0, +\infty]$ as follows: First, each symbol $y \in \mathcal{Y}$ is assigned the corresponding cost $c(y)$ such that $0 < c(y) \leq +\infty$ ($\forall y \in \mathcal{Y}$), and then the *additive cost* $c(\mathbf{y})$ of $\mathbf{y} = (y_1, y_2, \dots, y_k) \in \mathcal{Y}^k$ is defined by

$$c(\mathbf{y}) \equiv \sum_{i=1}^k c(y_i). \quad (1)$$

Definition 1: R is called an *achievable variable-length source coding cost-rate* for the source \mathbf{X} if there exists a variable-length prefix encoder $\varphi_n: \mathcal{X}^n \rightarrow \mathcal{Y}^*$ given the cost function $c: \mathcal{Y}^* \rightarrow \mathbf{R}^+$ such that

$$\limsup_{n \rightarrow \infty} \frac{1}{n} E\{c(\varphi_n(X^n))\} \leq R$$

and the infimum of R that are achievable variable-length source coding cost-rates is denoted by $R_v^c(\mathbf{X})$, which we call the *infimum achievable variable-length source coding cost-rate*. \square

Then, we have the following variable-length source coding theorem with cost¹ for the general source \mathbf{X} .

Theorem 1:

$$R_v^c(\mathbf{X}) = \frac{1}{\alpha_c} \limsup_{n \rightarrow \infty} \frac{1}{n} H(X^n) \quad (2)$$

where the *cost capacity* α_c is the positive unique root α of the equation

$$\sum_{y \in \mathcal{Y}} K^{-\alpha c(y)} = 1$$

and

$$H(X^n) \equiv - \sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) \log P_{X^n}(\mathbf{x}).$$

Proof: See the Appendix. \square

¹This kind of theorem has first been shown by Krause [8] for independent and identically distributed (i.i.d.) finite alphabet sources.

III. SOURCE CODE WITH COST AS A NONUNIFORM RANDOM NUMBER GENERATOR

In this section we address the relationship between source codes with cost and nonuniform independent and identically distributed (i.i.d.) random number generators. Given a variable-length prefix encoder $\varphi_n: \mathcal{X}^n \rightarrow \mathcal{Y}^*$, we define for any positive integer m as

$$\mathcal{D}_m \equiv \{\mathbf{x} \in \mathcal{X}^m \mid l(\varphi_n(\mathbf{x})) = m\}$$

where $l(\cdot)$ denotes the length of a string and we put

$$\mathcal{J}(\varphi_n) \equiv \{m \mid \Pr\{X^n \in \mathcal{D}_m\} > 0\}.$$

For any $m \in \mathcal{J}(\varphi_n)$, we define X_m^n as the random variable taking values in \mathcal{D}_m with the distribution given by

$$P_{X_m^n}(\mathbf{x}) \equiv \frac{P_{X^n}(\mathbf{x})}{\Pr\{X^n \in \mathcal{D}_m\}} \quad (\mathbf{x} \in \mathcal{D}_m).$$

For any positive integer m , $V^{(m)}$ indicates an i.i.d. sequence of length m . Let us now define the conditional divergence $D(\varphi_n(X^n) \parallel V^{(I_n)} | I_n)$ by

$$D(\varphi_n(X^n) \parallel V^{(I_n)} | I_n) \equiv \sum_{m \in \mathcal{J}(\varphi_n)} \Pr\{I_n = m\} D(\varphi_n(X_m^n) \parallel V^{(m)})$$

where I_n is the random variable such that $I_n = m$ for $X^n \in \mathcal{D}_m$.

Then, the following theorem shows that, with the cost function $c: \mathcal{Y}^* \rightarrow \mathbf{R}^+$, the optimal variable-length source code with cost can be considered as a variable-length random number generator generating the variable-length i.i.d. random sequence subject to the distribution \mathbf{q}_c corresponding to the cost function $c: \mathcal{Y}^* \rightarrow \mathbf{R}^+$, in the sense that the normalized conditional divergence between the distribution of the generated codeword process and the i.i.d. target distribution vanishes as block length n tends to infinity.

Theorem 2: We assume that the entropy rate of the general source \mathbf{X} has the limit $\lim_{n \rightarrow \infty} (1/n) H(X^n)$.² Let $\varphi_n: \mathcal{X}^n \rightarrow \mathcal{Y}^*$ be any *optimal* variable-length prefix encoder in the sense that

$$\lim_{n \rightarrow \infty} \frac{1}{n} E\{c(\varphi_n(X^n))\} = R_v^c(\mathbf{X}). \quad (3)$$

If we define the probability distribution $\mathbf{q}_c = \{q_c(y)\}_{y \in \mathcal{Y}}$ corresponding to the cost function c by

$$q_c(y) = K^{-\alpha c(y)} \quad (y \in \mathcal{Y}) \quad (4)$$

then we have

$$\lim_{n \rightarrow \infty} \frac{1}{n} D(\varphi_n(X^n) \parallel V^{(I_n)} | I_n) = 0 \quad (5)$$

where $V^{(m)}$ stands for the i.i.d. sequence of length m subject to the distribution \mathbf{q}_c .

Proof: Let $\mathbf{y} \equiv (y_1, y_2, \dots, y_m) \in \mathcal{Y}^m$. From (4) we have

$$\begin{aligned} \Pr\{V^{(m)} = \mathbf{y}\} &= \prod_{i=1}^m q_c(y_i) \\ &= \prod_{i=1}^m K^{-\alpha c(y_i)} \\ &= K^{-\alpha c(\mathbf{y})} \end{aligned}$$

²The sources satisfying this assumption are not limited only to stationary sources.

for all $m \in \mathcal{J}(\varphi_n)$. Then

$$\begin{aligned} D(\varphi_n(X_m^n) \| V^{(m)}) &= \sum_{\mathbf{y} \in \mathcal{Y}^m} \Pr\{\varphi_n(X_m^n) = \mathbf{y}\} \log \frac{\Pr\{\varphi_n(X_m^n) = \mathbf{y}\}}{\Pr\{V^{(m)} = \mathbf{y}\}} \\ &= \alpha_c \sum_{\mathbf{y} \in \mathcal{Y}^m} \Pr\{\varphi_n(X_m^n) = \mathbf{y}\} c(\mathbf{y}) - H(\varphi_n(X_m^n)) \\ &= \alpha_c \sum_{\mathbf{y} \in \mathcal{Y}^m} \Pr\{\varphi_n(X_m^n) = \mathbf{y}\} c(\mathbf{y}) - H(X_m^n) \end{aligned}$$

where the last equality follows from the fact that φ_n is the one-to-one mapping. Thus we have

$$\begin{aligned} \rho_n &\equiv \frac{1}{n} \sum_{m \in \mathcal{J}(\varphi_n)} \Pr\{I_n = m\} D(\varphi_n(X_m^n) \| V^{(m)}) \\ &= \frac{\alpha_c}{n} \sum_{m \in \mathcal{J}(\varphi_n)} \sum_{\mathbf{y} \in \mathcal{Y}^m} \Pr\{I_n = m\} \Pr\{\varphi_n(X_m^n) = \mathbf{y}\} c(\mathbf{y}) \\ &\quad - \frac{1}{n} \sum_{m \in \mathcal{J}(\varphi_n)} \Pr\{I_n = m\} H(X_m^n) \\ &= \frac{\alpha_c}{n} E\{c(\varphi_n(X^n))\} - \frac{1}{n} H(X^n | I_n) \\ &= \frac{\alpha_c}{n} E\{c(\varphi_n(X^n))\} - \frac{1}{n} H(X^n) + \frac{1}{n} H(I_n). \end{aligned} \quad (6)$$

Let

$$c_{\min} \equiv \min_{y \in \mathcal{Y}} c(y) > 0$$

then it follows from $c_{\min} I_n \leq c(\varphi_n(X^n))$ that

$$E(I_n) \leq \frac{E\{c(\varphi_n(X^n))\}}{c_{\min}}$$

which, together with (6) and the inequality (cf. [9])

$$H(I_n) \leq \log[e(E(I_n))]$$

yields

$$\begin{aligned} \rho_n &\leq \frac{\alpha_c}{n} E\{c(\varphi_n(X^n))\} - \frac{1}{n} H(X^n) \\ &\quad + \frac{1}{n} \log \left[e \left(\frac{E\{c(\varphi_n(X^n))\}}{c_{\min}} \right) \right]. \end{aligned} \quad (7)$$

We see from (3) that

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \left[e \left(\frac{E\{c(\varphi_n(X^n))\}}{c_{\min}} \right) \right] = 0.$$

On the other hand, a consequence of Theorem 1 is

$$R_v^c(\mathbf{X}) = \frac{1}{\alpha_c} \lim_{n \rightarrow \infty} \frac{1}{n} H(X^n). \quad (8)$$

Thus by (3), (7), and (8) we conclude that

$$\limsup_{n \rightarrow \infty} \rho_n \leq \alpha_c R_v^c(\mathbf{X}) - \alpha_c R_v^c(\mathbf{X}) = 0$$

which proves (5). \square

Remark 1: We point out that Iwata *et al.*'s universal code [7] satisfies the condition (3) for any stationary source \mathbf{X} , and, therefore, their code can be regarded as providing a *universal* algorithm for nonuniform i.i.d. random number generation in the sense of (5), although it works only when the source alphabet \mathcal{X} is *finite*.

IV. COMPARISON WITH PREVIOUS RESULTS

Han [5] has earlier established the following result on the *optimal* variable-length prefix code with *equal cost* $c(y) = 1$ ($\forall y \in \mathcal{Y}$), i.e., $c(y) = l(\mathbf{y})$ ($\forall \mathbf{y} \in \mathcal{Y}^*$).

Theorem 3 [5]: We assume that the entropy rate of the general source \mathbf{X} has the limit $\lim_{n \rightarrow \infty} (1/n)H(X^n)$. Let $\varphi_n : \mathcal{X}^n \rightarrow \mathcal{Y}^*$ be any *optimal* variable-length prefix encoder satisfying

$$\lim_{n \rightarrow \infty} \frac{1}{n} E\{l(\varphi_n(X^n))\} = R_v(\mathbf{X}) \quad (9)$$

where $R_v(\mathbf{X})$ is the *infimum of achievable variable-length source coding rates*. Then, we have

$$\lim_{n \rightarrow \infty} \frac{1}{n} D(\varphi_n(X^n) \| U^{(I_n)} | I_n) = 0 \quad (10)$$

where $U^{(m)}$ is the i.i.d. sequence subject to *uniform* distribution on \mathcal{Y}^m . \square

We notice here (cf. [5]) that, under the assumption of Theorem 3, $R_v(\mathbf{X})$ is given by

$$R_v(\mathbf{X}) = \lim_{n \rightarrow \infty} \frac{1}{n} H(X^n).$$

It is easy to check that Theorem 3 is a special case of Theorem 2, because, in the case where all code symbols have equal cost $c(y) = 1$, the cost capacity $\alpha_c = 1$ and hence $R_v^c(\mathbf{X}) = R_v(\mathbf{X})$. Our proof of Theorem 2 is just paralleling the original proof of Theorem 3, and hence Theorem 2 is a straightforward generalization of Theorem 3. On the other hand, Visweswariah *et al.* [4] have also shown a variant of Theorem 3, i.e., they have shown that the *optimal* variable-length source code with equal cost $c(y) = 1$ can be considered as a random number generator in the following sense.

Theorem 4: Let $\varphi_n : \mathcal{X}^n \rightarrow \mathcal{Y}^*$ be any variable-length prefix encoder satisfying the condition (9), where the source alphabet \mathcal{X} is *finite*, unlike in Theorems 2 and 3. Then, there exists a sequence of sets G_n of positive integers such that

$$\begin{aligned} \lim_{n \rightarrow \infty} \Pr\{I_n \in G_n\} &= 1 \\ \lim_{n \rightarrow \infty} \max_{m \in G_n} \frac{1}{m} D(\varphi_n(X_m^n) \| U^{(m)}) &= 0. \end{aligned} \quad \square$$

However, it does not seem to be easy to generalize Theorem 4 to be valid also in the case with unequal costs $c(y)$. One reason is that the rather intractable set G_n intervenes in Theorem 4 but not in Theorem 3. It should be noted that the proof demonstrated in this correspondence does not need the assumption that the source alphabet \mathcal{X} is *finite* and also that either of Theorems 3 or 4 does not imply one another because $G_n \neq \mathcal{J}(\varphi_n)$ in general.

Remark 2: The existence of the limit $\lim_{n \rightarrow \infty} (1/n)H(X^n)$ for the source \mathbf{X} is the necessary and sufficient condition for (10) to hold under the condition (9). To see this, we need the following theorem on the variable-length random number generation. First, we call R an *achievable variable-length intrinsic randomness rate* for the source \mathbf{X} if there exists a variable-length mapping $\varphi_n : \mathcal{X}^n \rightarrow \mathcal{Y}^*$ such that

$$\liminf_{n \rightarrow \infty} \frac{1}{n} E\{l(\varphi_n(X^n))\} \geq R$$

and

$$\lim_{n \rightarrow \infty} \frac{1}{n} D(\varphi_n(X^n) \| U^{(I_n)} | I_n) = 0.$$

Moreover, the supremum of R that are achievable variable-length intrinsic randomness rates is denoted by $S_v^*(\mathbf{X})$, which we call the *supremum achievable variable-length intrinsic randomness rate*. Then, we have

Theorem 5 (Han [5], [10]): For any general source \mathbf{X} with a countably infinite source alphabet \mathcal{X}

$$S_v^*(\mathbf{X}) = \liminf_{n \rightarrow \infty} \frac{1}{n} H(X^n). \quad \square$$

Since the sufficiency is implied by Theorem 3, it suffices to show the necessity. Suppose that (10) holds. Then, from (10) and Theorem 5, we have

$$\liminf_{n \rightarrow \infty} \frac{1}{n} E\{I(\varphi_n(X^n))\} \leq S_v^*(\mathbf{X}) = \liminf_{n \rightarrow \infty} \frac{1}{n} H(X^n).$$

Moreover, by means of Theorem 1 with $c(y) = 1$ ($\forall y \in \mathcal{Y}$)

$$\limsup_{n \rightarrow \infty} \frac{1}{n} E\{I(\varphi_n(X^n))\} \geq R_v(\mathbf{X}) = \limsup_{n \rightarrow \infty} \frac{1}{n} H(X^n).$$

As a consequence, since (9) implies

$$\limsup_{n \rightarrow \infty} \frac{1}{n} E\{I(\varphi_n(X^n))\} = \liminf_{n \rightarrow \infty} \frac{1}{n} E\{I(\varphi_n(X^n))\}$$

it follows that

$$\liminf_{n \rightarrow \infty} \frac{1}{n} H(X^n) \geq \limsup_{n \rightarrow \infty} \frac{1}{n} H(X^n)$$

which claims that the source \mathbf{X} must have the limit

$$\lim_{n \rightarrow \infty} (1/n)H(X^n) \quad \square$$

V. VARIABLE-LENGTH CODING WITH GENERAL COST FUNCTION

In Section III, we have shown that the optimal variable-length source code with the *additive* cost function $c: \mathcal{Y}^* \rightarrow \mathbf{R}^+$ defined by (1) can be regarded as a variable-length random number generator generating the variable-length *i.i.d.* random sequence $V^{(I_n)}$ subject to the distribution q_c depending on the cost function c . In the same spirit, we may consider the problem of generating a more general stochastic process instead of $V^{(I_n)}$. To do so, what kind of cost function should we introduce? In the following, we consider the generation of an arbitrarily prescribed general stochastic process (which may be nonstationary or nonergodic) satisfying the consistency condition

$$q(\mathbf{y}) = \sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y}\mathbf{y}) \quad (\mathbf{y} \in \mathcal{Y}^*) \quad (11)$$

where q denotes the probability measure. We denote the conditional probability of $y_i \in \mathcal{Y}$ given the sequence $y_1^{i-1} \equiv (y_1, y_2, \dots, y_{i-1}) \in \mathcal{Y}^{i-1}$ by $q(y_i|y_1^{i-1})$ and we assume that there exist some constants q_{\min}, q_{\max} such that

$$0 \leq q(y_i|y_1^{i-1}) \leq q_{\max} < 1 \quad (\forall i, \forall y_i \in \mathcal{Y}, \forall y_1^{i-1} \in \mathcal{Y}^{i-1}) \quad (12)$$

$$0 < q_{\min} \leq \inf_{i, y_i, y_1^{i-1}: q(y_i|y_1^{i-1}) > 0} q(y_i|y_1^{i-1}). \quad (13)$$

Using this conditional probability, the probability $q(\mathbf{y})$ of $\mathbf{y} \in \mathcal{Y}^l$ is written as

$$q(\mathbf{y}) = \prod_{i=1}^l q(y_i|y_1^{i-1}) \quad (\mathbf{y} \in \mathcal{Y}^l).$$

Let us now define the *general cost function* $c: \mathcal{Y}^* \rightarrow \mathbf{R}^+$ as

$$c(\mathbf{y}) \equiv -\log q(\mathbf{y}) = -\sum_{i=1}^l \log q(y_i|y_1^{i-1}) \quad (\mathbf{y} \in \mathcal{Y}^l). \quad (14)$$

Define the *conditional cost* $c(y_i|y_1^{i-1})$ of $y_i \in \mathcal{Y}$ given the sequence $y_1^{i-1} \in \mathcal{Y}^{i-1}$ by

$$c(y_i|y_1^{i-1}) \equiv -\log q(y_i|y_1^{i-1})$$

and call the root $\alpha = \alpha_c$ of the equation

$$\sum_{y_i \in \mathcal{Y}} K^{-\alpha c(y_i|y_1^{i-1})} = 1$$

the *cost capacity* α_c of the general cost function c . It is then obvious that $\alpha_c = 1$ for all $y_1^{i-1} \in \mathcal{Y}^{i-1}$. Then, as a general version of Theorem 1, we have the following variable-length source coding theorem with the general cost function (14) for the general source \mathbf{X} . First, let us call R an *achievable variable-length source coding cost-rate* for the source \mathbf{X} if there exists a variable-length prefix encoder $\varphi_n: \mathcal{X}^n \rightarrow \mathcal{Y}^*$ with the general cost function (14) such that

$$\limsup_{n \rightarrow \infty} \frac{1}{n} E\{c(\varphi_n(X^n))\} \leq R$$

and the infimum of R that are achievable variable-length source coding cost-rates is denoted by $R_v^c(\mathbf{X})$, which we call the *infimum achievable variable-length source coding cost-rate*.

Theorem 6:

$$R_v^c(\mathbf{X}) = \frac{1}{\alpha_c} \limsup_{n \rightarrow \infty} \frac{1}{n} H(X^n) \quad (\alpha_c = 1).$$

Proof: On the basis of the assumption (13), we see that there exists a constant c_{\max} such that

$$\sup_{i, y_i, y_1^{i-1}: c(y_i|y_1^{i-1}) < \infty} c(y_i|y_1^{i-1}) \leq c_{\max} < \infty$$

Then, Theorem 6 follows in entirely the same manner as in the proof of Theorem 1, provided that the additive cost $c(y_i)$ is replaced by the conditional cost $c(y_i|y_1^{i-1})$, and accordingly $q(y_i) = K^{-\alpha_c c(y_i)}$ by $q(y_i|y_1^{i-1}) = K^{-\alpha_c c(y_i|y_1^{i-1})}$. \square

Finally, we have the following main theorem of this section which says that the optimal variable-length source code with the general cost function c defined by (14) can be considered as a variable-length random number generator generating the random sequence subject to the given probability measure q .

Theorem 7: We assume that the entropy rate of the general source \mathbf{X} has the limit $\lim_{n \rightarrow \infty} (1/n)H(X^n)$. Given an arbitrary probability measure q satisfying (11)–(13), we define the cost function $c: \mathcal{Y}^* \rightarrow \mathbf{R}^+$ by (14) and let $\varphi_n: \mathcal{X}^n \rightarrow \mathcal{Y}^*$ be any *optimal* variable-length prefix encoder such that

$$\lim_{n \rightarrow \infty} \frac{1}{n} E\{c(\varphi_n(X^n))\} = R_v^c(\mathbf{X}).$$

Then, we have

$$\lim_{n \rightarrow \infty} \frac{1}{n} D(\varphi_n(X^n) \| V_q^{(I_n)} | I_n) = 0$$

where $V_q^{(m)}$ is the random variable subject to the marginal distribution on \mathcal{Y}^m of the probability measure q .

Proof: From the assumption (12) we see that there exists a constant c_{\min} such that

$$0 < c_{\min} \leq c(y_i|y_1^{i-1}) \quad (\forall i, \forall y_i \in \mathcal{Y}, \forall y_1^{i-1} \in \mathcal{Y}^{i-1}).$$

Using this property, we can show Theorem 7 in entirely the same manner as in the proof of Theorem 2, provided that $c(y_i)$, $q(y_i)$ are replaced by $c(y_i|y_1^{i-1})$, $q(y_i|y_1^{i-1})$, respectively. \square

Example: With a finite code alphabet \mathcal{Y} let us consider a Markov process subject to transition probabilities $q(y|y')$ such that $q(y|y') < 1$ ($\forall y, y' \in \mathcal{Y}$). Denoting the initial distribution by $q(y)$, set

$$c(y) = -\log q(y) \quad c(y|y') = -\log q(y|y')$$

and define the cost $c(\mathbf{y})$ of a sequence $\mathbf{y} = (y_1, y_2, \dots, y_n) \in \mathcal{Y}^*$ by

$$c(\mathbf{y}) = c(y_1) + c(y_2|y_1) + c(y_3|y_2) + \dots + c(y_n|y_{n-1}). \quad (15)$$

Then, Theorem 7 tells us that the *optimal* variable-length prefix coding for any general source \mathbf{X} with the cost function (15) asymptotically generates the Markov process subject to the transition probabilities $q(y|y')$. \square

APPENDIX

Proof of Theorem 1

1) *Direct Part:* Without loss of generality we may assume that $0 < c(y) < \infty$ ($\forall y \in \mathcal{Y}$). Let $\mathcal{Y} \equiv \{1, 2, \dots, K\}$ and set $q(i) \equiv K^{-\alpha c(i)}$ ($i = 1, 2, \dots, K$). For any $\mathbf{y} = (y_1, y_2, \dots, y_l) \in \mathcal{Y}^*$, we define

$$\begin{aligned} \alpha(\mathbf{y}) &= \sum_{\mathbf{y}': \mathbf{y}' \prec \mathbf{y}} q(\mathbf{y}') \\ \beta(\mathbf{y}) &= \sum_{\mathbf{y}': \mathbf{y}' \preceq \mathbf{y}} q(\mathbf{y}') \equiv \alpha(\mathbf{y}) + q(\mathbf{y}) \end{aligned}$$

where \prec, \preceq indicate the lexicographic order on the set \mathcal{Y}^l and we have put for $\mathbf{z} = (z_1, z_2, \dots, z_l) \in \mathcal{Y}^l$

$$q(\mathbf{z}) = q(z_1)q(z_2)\dots q(z_l).$$

Let the interval $[\alpha(\mathbf{y}), \beta(\mathbf{y})]$ be denoted by $I(\mathbf{y})$. Obviously, $I(\mathbf{y}) \subset [0, 1]$ and $|I(\mathbf{y})| = K^{-\alpha c(\mathbf{y})}$ (the width of $I(\mathbf{y})$). Then, we first have the following trivial lemma.

Lemma 1: A code $\mathcal{C} \equiv \{\mathbf{y}_1, \mathbf{y}_2, \dots\}$ ($\mathbf{y}_i \in \mathcal{Y}^*$) is prefix if and only if all intervals $I(\mathbf{y}_1), I(\mathbf{y}_2), \dots \subset [0, 1]$ are mutually disjoint. \square

Let all the elements of \mathcal{X}^n be ordered as $\mathcal{X}^n \equiv \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$ and define

$$\begin{aligned} P_i &\equiv \sum_{j=1}^{i-1} P_{X^n}(\mathbf{x}_j) & (i = 1, 2, \dots) \\ Q_i &\equiv P_i + \frac{1}{2} P_{X^n}(\mathbf{x}_i) & (i = 1, 2, \dots) \end{aligned}$$

where $P_1 \equiv 0$. Now, to each \mathbf{x}_i we uniquely assign \mathbf{y}_i as

$$\mathbf{y}_i \equiv \arg \min_{\mathbf{y} \in \mathcal{K}(\mathbf{y})} |\mathbf{y}|$$

where $\mathcal{K}(\mathbf{y})$ is the set of $\mathbf{y} \in \mathcal{Y}^*$ such that $I(\mathbf{y})$ includes Q_i but does not include either P_i or P_{i+1} . It then follows from $I(\mathbf{y}_i) \subset [P_i, P_{i+1})$ that each interval $I(\mathbf{y}_1), I(\mathbf{y}_2), \dots$ is disjoint. Then, from Lemma 1, the code $\mathcal{C} = \{\mathbf{y}_1, \mathbf{y}_2, \dots\}$ is prefix. Therefore, we can define the encoder $\varphi_n: \mathcal{X}^n \rightarrow \mathcal{Y}^*$ by

$$\varphi_n(\mathbf{x}_i) \equiv \mathbf{y}_i.$$

Now, set $\bar{\mathbf{y}}_i \equiv (y_1, y_2, \dots, y_{l-1})$ for each sequence $\mathbf{y}_i = (y_1, y_2, \dots, y_{l-1}, y_l)$. Since $I(\mathbf{y}_i) \subset I(\bar{\mathbf{y}}_i)$ we have $Q_i \in I(\bar{\mathbf{y}}_i)$. Moreover, we see from the definition of $I(\mathbf{y}_i)$ that $P_i \in I(\bar{\mathbf{y}}_i)$ or $P_{i+1} \in I(\bar{\mathbf{y}}_i)$. Then, the width $|I(\bar{\mathbf{y}}_i)|$ of the interval $I(\bar{\mathbf{y}}_i)$ must be larger than $P_{X^n}(\mathbf{x}_i)/2$, so that

$$|I(\bar{\mathbf{y}}_i)| = K^{-\alpha c(\bar{\mathbf{y}}_i)} > \frac{P_{X^n}(\mathbf{x}_i)}{2}$$

from which it follows that

$$\begin{aligned} c(\mathbf{y}_i) &\leq c(\bar{\mathbf{y}}_i) + c_{\max} \\ &< \frac{-\log P_{X^n}(\mathbf{x}_i)}{\alpha_c} + \frac{\log 2}{\alpha_c} + c_{\max} \end{aligned}$$

where $c_{\max} \equiv \max_{y \in \mathcal{Y}} c(y) < \infty$. Then, we have

$$E\{c(\varphi_n(X^n))\} < -\sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) \frac{\log P_{X^n}(\mathbf{x})}{\alpha_c} + \frac{\log 2}{\alpha_c} + c_{\max}$$

which concludes that

$$\limsup_{n \rightarrow \infty} \frac{1}{n} E\{c(\varphi_n(X^n))\} \leq \frac{1}{\alpha_c} \limsup_{n \rightarrow \infty} H(X^n). \quad \square$$

2) Converse Part:

Let $\varphi_n: \mathcal{X}^n \rightarrow \mathcal{Y}^*$ be any variable-length prefix encoder and put

$$c_i \equiv c(\varphi_n(\mathbf{x}_i)) \quad (i = 1, 2, \dots)$$

and define $q_i \equiv K^{-\alpha c_i}$. Then, from Lemma 1, we have

$$q \equiv \sum_{i=1}^{\infty} q_i \leq 1.$$

Set $p_i \equiv P_{X^n}(\mathbf{x}_i)$ and $p \equiv \sum_{i=1}^{\infty} p_i = 1$. From the log-sum inequality [9], we have

$$\begin{aligned} \sum_{i=1}^{\infty} p_i \log \frac{p_i}{q_i} &\geq p \log \frac{p}{q} \\ &= \log \frac{1}{q} \\ &\geq 0. \end{aligned} \quad (16)$$

On the other hand,

$$\begin{aligned} \sum_{i=1}^{\infty} p_i \log \frac{p_i}{q_i} &= -\sum_{i=1}^{\infty} p_i \log q_i + \sum_{i=1}^{\infty} p_i \log p_i \\ &= \alpha_c \sum_{i=1}^{\infty} p_i c_i + \sum_{i=1}^{\infty} p_i \log p_i \\ &= \alpha_c E\{c(\varphi_n(X^n))\} - H(X^n) \end{aligned}$$

which together with (16) implies that

$$E\{c(\varphi_n(X^n))\} \geq \frac{1}{\alpha_c} H(X^n).$$

Then, we conclude that

$$\limsup_{n \rightarrow \infty} \frac{1}{n} E\{c(\varphi_n(X^n))\} \geq \frac{1}{\alpha_c} \limsup_{n \rightarrow \infty} H(X^n). \quad \square$$

REFERENCES

- [1] J. von Neumann, "Various techniques used in connection with random digits," in *Applied Mathematics Series*. Notes by G. E. Forstyle, Washington, DC: Nat. Bur. Stand., 1951, vol. 12, pp. 36–38.
- [2] P. Elias, "The efficient construction of an unbiased random sequences," in *Ann. Math. Statist.*, 1972, vol. 43, pp. 865–870.
- [3] S. Vembu and S. Verdú, "Generating random bits from an arbitrary source: Fundamental limits," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1322–1332, Sept. 1995.
- [4] K. Visweswariah, S. R. Kulkarni, and S. Verdú, "Source codes as random number generators," *IEEE Trans. Inform. Theory*, vol. 44, pp. 462–471, Mar. 1998.
- [5] T. S. Han, *Information-Spectrum Methods in Information Theory* (in Japanese). Tokyo: Baifukan, 1998.
- [6] R. M. Karp, "Minimum redundancy coding for the discrete noiseless channel," *IRE Trans. Inform. Theory*, vol. IT-7, pp. 27–38, Jan. 1961.

- [7] K. Iwata, M. Morii, and T. Uyematsu, "An efficient universal coding algorithm for noiseless channel with symbols of unequal cost," *IEICE Trans. Fundamentals*, vol. E80-A, no. 11, pp. 2232–2237, Nov. 1997.
- [8] R. M. Krause, "Channels which transmit letters of unequal duration," *Inform. Control*, vol. 5, pp. 13–24, 1962.
- [9] I. Csiszár and J. Körner, *Information Theory, Coding Theorems for Discrete Memoryless Systems*. New York: Academic, 1981.
- [10] T. S. Han, Theorems on the variable-length intrinsic randomness, to be published.

A New Recursive Universal Code of the Positive Integers

Hirosuke Yamamoto, *Member, IEEE*

Abstract—A new recursive universal code of the positive integers is proposed, in which any given sequence can be used as a delimiter of codeword while bit "0" is used as a delimiter in known universal codes, e.g., Levenshtein code, Elias ω code, Even–Rodeh code, Stout code, Bentley–Yao code, etc. The codeword length of the proposed code is shorter than $\log_2^* n$ in almost all of sufficiently large positive integers although the known codes are longer than $\log_2^* n$ for any positive integer n .

Index Terms—Elias ω code, log-star function, universal code of positive integers, universal coding.

I. INTRODUCTION

Many researchers have treated the universal coding of the positive integers that satisfy

$$P(n) \geq P(n+1), \quad \text{for any } n \in \mathcal{N}, \quad (1)$$

where $P(n)$ is a probability distribution on the set of positive integers $\mathcal{N} = \{1, 2, 3, \dots\}$ [1]–[7]. These codes can be used practically in various adaptive dictionary codes [8]. Besides the practical uses, it is an interesting coding problem to consider how efficiently we can encode the positive integers under the prefix condition.

Let $\log_2^k n$ be the k -fold composition of the function $\log_2 n$ and let $\log_2^* n$ be

$$\log_2^* n = \log_2 n + \log_2^2 n + \dots + \log_2^{w^*(n)} n \quad (2)$$

where $w^*(n)$ is the largest integer w which satisfies $\log_2^w n \geq 0$. Then, it is shown theoretically that any positive integer n can be represented with $\log_2^* n - \alpha w^*(n)$ bits if $\alpha < \log_2 \log_2 e$ [2], [3].

On the other hand, many researchers, e.g., Levenshtein [2],¹ Elias [4], Bentley–Yao [5], Even–Rodeh [6], Stout [7], etc., have proposed $\log_2^* n$ -type codes with a recursive structure to attain high performance in large n . But, in their codes, codeword length $l(n)$ cannot become shorter than $\log_2^* n$ although it satisfies $l(n) \leq \log_2^* n + w^*(n) + c$ where c is a constant.

Manuscript received June 4, 1998; revised July 23, 1999. The material in this paper was presented in part at the 1998 IEEE International Symposium on Information Theory, MIT, Cambridge, MA, August 16–21, 1998.

The author is with the Department of Mathematical Engineering and Information Physics, University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku, Tokyo 113-8656, Japan (e-mail: yamamoto@hyt.u-tokyo.ac.jp).

Communicated by N. Merhav, Associate Editor for Source Coding.

Publisher Item Identifier S 0018-9448(00)01667-9.

In this correspondence, we propose a new $\log_2^* n$ -type code with a recursive structure, which satisfies that

$$l(n) \leq \log_2^* n - \log_2(1 - 2^{-f})w_f^*(n) + c_f$$

even in the worst cases and

$$l(n) \leq \log_2^* n - (1 + \log_2(1 - 2^{-f}))w_f^*(n) + c_f$$

in the best cases. Here, f is a parameter of the code and c_f is a constant which depends on f . $w_f^*(n)$ is a similar function to $w^*(n)$, which satisfies $w_f^*(n) \leq w^*(n)$.

Since the best and worst cases occur at infinitely many n 's, and, roughly speaking, $l(n)$ is distributed uniformly between two extreme cases, $l(n)$ can become shorter than $\log_2^* n$ in large parts of integers.

In Section II, we review Elias ω code, which is a typical one of the known $\log_2^* n$ -type codes, and we show the reason why the codeword length cannot become shorter than $\log_2^* n$ in the known codes. To overcome this defect, we devise a new representation of binary numbers that never has a given sequence as a prefix. In Section III, we propose a new recursive universal code of the positive integers based on the new binary number representation and we evaluate the performance of the proposed code theoretically. It is shown that the codeword length of the proposed code is shorter than $\log_2^* n$ in almost all of sufficiently large positive integers. The case of r -ary universal codes are treated in Section IV.

We use the following notation in this correspondence.

- $[n]_r$ is the ordinary r -ary number of positive integer n such that the most significant digit of $[n]_r$ is nonzero.
- $[n]_r^i$ is the ordinary r -ary number of n with i digits.
- $[t]$ is the largest integer not exceeding t .

Examples: $[14]_2 = 1110$, $[14]_2^5 = 01110$, $[14]_3 = 112$, $[14]_3^5 = 00112$, $[\log_2 14] = 3$.

II. NEW BINARY NUMBER REPRESENTATION EXCLUDING A FORBIDDEN PREFIX

Elias ω code $C_E(n)$ has the following recursive structure [4]:

$$C_E(n_0) = [n_K]_2 [n_{K-1}]_2 \dots [n_1]_2 [n_0]_2 0 \quad (3)$$

where $[n]_2$ is the ordinary binary number of n , the most significant bit (MSB) of which is always one. Each n_k in (3) is determined recursively by $n_k = \lfloor \log_2 n_{k-1} \rfloor$. In other words, $n_k + 1$ represents the bit length of $[n_{k-1}]_2$. The recursion in (3) stops when the length of $[n_k]_2$ is two. Finally, bit "0" is attached as a delimiter to indicate the end of $C_E(n_0)$.² In the decoding, n_K is obtained from the first two bits of $C_E(n_0)$, and the length of $[n_{k-1}]_2$ is recursively obtained from n_k . Since the MSB of every $[n_k]_2$ is "1," delimiter "0" can stop the recursion and $[n_0]_2$ can easily be found.

Levenshtein W_2 code [2], Even–Rodeh code [6], and Stout code [7] have similar structures and their codes also use bit "0" as a delimiter in the same way as Elias ω code. Levenshtein W'_2 code [2] and Bentley–Yao search-tree code [5] have a little different structure. However, it is known that their code can be derived from Elias ω like code by gathering the MSB's of all $[n_k]_2$ and delimiter "0" as a prefix.

¹Levenshtein code is the first $\log_2^* n$ -type code although Elias ω code is famous.

²" $n_0 = 1$ " is the exception case, for which the codeword is defined as " $C_E(1) = 0$."

We note that the MSB of each $[n_k]_2$ is always "1." This means that the MSB has no information, or it is a redundant bit. But this redundant bit cannot be omitted because the MSB is used to distinguish the delimiter "0." Since each length of $[n_k]_2$ is given by $\lfloor \log_2 n_k \rfloor + 1$ that is larger than $\log_2 n_k$, the codeword length cannot become shorter than $\log_2^* n$ in the known recursive codes of the positive integers.

The above note suggests that if we use some sequence with length $f > 1$, instead of "0," as a delimiter, then some of the redundant bits may be saved from a codeword. When "0" is used as a delimiter, the prefix, i.e., the MSB "1," of each $[n_k]_2$ does not coincide with the delimiter "0." Hence, if we use a sequence $[a]_2^f$, which is the ordinary binary number of integer a with f bits, as a delimiter, then we must devise a new binary number representation of the integers, say $B_{a,f}(n)$, such that the prefix of $B_{a,f}(n)$ does not coincide with delimiter $[a]_2^f$.

Consider binary sequences whose length is less than j bits. Then, the total number of such binary sequences is given by $2^1 + 2^2 + \dots + 2^{j-1} = 2^j - 2$ while the number of the binary sequences with prefix $[a]_2^f$ is given by $2^0 + 2^1 + 2^2 + \dots + 2^{j-1-f} = 2^{j-f} - 1$ if $j - 1 \geq f$ or 0 if $j \leq f$. This means that the number of binary sequences not having prefix $[a]_2^f$ is given by $2^j - 2^{(j-f)_+} - 1$ for any $j \geq 1$ and $f \geq 1$, where $(t)_+$ is defined as

$$(t)_+ = \max\{t, 0\}. \quad (4)$$

Hence, $B_{a,f}(n)$ can be represented by the following formula:

$$B_{a,f}(n) = \begin{cases} [n - M_2(j, f)]_2^j, & \\ \text{if } M_2(j, f) \leq n < M_2(j, f) + N_2(j, f, a), & \\ [n - M_2(j, f - 1)]_2^j, & \\ \text{if } M_2(j, f) + N_2(j, f, a) \leq n < M_2(j + 1, f), & \end{cases} \quad (5)$$

where

$$M_2(j, f) = 2^j - 2^{(j-f)_+} \quad (6)$$

$$N_2(j, f, a) = \lfloor 2^{j-f} \rfloor a = \begin{cases} 2^{j-f} a, & \text{if } j \geq f \\ 0, & \text{if } j < f. \end{cases} \quad (7)$$

Especially, if $a = 0$, i.e., $[a]_2^f = 00 \dots 0$, then (5) can be simplified as follows:

$$B_{0,f}(n) = [n - M_2(j, f - 1)]_2^j, \quad \text{if } M_2(j, f) \leq n < M_2(j + 1, f). \quad (8)$$

We note that letting $\lambda_f(n)$ be the length of $B_{a,f}(n)$, then $\lambda_f(n)$ is given by

$$\lambda_f(n) = j, \quad \text{if } M_2(j, f) \leq n < M_2(j + 1, f). \quad (9)$$

Some examples of $B_{a,f}(n)$ are shown in Table I. Any $B_{a,f}(n)$ does not have $[a]_2^f$ as a prefix. But, $[a]_2^f$ may appear as a prefix when $B_{a,f}(n)$ with length $\lambda_f(n) < f$ is concatenated by another $B_{a,f}(n)$. For instance, when $f = 3$ and $[a]_2^3 = 100$, " $B_{a,f}(5) = 10$ " and " $B_{a,f}(7) = 000$ " makes " $B_{a,f}(5) B_{a,f}(7) = 10000$." In order to prevent such cases, we remove the sequences that coincide with a prefix of $[a]_2^f$ from $\{B_{a,f}(n)\}$. Since one sequence is removed for each length j if $j < f$, the obtained binary number $\tilde{B}_{a,f}(n)$ is given by

$$\tilde{B}_{a,f}(n) = \begin{cases} [n - M_2(j, f) + L(j, f)]_2^j, & \\ \text{if } M_2(j, f) - L(j, f) \leq n < M_2(j, f) & \\ \quad - L(j, f) + \tilde{N}_2(j, f, a) & \\ [n - M_2(j, f - 1) + L(j + 1, f)]_2^j, & \\ \text{if } M_2(j, f) - L(j, f) + \tilde{N}_2(j, f, a) \leq n & \\ \quad < M_2(j + 1, f) - L(j + 1, f) & \end{cases} \quad (10)$$

TABLE I
EXAMPLES OF $B_{a,f}(n)$ AND
 $\tilde{B}_{a,f}(n)$

n	$B_{a,f}(n)$		$\tilde{B}_{a,f}(n)$	
	$[a]_2^3 = 00$	$[a]_2^3 = 100$	$[a]_2^3 = 00$	$[a]_2^3 = 100$
1	0	0	1	0
2	1	1	01	00
3	01	00	10	01
4	10	01	11	11
5	11	10	010	000
6	010	11	011	001
7	011	000	100	010
8	100	001	101	011
9	101	010	110	101
10	110	011	111	110
11	111	101	0100	111
12	0100	110	0101	0000
13	0101	111	0110	0001
14	0110	0000	0111	0010
15	0111	0001	1000	0011
16	1000	0010	1001	0100
17	1001	0011	1010	0101
18	1010	0100	1011	0110
19	1011	0101	1100	0111
20	1100	0110	1101	1010
21	1101	0111	1110	1011
22	1110	1010	1111	1100
23	1111	1011	01000	1101
24	01000	1100	01001	1110
25	01001	1101	01010	1111
26	01010	1110	01011	00000

where

$$L(j, f) = (f - 1) - (f - j)_+, \quad (11)$$

$$\tilde{N}_2(j, f, a) = \lfloor 2^{j-f} \rfloor a. \quad (12)$$

If $[a]_2^f = 00 \dots 0$, then $\tilde{B}_{a,f}(n)$ becomes

$$\tilde{B}_{0,f}(n) = [n - M_2(j, f - 1) + L(j + 1, f)]_2^j, \quad \text{if } M_2(j, f) - L(j, f) \leq n < M_2(j + 1, f) - L(j + 1, f). \quad (13)$$

The length $\tilde{\lambda}_f(n)$ of $\tilde{B}_{a,f}(n)$ is given by

$$\tilde{\lambda}_f(n) = j, \quad \text{if } M_2(j, f) - L(j, f) \leq n < M_2(j + 1, f) - L(j + 1, f). \quad (14)$$

Some examples of $\tilde{B}_{a,f}(n)$ are also shown in Table I.

III. NEW RECURSIVE UNIVERSAL CODE OF POSITIVE INTEGERS

Using $\tilde{B}_{a,f}(n)$ defined by (10), a new recursive universal code of the positive integers can be defined similarly to (3) as follows:

$$C_{a,f}(n_0) = \tilde{B}_{a,f}(n_K) \tilde{B}_{a,f}(n_{K-1}) \dots \tilde{B}_{a,f}(n_1) \tilde{B}_{a,f}(n_0) [a]_2^f \quad (15)$$

where each n_k is given by

$$n_k = \tilde{\lambda}_f(n_{k-1}) - 1 \quad (16)$$

and K is the integer k that satisfies $n_k = 1$. $C_{a,f}(n)$ can also be represented recursively as follows:

$$C_{a,f}(n) = \tilde{C}_{a,f}(n)[a]_2^f \quad (17)$$

$$\tilde{C}_{a,f}(n) = \begin{cases} \tilde{B}_{a,f}(n), & \text{if } n = 1 \\ \tilde{C}_{a,f}(\tilde{\lambda}_f(n) - 1)\tilde{B}_{a,f}(n), & \text{if } n > 1. \end{cases} \quad (18)$$

We note that since $\tilde{B}_{a,f}(1)$ is always equal to "0" or "1," the first segment $\tilde{B}_{a,f}(1)$ can be omitted in the binary case. Some examples of $C_{a,f}(n)$ are shown in Table II. Since a prefix of any $\tilde{B}_{a,f}(n_k)$ does not coincide with $[a]_2^f$, $[a]_2^f$ can delimit the codeword.

We now derive upper bounds on the codeword length $l_{a,f}(n)$ of code $C_{a,f}(n)$ defined by (15) (or (17) and (18)). Let $w_f^*(n)$ be $K + 1$ or the integer k that satisfies $n_k = 0$ for n_k recursively defined by (16). Note that $w_f^*(n)$ is a monotonically increasing function of n and $w_f^*(n) \leq w^*(n)$. Then, the following theorem holds.

Theorem 1: $l_{a,f}(n)$ satisfies for any n that

$$l_{a,f}(n) \leq \log_2^* n + F_2(f)w_f^*(n) + c_f + \delta_2(n) \quad (19)$$

and $l_{a,f}(n)$ satisfies for infinitely many n that

$$l_{a,f}(n) \leq \log_2^* n - (1 - F_2(f))w_f^*(n) + c_f + 2\delta_2(n) \quad (20)$$

where

$$F_2(f) = -\log_2(1 - 2^{-f}) \quad (21)$$

$$c_f = 5(f - 2)_+ + f + 5F_2(f) \quad (22)$$

$$\delta_2(n) \leq \frac{\log_2 e}{n} \left[1 + \frac{(w^*(n) - 1)(\log_2 e)^{w^*(n) - 1}}{\log_2 n} \right] \leq \frac{4.7}{n}. \quad (23)$$

Proof: We note from (14) that in integers n with $\tilde{\lambda}(n) = j$, the smallest and largest ones are given by $n = M_2(j, f) - L(j, f)$ and $n = M_2(j + 1, f) - L(j + 1, f) - 1$, respectively. Hence, from (16), $l_{a,f}(n) - \log_2^* n$ has a local maximum at the following n :

$$n_K = 1 \quad (24)$$

$$n_{k-1} = M_2(n_k + 1, f) - L(n_k + 1, f) \quad (25)$$

$$n = n_0 \quad (26)$$

while it has a local minimum at the following n :

$$n_{k-1} = M_2(n_k + 2, f) - L(n_k + 2, f) - 1. \quad (27)$$

In the following, we derive an upper bound of $l_{a,f}(n)$ for these two extreme cases. We first consider the former case, i.e., the worst case. For $n_k \geq f - 1$, (25) becomes

$$n_{k-1} = 2^{n_k+1} - 2^{n_k+1-f} - (f - 1). \quad (28)$$

Furthermore, for $n_k \leq f - 2$, n_k satisfies

$$\begin{aligned} n_{k-1} &= 2^{n_k+1} - 1 - (f - 1) + (f - n_k - 1) \\ &\geq 2^{n_k+1} - 2^{n_k+1-f} - (f - 1) \\ &= 2^{n_k+1}(1 - 2^{-f}) - (f - 1). \end{aligned} \quad (29)$$

TABLE II
EXAMPLES OF $C_{a,f}(n)$

n	$[a]_2^f = 00$	$[a]_2^f = 100$
1	1 00	0 100
2	1 01 00	0 00 100
3	1 10 00	0 01 100
4	1 11 00	0 11 100
5	1 01 010 00	0 00 000 100
6	1 01 011 00	0 00 001 100
7	1 01 100 00	0 00 010 100
8	1 01 101 00	0 00 011 100
9	1 01 110 00	0 00 101 100
10	1 01 111 00	0 00 110 100
11	1 10 0100 00	0 00 111 100
12	1 10 0101 00	0 01 0000 100
13	1 10 0110 00	0 01 0001 100
14	1 10 0111 00	0 01 0010 100
15	1 10 1000 00	0 01 0011 100
16	1 10 1001 00	0 01 0100 100
17	1 10 1010 00	0 01 0101 100
18	1 10 1011 00	0 01 0110 100
19	1 10 1100 00	0 01 0111 100
20	1 10 1101 00	0 01 1010 100
21	1 10 1110 00	0 01 1011 100
22	1 10 1111 00	0 01 1100 100
23	1 11 01000 00	0 01 1101 100
24	1 11 01001 00	0 01 1110 100
25	1 11 01010 00	0 01 1111 100
26	1 11 01011 00	0 11 00000 100

Hence, for any $k = 1, 2, \dots$, we have

$$n_k + 1 \leq \log_2(n_{k-1} + (f - 1)) + F_2(f). \quad (30)$$

From (30), n_1 is upper-bounded by

$$\begin{aligned} n_1 + 1 &\leq \log_2(n + (f - 1)) + F_2(f) \\ &= \log_2 n + \log_2 \left(1 + \frac{f - 1}{n} \right) + F_2(f) \\ &\leq \log_2 n + \frac{\log_2 e}{n} (f - 1) + F_2(f). \end{aligned} \quad (31)$$

From (30) and (31), $n_2 + 1$ is bounded by

$$\begin{aligned} n_2 + 1 &\leq \log_2(n_1 + (f - 1)) + F_2(f) \\ &\leq \log_2 \left(\log_2 n + \frac{\log_2 e}{n} (f - 1) + F_2(f) + (f - 2) \right) \\ &\quad + F_2(f) \\ &= \log_2^2 n + \log_2 \left(1 + \frac{\frac{\log_2 e}{n} (f - 1) + F_2(f) + (f - 2)}{\log_2 n} \right) \\ &\quad + F_2(f) \\ &\leq \log_2^2 n + \frac{\log_2 e}{\log_2 n} \left(\frac{\log_2 e}{n} (f - 1) + F_2(f) + (f - 2) \right) \\ &\quad + F_2(f). \end{aligned} \quad (32)$$

Repeating such procedure, we obtain

$$\begin{aligned}
n_k + 1 &\leq \log_2^k n + \left(\frac{\log_2 e}{\log_2^{k-1} n} + \frac{(\log_2 e)^2}{(\log_2^{k-1} n)(\log_2^{k-2} n)} \right. \\
&\quad \left. + \cdots + \frac{(\log_2 e)^{k-1}}{(\log_2^{k-1} n)(\log_2^{k-2} n) \cdots (\log_2 n)} \right) \\
&\quad \cdot (F_2(f) + (f-2)) \\
&\quad + \frac{(\log_2 e)^k}{(\log_2^{k-1} n)(\log_2^{k-2} n) \cdots (\log_2 n)} (f-1) + F_2(f) \\
&= \log_2^k n + (1 + A_k(n))F_2(f) + A_{k+1}(2^n)(f-2) + B_k(n)
\end{aligned} \tag{33}$$

where $A_k(n)$ and $B_k(n)$ are defined as

$$A_1(n) = 0 \tag{34}$$

$$A_k(n) = \sum_{j=0}^{k-2} \frac{(\log_2 e)^{j+1}}{(\log_2^{k-1} n)(\log_2^{k-2} n) \cdots (\log_2^{k-1-j} n)} \tag{35}$$

$$B_k(n) = \frac{(\log_2 e)^k}{(\log_2^{k-1} n)(\log_2^{k-2} n) \cdots (\log_2 n)} \tag{36}$$

The codeword length in the local maximum (or worst) cases, say $l_{a,f}^{(W)}(n)$, has the following upper-bound from (16) and (33)

$$\begin{aligned}
l_{a,f}^{(W)}(n) &= \sum_{k=0}^{w_f^*(n)-1} \tilde{\lambda}(n_k) + f \\
&= \sum_{k=0}^{w_f^*(n)-1} (n_{k+1} + 1) + f \\
&= \sum_{k=1}^{w_f^*(n)} (n_k + 1) + f \\
&\leq \sum_{k=1}^{w_f^*(n)} (\log_2^k n + (1 + A_k(n))F_2(f) \\
&\quad + A_{k+1}(2^n)(f-2) + B_k(n)) + f \\
&\leq \log_2^* n + w_f^*(n)F_2(f) + \sum_{k=1}^{w^*(n)} A_k(n)F_2(f) \\
&\quad + \sum_{k=1}^{w^*(n)} A_{k+1}(2^n)(f-2) + \sum_{k=1}^{w^*(n)} B_k(n) + f.
\end{aligned} \tag{37}$$

$$\tag{38}$$

We note from [3, eq. (A-12)] that inequality

$$\sum_{k=1}^{w^*(n)-2} A_k(n) \leq \frac{4b}{3(4-b)} \tag{39}$$

holds for $b = \log_2 e$. Furthermore, $A_{w^*(n)-1}(n)$ and $A_{w^*(n)}(n)$ can be bounded above as follows:

$$\begin{aligned}
A_{w^*(n)-1}(n) &= \frac{b}{\log_2^{w^*(n)-2} n} + \frac{b^2}{(\log_2^{w^*(n)-2} n)(\log_2^{w^*(n)-3} n)} \\
&\quad + \cdots + \frac{b^{w^*(n)-2}}{(\log_2^{w^*(n)-2} n)(\log_2^{w^*(n)-3} n) \cdots \log_2 n} \\
&\leq^* \frac{b}{2} + \frac{b^2}{2 \exp_2(2)} + \frac{b^3}{2 \exp_2(2) \exp_2^2(2)} \\
&\quad + \cdots + \frac{b^{w^*(n)-2}}{2 \exp_2(2) \cdots \exp_2^{w^*(n)-3}(2)} \\
&\leq \frac{b}{2} \left(1 + \frac{b}{4} + \frac{b^2}{4^2} + \cdots + \frac{b^{w^*(n)-3}}{4^{w^*(n)-3}} \right) \\
&\leq \frac{b}{2} \sum_{i=0}^{\infty} \left(\frac{b}{4} \right)^i \\
&= \frac{2b}{4-b}
\end{aligned} \tag{40}$$

$$\begin{aligned}
A_{w^*(n)}(n) &\leq^* b + \frac{b^2}{2} + \frac{b^3}{2 \exp_2(2)} + \cdots \\
&= b \left(1 + \frac{b}{2} + \frac{b^2}{2 \exp_2(2)} + \cdots \right) \\
&= b \left(1 + \frac{2b}{4-b} \right) = \frac{b(4+b)}{4-b}
\end{aligned} \tag{41}$$

where inequality \leq^* holds because of

$$\begin{aligned}
\log^{w^*(n)-1} n &\geq 1 \\
\log^{w^*(n)-2} n &\geq 2
\end{aligned}$$

and

$$\log^{w^*(n)-k} n \geq \exp_2^{k-2}(2)$$

for $k \geq 3$. Hence, we have

$$\begin{aligned}
\sum_{k=1}^{w^*(n)} A_k(n) &\leq \frac{4b}{3(4-b)} + \frac{2b}{4-b} + \frac{b(4+b)}{4-b} \\
&= \frac{b}{4-b} \left(\frac{22}{3} + b \right) \equiv G_2.
\end{aligned} \tag{42}$$

On the other hand,

$$\begin{aligned}
\sum_{k=1}^{w^*(n)} A_{k+1}(2^n) &= \sum_{k=2}^{w^*(n)+1} A_k(2^n) \\
&= \sum_{k=1}^{w^*(2^n)} A_k(2^n)
\end{aligned} \tag{43}$$

because $w^*(2^n) = w^*(n) + 1$ and $A_1(2^n) = 0$ by the definition. Therefore, (43) is also bounded by

$$\sum_{k=1}^{w^*(n)} A_{k+1}(2^n) \leq G_2. \tag{44}$$

Furthermore, the sum of $B_k(n)$, say $\delta_2(n)$, can be bounded by³

$$\begin{aligned} \delta_2(n) &= \sum_{k=1}^{w_f^*(n)} B_k(n) \\ &= \frac{\log_2 e}{n} \\ &\quad \cdot \left[1 + \sum_{k=2}^{w_f^*(n)} \frac{(\log_2 e)^{k-1}}{(\log_2^{k-1} n)(\log_2^{k-2} n) \cdots (\log_2 n)} \right] \end{aligned} \quad (45)$$

$$\leq \frac{\log_2 e}{n} \left[1 + \frac{(w_f^*(n) - 1)(\log_2 e)^{w_f^*(n)-1}}{\log_2 n} \right] \quad (46)$$

$$\leq \frac{4.7}{n} \quad (47)$$

where the last inequality holds because the second term in the bracket of (46) has the maximum at $n = 16$.

From (38), (42), and (44), $l_{a,f}^{(W)}(n)$ can be bounded by

$$\begin{aligned} l_{a,f}^{(W)}(n) &\leq \log_2^* n + w_f^*(n)F_2(f) \\ &\quad + G_2F_2(f) + G_2(f-2)_+ + f + \delta_2(n). \end{aligned} \quad (48)$$

Next we treat the local minimum (or best) case given by (27). In this case, we have for $n_k \geq f - 2$

$$n_{k-1} = 2^{n_k+2} - 2^{n_k+2-f} - (f-1) - 1 \quad (49)$$

and for $n_k \leq f - 3$

$$\begin{aligned} n_{k-1} &= 2^{n_k+2} - 1 - (f-1) + (f - n_k - 2) - 1 \\ &\geq 2^{n_k+2} - 2^{n_k+2-f} - (f-1) - 1 \\ &= 2^{n_k+2}(1 - 2^{-f}) - f. \end{aligned} \quad (50)$$

This means that, for any k , $n_k + 2$ can be bounded by

$$n_k + 2 \leq \log_2(n_{k-1} + f) + F_2(f). \quad (51)$$

Hence, in the same way as (33), we have

$$n_k + 2 \leq \log_2^k n + (1 + A_k(n))F_2(f) + A_{k+1}(2^n)(f-2) + 2B_k(n). \quad (52)$$

Furthermore, from (37), (42), (44), and (52), $l_{a,f}^{(B)}(n)$ can be bounded by

$$\begin{aligned} l_{a,f}^{(B)}(n) &= \sum_{k=1}^{w_f^*(n)} (n_k + 1) + f \\ &= \sum_{k=1}^{w_f^*(n)} (n_k + 2) - w_f^*(n) + f \\ &\leq \sum_{k=1}^{w_f^*(n)} (\log_2^k n + (1 + A_k(n))F_2(f) \\ &\quad + A_{k+1}(2^n)(f-2) + 2B_k(n)) - w_f^*(n) + f \\ &\leq \log_2^* n - (1 - F_2(f))w_f^*(n) + G_2F_2(f) \\ &\quad + G_2(f-2)_+ + f + 2\delta_2(n). \end{aligned} \quad (53)$$

³A tight bound $\delta_2(n) \leq 4/n$ can be obtained by directly calculating (45), where the second term in the bracket has the maximum at $n = 4$.

G_2 is bounded by $G_2 = 4.95 \cdots < 5$ because $b = \log_2 e = 1.4427 \cdots$. Hence, we have from (48) and (53) that

$$l_{a,f}^{(W)}(n) \leq \log_2^* n + F_2(f)w_f^*(n) + 5(f-2)_+ + f + 5F_2(f) + \delta_2(n) \quad (54)$$

$$\begin{aligned} l_{a,f}^{(B)}(n) &\leq \log_2^* n - (1 - F_2(f))w_f^*(n) + 5(f-2)_+ \\ &\quad + f + 5F_2(f) + 2\delta_2(n). \end{aligned} \quad (55)$$

Finally, note that any n satisfies

$$n_{k-1} \geq M_2(n_k + 1, f) - L(n_k + 1, f) \quad (56)$$

instead of (25). But (56) also induces the same inequality (30) as (25). Hence (54) holds for any n . On the other hand, (55) holds for infinite many n 's that satisfy (27). Q.E.D.

We note from (21), (54), and (55) that $F_2(f)$ can be approximated as $F_2(f) \approx (\log_2 e)/2^f$, and by setting f large, the coefficient of $w_f^*(n)$ in the worst case becomes very small while the one in the best case becomes almost -1 . Hence, we can conjecture that $l_{a,f}(n)$ is shorter than $\log_2^* n$ in large parts of the positive integers. In the remainder of this section, we show that this conjecture is true by considering a general case instead of the best and worst cases.

For a given n_k , n_{k-1} must be included in a region $\mathcal{R}(n_k)$ defined as

$$\begin{aligned} \mathcal{R}(n_k) &= \{n_{k-1}: M_2(n_k + 1, f) - L(n_k + 1, f) \leq n_{k-1} \\ &\quad < M_2(n_k + 2, f) - L(n_k + 2, f)\}. \end{aligned} \quad (57)$$

We divide this region $\mathcal{R}(n_k)$ into two regions, the worse region $\mathcal{R}^{(W)}(n_k)$ and the better region $\mathcal{R}^{(B)}(n_k)$, which are defined as

$$\begin{aligned} \mathcal{R}^{(W)}(n_k) &= \{n_{k-1}: M_2(n_k + 1, f) - L(n_k + 1, f) \leq n_{k-1} \\ &\quad < 1.5M_2(n_k + 1, f) - L(n_k + 1, f)\} \end{aligned} \quad (58)$$

$$\begin{aligned} \mathcal{R}^{(B)}(n_k) &= \{n_{k-1}: 1.5M_2(n_k + 1, f) - L(n_k + 1, f) \\ &\quad \leq n_{k-1} < M_2(n_k + 2, f) - L(n_k + 2, f)\}. \end{aligned} \quad (59)$$

Since the cardinality of $\mathcal{R}(n_k)$, $|\mathcal{R}(n_k)|$, is equal to $M_2(n_k + 1, f)$ for $n_k \geq f$, the following relation holds:

$$\frac{|\mathcal{R}^{(W)}(n_k)|}{|\mathcal{R}(n_k)|} = \frac{|\mathcal{R}^{(B)}(n_k)|}{|\mathcal{R}(n_k)|} = \frac{1}{2}. \quad (60)$$

When $n_{k-1} \in \mathcal{R}^{(W)}(n_k)$, it satisfies

$$n_{k-1} \geq M_2(n_k + 1, f) - L(n_k + 1, f). \quad (61)$$

Hence, in this case, we have from (30) that

$$n_k + 1 \leq \log_2(n_{k-1} + (f-1)) + F_2(f). \quad (62)$$

When $n_{k-1} \in \mathcal{R}^{(B)}(n_k)$, it satisfies

$$n_{k-1} \geq 1.5M_2(n_k + 1, f) - L(n_k + 1, f) \quad (63)$$

$$= M_2(n_k + 1 + \log_2 1.5, f) - L(n_k + 1, f). \quad (64)$$

Therefore, we can obtain similarly that

$$n_k + 1 + \log_2 1.5 \leq \log_2(n_{k-1} + (f-1)) + F_2(f). \quad (65)$$

For a given n , letting $\mathcal{K}^{(B)}(n)$ be the set of k such that $n_k \in \mathcal{R}^{(B)}(n_{k+1})$ and letting $\beta(n)$ be the ratio defined as

$$\beta(n) = \frac{|\mathcal{K}^{(B)}(n)|}{w_f^*(n)} \quad (66)$$

then the codeword length of n satisfies from (37), (62), and (65) that

$$\begin{aligned} l_{a,f}(n) &= \sum_{k=1}^{w_f^*(n)} (n_k + 1) + f \\ &= \sum_{k \notin \mathcal{K}^{(B)}(n)} (n_k + 1) \\ &\quad + \sum_{k \in \mathcal{K}^{(B)}(n)} ((n_k + 1 + \log_2 1.5) - \log_2 1.5) + f \\ &\leq \log_2^* n - (\beta(n) \log_2 1.5 - F_2(f)) w_f^*(n) \\ &\quad + G_2 F_2(f) + G_2(f-2)_+ + f + \delta_2(n). \end{aligned} \quad (67)$$

Hence, when n is sufficiently large, the codeword length $l_{a,f}(n)$ becomes shorter than $\log_2^* n$ if $\beta(n) \log_2 1.5 - F_2(f) > 0$, i.e.,

$$\beta(n) > \frac{-\log_2(1-2^{-f})}{\log_2 1.5} \approx \frac{\log_2 e}{\log_2 1.5} 2^{-f} \approx 2.5 \cdot 2^{-f} \quad (68)$$

where the approximation holds for $2^{-f} \ll 1$.

We now show that (68) holds for almost all positive integers. Assume that n is uniformly distributed over the set of integers n satisfying $w_f^*(n) = w$ for a given integer w , and a random variable X_k is defined as

$$X_k = \begin{cases} 0, & \text{if } n_k \in \mathcal{R}^{(W)}(n_{k+1}) \\ 1, & \text{if } n_k \in \mathcal{R}^{(B)}(n_{k+1}) \end{cases} \quad (69)$$

for such probability distribution. Then, from the definition of $\mathcal{R}^{(W)}(n_{k+1})$ and $\mathcal{R}^{(B)}(n_{k+1})$, we have

$$\Pr\{X_K = 0\} = 1 \quad (70)$$

where $K = w - 1$. Furthermore, since

$$|\mathcal{R}^{(W)}(n_{k+1})| = |\mathcal{R}^{(B)}(n_{k+1})|$$

holds for any $n_{k+1} \geq f$ and any integer included in $\mathcal{R}^{(B)}(n_{k+1})$ is larger than integers in $\mathcal{R}^{(W)}(n_{k+1})$, we can easily show that for $w \geq 3$ and $1 \leq k \leq K - 1$

$$\Pr\{X_k = 0 | X_K = 0, X_{K-1} = x_{k-1}, \dots, X_{k+1} = x_{k+1}\} < \frac{1}{4} \quad (71)$$

$$\Pr\{X_0 = 0 | X_K = 0, X_{K-1} = x_{k-1}, \dots, X_2 = x_2, X_1 = x_1\} = \frac{1}{2} \quad (72)$$

hold for any $x_{K-1}x_{K-2} \dots x_{k+1}$ and $x_{K-1}x_{K-2} \dots x_2x_1$, respectively.⁴ Obviously,

$$\Pr\{X_k = 1 | X_K = 0, X_{K-1} = x_{k-1}, \dots, X_{k+1} = x_{k+1}\} \leq \frac{1}{4} \quad (73)$$

⁴For simplicity, a rough upper bound $1/4$ is used in (71) although $\Pr\{X_k = 1 | X_K = 0, X_{K-1} = x_{k-1}, \dots, X_{k+1} = x_{k+1}\} \ll 1/4$ holds.

also holds for any k .

Hence, a sequence $x_{K-1}x_{K-2} \dots x_1x_0$ with $m = \sum_{k=0}^K x_k < w/2$ occurs with probability $\Pr(\mathbf{x})$ bounded above by

$$\begin{aligned} \Pr(\mathbf{x}) &< \left(\frac{1}{2}\right) \left(\frac{1}{4}\right)^{w-1-m} 1^m \\ &= \left(\frac{1}{2}\right) \left(\frac{1}{4}\right)^{w-1-2m} \left(\frac{1}{4}\right)^m 1^m \\ &\leq \left(\frac{1}{2}\right)^{w-2m} \left(\frac{1}{2}\right)^{2m} \\ &= \left(\frac{1}{2}\right)^w. \end{aligned} \quad (74)$$

Therefore, we have

$$\Pr\left\{\beta(N) = \frac{m}{w}\right\} = \Pr\left\{\sum_{k=0}^K X_k = m\right\} < \binom{w}{m} 2^{-w} \quad (75)$$

where N is the random variable of n . This means from the law of the large number for the binomial distribution that for any fixed $\beta_0 < 1/2$

$$\lim_{w \rightarrow \infty} \Pr\{\beta(N) < \beta_0\} = 0. \quad (76)$$

Since it holds that $-\log_2(1-2^{-f})/\log_2 1.5 < 1/2$ for $f \geq 3$, we can conclude that

$$\begin{aligned} \lim_{w \rightarrow \infty} \Pr\{l_{a,f}(N) > \log_2^* N\} \\ \leq \lim_{w \rightarrow \infty} \Pr\left\{\beta(N) < \frac{-\log_2(1-2^{-f})}{\log_2 1.5}\right\} = 0. \end{aligned} \quad (77)$$

Theorem 2: In case of $f \geq 3$, the codeword length of code $C_{a,f}(n)$ is shorter than $\log_2^* n$ in almost all of sufficiently large positive integers in the sense of (77).

IV. CONCLUDING REMARKS

We proposed a new recursive universal code of the positive integers, the codeword length of which is shorter than $\log_2^* n$ in almost all of sufficiently large positive integers.

Although we treated the binary case in the previous sections, the results can easily be extended to r -ary code by using $\log_r n, [n]_r^j, [a]_r^j$,

$$M_r(j, f) = \frac{r^j - r^{(j-f)_+}}{r-1} \quad (78)$$

$$\tilde{N}_r(j, f, a) = \lfloor r^{j-f} a \rfloor \quad (79)$$

instead of $\log_2 n, [n]_2^j, [a]_2^j, M_2(j, f), \tilde{N}_2(j, f, a)$, respectively. For this case, we can show, in the same way as the binary case, that $l_{a,f}^{(W)}(n)$ and $l_{a,f}^{(B)}(n)$ are bounded as follows:

$$\begin{aligned} l_{a,f}^{(W)}(n) &\leq \log_r^* n + F_r(f) w_f^*(n) + G_r(f-2)_+ + f \\ &\quad + G_r F_r(f) + \delta_r(n) \end{aligned} \quad (80)$$

$$\begin{aligned} l_{a,f}^{(B)}(n) &\leq \log_r^* n - (1 - F_r(f)) w_f^*(n) + G_r(f-2)_+ + f \\ &\quad + G_r F_r(f) + 2\delta_r(n) \end{aligned} \quad (81)$$

where

$$F_r(f) = -\log_r \frac{1 - r^{-f}}{r - 1} \quad (82)$$

$$\begin{aligned} G_r &= \frac{r^r \log_r e}{(r^r - 1)(r^r - \log_r e)} + \frac{r^r \log_r e}{r(r^r - \log_r e)} \\ &\quad + \frac{(r^r + (r^r - r) \log_r e) \log_r e}{r(r^r - \log_r e)} \\ &= \frac{\log_r e}{r^r - \log_r e} \left[\frac{(r^r)^2}{r^r - 1} + \frac{r^r + (r^r - r) \log_r e}{r} \right] \end{aligned} \quad (83)$$

$$\delta_r(n) \leq \frac{\log_r e}{n} \left[1 + \frac{w^*(n) - 1}{\log_r n} \right]. \quad (84)$$

G_r goes to $\log_r e$ as r becomes large. We also note that, as f becomes large, $F_r(f)$ goes to $\log_r(r - 1)$, instead of zero, which is approximately equal to one when r is large. This corresponds to the fact that in the r -ary case, the optimal code length is given by $\log_r^* n - \alpha w^*(n)$ with $\alpha < \log_r \log_r e < 0$ [2], [3]. This means that the codeword length cannot become shorter than $\log_r^* n$ for all large n .⁵

Since function $w_f^*(n)$ and $w^*(n)$ are monotonically increasing functions of n , the term of $w_f^*(n)$ in (19) and (20) becomes significant compared with constant terms for large n . But, the rate of increase is very slow. Hence, the overhead of delimiter length f is more severe than the effect of $w_f^*(n)$ in moderate n , and the performance of the proposed code $C_{a,f}(n)$ is no better than the known codes in practical use. We note that $C_{a,f}(n)$ can be improved in the same way as Stout code [7]. But even such modified code is not suited for practical use.

As we noted in Section II, Levenshtein W'_2 code [2] or Bentley–Yao code [5] have a code structure such that

$$C_0(n_0) = 1^{K_0} 0 [n_{K_0}]_2^- [n_{K_0-1}]_2^- \cdots [n_1]_2^- [n_0]_2^- \quad (85)$$

where 1^{K_0} is a sequence of 1's with length K_0 and $[n_k]_2^- = 1[n_k]_2^-$, i.e., $[n_k]_2^-$ is obtained by deleting the MSB from $[n_k]_2$. Compared with $C_E(n_0)$ given by (3), the MSB's and delimiter "0" are gathered as the prefix $1^{K_0} 0$, which is the unary code denoting the recurrence number K_0 . Since the unary code is inefficient for large K_0 , we can use the

⁵ $l_{a,f}^{(B)}(n)$ is a little shorter than $\log_r^* n$ in (81). But this is attained by the loss of (80) which is larger than $\log_r n - \alpha w^*(n)$.

same code $C_0(K_0)$ to represent K_0 instead of $1^{K_0} 0$ [9]. Then the code have the following structure:

$$C_1(n_0) = 1^{S_0} 0 [K_{S_0}]_2^- [K_{S_0-1}]_2^- \cdots [K_1]_2^- [K_0]_2^- [n_{K_0}]_2^- [n_{K_0-1}]_2^- \cdots [n_1]_2^- [n_0]_2^- \quad (86)$$

Furthermore, S_0 in (86) can be represented by $C_0(S_0)$ instead of $1^{S_0} 0$. By repeating such recurrence arbitrarily fixed times t , code $C_t(n_0)$ can be defined. But note that $C_t(n_0)$ is a doubly recursive code while our code $C_{a,f}(n_0)$ is a simple recursive code. The evaluation of the asymptotic performance for $C_t(n_0)$ is an open problem.

Finally, we note that Levenshtein W'_2 code and Bentley–Yao code satisfy the lexicographic property. Recently, Nakamura and Murashima [10] showed that if their devised code $C_{NM}(K_0)$ is used instead of unary code $1^{K_0} 0$ in (85), the length of the code satisfies

$$l(n) \leq \log_2^* n + \frac{w_2^*(n)}{m} + \log_2 m + c \quad (87)$$

for any given integer $m > 0$, and the lexicographic property also holds in their code.

REFERENCES

- [1] T. Amemiya and H. Yamamoto, "A new class of the universal representation for the positive integers," *IEICE Trans. Fundamentals*, vol. E76-A, no. 3, pp. 447–452, Mar. 1993.
- [2] V. I. Levenshtein, "On the redundancy and delay of decodable coding of natural numbers," *Probl. Cybern.*, vol. 20, pp. 149–155, 1968.
- [3] R. Ahlswede, T. S. Han, and K. Kobayashi, "Universal coding of integers and unbounded search trees," *IEEE Trans. Inform. Theory*, vol. 43, pp. 669–682, Mar. 1997.
- [4] P. Elias, "Universal codeword sets and representation of the integers," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 194–203, 1975.
- [5] J. L. Bentley and A. C. Yao, "An almost optimal algorithm for unbounded searching," *Inform. Processing Lett.*, vol. 5, no. 3, pp. 82–87, 1976.
- [6] S. Even and M. Rodeh, "Economical encoding of comma between strings," *Commun. ACM*, vol. 21, no. 4, pp. 315–317, 1978.
- [7] Q. F. Stout, "Improved prefix encodings of the natural numbers," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 607–699, 1980.
- [8] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
- [9] V. I. Levenshtein, private communication, 1998.
- [10] H. Nakamura and S. Murashima, "Construction method of positive integer code by coding MSB string of length information of existing codes" (in Japanese), *Trans. Inform. Processing Soc. of Japan*, vol. 40, no. 4, pp. 1745–1753, Apr. 1999.