[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 decoding 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 $x_1, x_2, \cdots$. The receiver observes the interdeparture times, $y_1, y_2, \cdots$, 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 $y^n = (y_1, \cdots, y_n)$ given x^n and y_0 is

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

where

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

is the server's idling time before serving the ith 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 $u_t \in \{0, 1, \cdots, M-1\}$, and the sequence $\boldsymbol{u} = (u_1, u_2, \cdots)$ is encoded by a tree code $\boldsymbol{g}$. The tree $\boldsymbol{g}$ 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 Septepmber 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 $u^t = (u_1, u_2, \cdots, u_t)$ the path leading from the root node to the tth level. The code corresponding to the source sequence u^t is given by $x^{Nt}\left(u^t\right) \in \mathcal{R}_+^{Nt}$, where

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

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 $\boldsymbol{u}$ by

$$\boldsymbol{x}(\boldsymbol{u}) = \left(x_1\left(u^1\right), \cdots, x_N\left(u^1\right), x_{N+1}\left(u^2\right), \cdots\right).$$

The source sequence from m to l is defined to be

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

Similarly, we define

$$x_{Nm+1}^{Nl}\left(u^l\right) = \left(x_{Nm+1}\left(u^{m+1}\right), \cdots, x_{Nl}\left(u^l\right)\right).$$

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

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

Let $\boldsymbol{g}$ be a tree code. We characterize the source as follows. The source sequence $\boldsymbol{U} = (U_1, U_2, \cdots)$ 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 $\{0, 1, \cdots, M-1\}$. The tree code $\boldsymbol{g}$ then transmits information at a rate R nats per unit time, where

$$R = \lim_{t \to \infty} \frac{\log M^t}{E\left[\sum\limits_{i=0}^{Nt} Y_i\right]} \tag{3}$$

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 $\boldsymbol{g}$. This rate can also be written as

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

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

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

where

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

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

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

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

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,

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

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 $\boldsymbol{g}$, 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

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

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 mth 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

$$\begin{aligned}\Gamma\left(u^j|y_0, y^{Nj}\right) &= \Gamma\left(u^{m-1}|y_0, y^{N(m-1)}\right)\\ &\quad + \Gamma\left(u_m^j|\tilde{y}_{m-1}, y_{N(m-1)+1}^{Nj}\right) \qquad (8)\end{aligned}$$

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

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

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

$$C_m(\boldsymbol{g}) = E\left[C_m(\boldsymbol{g}, \boldsymbol{U}, \boldsymbol{Y})\right]$$

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

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

$D_L(\boldsymbol{g})$ 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 $\boldsymbol{g}$ and a constant $A < \infty$ such that the rate of information transfer is R nats per second where $R(1+\delta) > \mu/(2e)$, and $D_L(\boldsymbol{g}) \leq A$ 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 $\boldsymbol{g}$ 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 $\boldsymbol{G}$ on the set of infinite trees is described as follows. Q_0 is selected independent of the other branch labelings according to the distribution

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

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 $\boldsymbol{G}$ on the set of infinite trees.

Let

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

denote the average time for a packet to exit, given the input message is u^L, and the output stream is y^{NL}. Let $T_L(\boldsymbol{g}) = ET_L(\boldsymbol{g}, U^L, Y^{NL})$. Consider the random variable $T_L(\boldsymbol{G})$. 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 $L \geq 1$

$$ET_L(\boldsymbol{G}) = 1/\lambda \qquad (9)$$

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

From the argument in Section I, while finding the expected number of computations in the mth 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 $C_m(\boldsymbol{G}), m \geq 1$, are identically distributed. Recall that

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

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 $EC_1(\boldsymbol{G}) \leq K$.

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

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

Hence, with

$$T(\boldsymbol{G}) = T'(\boldsymbol{G}) + \lim_{L\to\infty}\ (Y_0(\boldsymbol{G})/(NL))$$

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

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

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

which implies that

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

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

Following the argument in [1], we then get

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

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

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

This concludes the proof of the Theorem. □

B. Expected Number of Computations Over the Tree Ensemble

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

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

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 $\boldsymbol{G}$ is such that the choice of $x_1(\hat{u}^1), \cdots, x_{Nj}(\hat{u}^j)$, is independent of the choice of $\boldsymbol{x}(\boldsymbol{u})$. Consequently, taking the expectation with respect to the choice of $x_1(\hat{u}^1), \cdots, x_{Nj}(\hat{u}^j)$, and denoting that expectation by $\hat{E}[\cdot]$ as in [2], we get

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

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:

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

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

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

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

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

where (12) follows from $(\mu' y_0)^z e^{-\mu' y_0}/z! \le 1$ 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:

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

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. □

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

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

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

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

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

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

and

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

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

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

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

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 $P(S = k) = \mu(1-\mu)^{k-1}, k \ge 1$, 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

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

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. Azizoğlu, "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 Neworks*. 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 $\boldsymbol{X}$ taking its values in $\mathbb{R}^n$ with probability density function $p(\boldsymbol{x})$ is defined by

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

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.