Practical Codes for Queueing Channels: An Algebraic, State-Space, Message-Passing Approach

Todd P. Coleman
ECE Department
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801
colemant@uiuc.edu

Negar Kiyavash
CS Department
Information Trust Institute
University of Illinois at Urbana-Champaign
Urbana, IL 61801
kiyavash@uiuc.edu

Abstract—This paper examines more closely the probabilistic dynamics of queueing timing channels and discusses a new practical coding scheme which is tailored to them and approaches capacity. We consider using sparse graph coset codes over non-binary finite fields. We use a shaping technique to map algebraic symbols to non-uniform codewords using the inverse cumulative distribution of a target random variable. We exploit the graphical structure of the conditional distribution of the departure process given the arrival process to arrive at a Forney Factor graph of the joint likelihood that has graphical structure reminiscent of coding on inter-symbol interference channels with LDPC codes. We show through simulation that this technique, when using low-complexity iterative decoding, is capacity-approaching.

I. INTRODUCTION

This paper discusses a practical implementation of coding schemes for queueing channels that approaches the capacity. Here we consider a communication channel where the encoder communicates information based upon timings between successive packets. A receiver observes packet timings after they have traveled through a communication network with queues at intermediate router nodes. Based upon the encoding mechanism, the statistical structure of the network queues, and the packet timings it observes, the receiver finds the most likely bit sequence. Anantharam & Verdu characterized - in closed form - the capacity of an instance of the problem where a single server /M/1 queue is placed between the packets at the transmitters and the packets at the receiver [1]. The characterization of capacity is nontrivial - due to queueing systems being nonstationary, nonlinear, and non-memoryless. But to date, there has been a lack of practical ways to realize the capacity limit. Indeed, the authors of [1] state this pessimism themselves in [2, Sec VI]: “Coding theory for queueing channels is virtually nonexistent.” We introduce in this paper an architecture that has low decoding complexity and can approach the capacity of communication over queueing channels. To the best of the authors’ knowledge, this is the first known such scheme with these properties.

II. PREVIOUS WORK

Sundaresan and Verdu [3] showed the existence of tree codes with sequential decoding for the exponential server timing channel that can achieve half of capacity. However, 1) they can only achieve the cutoff rate (which is in this case half of capacity) at best, and 2) such codes have infinite worst-case decoding time. More recently, Wagner and Anantharam [4] developed an elegant distance measure for point processes that was able to capture the zero-rate reliability function of the “Bits Through Queues” channel. This metric for the queue is in analogy to the Hamming or Euclidean distance for conventional channels. Wagner and Anantharam’s [4] metric was recently used as a distortion metric to prove the rate distortion function of a Poisson process that is in analogy with the proofs associated with the rate distortion functions of a Bernoulli source with Hamming distortion measure and a Gaussian source with squared-error distortion measure [5]. That being said, it is unclear how this metric leads to code constructions with practical decoding algorithms.

III. DETAILED TECHNICAL SUMMARY

We here exploit the graphical structure of the conditional distribution of the departure process \( d = (d_1, \ldots, d_n)^T \) given the arrival process \( a = (a_1, \ldots, a_n)^T \), to develop algebraic codes with low-complexity decoding algorithms and performance approaching the capacity. The methodology draws from understanding the dynamics of queueing systems [6], [7], as well as algebraic coding theory [8], [9], and message-passing on graphs [9], [10].

![Fig. 1. Conveying information through packet timings in a queueing system. Bits are encoded into the arrival times of the packet sequence originating from the source host (blue). The routers in a network are modeled as a queueing system, which introduces “noise” in the sense that the timings of the packet sequence en route to the destination host (red) will be different than those of the source packet sequence. It is the job of the destination host to decode the message from its arrival packet timings (red).](image)

For the /M/1 queue with service rate \( \mu \), the maximum rate of communication satisfies [1]

\[
C(\lambda) \geq - \lambda \log_2 \frac{\mu}{\lambda} \quad \lambda < \mu, \quad \text{(bits/s)}
\]  

\[
C \geq e^{-1} \mu \log_2 e \quad \text{(bits/s)}
\]  

(1a)  

(1b)
where the maximum corresponding to (1b) is achieved in (1a) with an input Poisson process at rate $\lambda = e^{-1}\mu$. Since on average there will be $n = \lambda T$ packet transmissions arriving in $T$ seconds, this can equivalently be stated as:

$$\tilde{C}(\lambda) \geq \log_2 \frac{\mu}{\lambda}, \quad \lambda < \mu, \quad \text{(bits/packet)} \quad (2a)$$

$$\tilde{C} \geq \log_2 e \quad \text{(bits/packet)} \quad (2b)$$

Analogously, the discrete-time equivalent for “Bits through Queues” was analyzed in [11], [12]. The maximum achievable rate for a Bernoulli $\lambda$ process is given by

$$C(\lambda) \geq h(\lambda) - \frac{1}{\mu} h(\mu) \quad \text{nats/slot} \quad (3a)$$

$$C \geq \ln [1 + \rho] \quad \text{nats/slot} \quad (3b)$$

where $h(\cdot)$ is the binary entropy function in base $e$. The timing capacity (3b) is shown to be the supremum of $\lambda$-timing capacities over $0 \leq \lambda < \mu$, and the capacity-achieving rate $\lambda^*$ and $\rho$ are given by

$$\lambda^* = \frac{\rho}{\rho + 1} \quad (4)$$

$$\rho \triangleq e^{-h(\mu)} \quad (5)$$

Since 1 bit equals ln 2 nats and on average there will be $n = \lambda T$ packets arriving in $T$ slots, this can equivalently be stated as:

$$\tilde{C}(\lambda) \geq \frac{h_2(\lambda)}{\lambda} - \frac{h_2(\mu)}{\mu}, \quad \lambda < \mu, \quad \text{(bits/packet)} \quad (6a)$$

$$\tilde{C} \geq \left[1 + \frac{1}{\rho}\right] \log_2 [1 + \rho] \quad \text{(bits/packet)} \quad (6b)$$

where $h_2(\cdot)$ is the binary entropy function in base 2 and $\rho$ is given by (5).

### A. Shaping

We now consider forcing the inter-arrival times to satisfy certain algebraic conditions. We know that for in the discrete-time case, the inter-arrival times should be i.i.d. following a geometric distribution, and for the continuous case, the inter-arrival times should be i.i.d. following an exponential distribution. So we consider doing the following. We know that if we would like to construct a random variable $Z$ with cumulative distribution function $F_Z(z)$, then we can first construct a uniform random variable $U$ on $[0, 1]$ and then construct $X$ as

$$Z = F_Z^{-1}(U). \quad (7)$$

#### a) Examples:

- **Continuous-Time:**
  For example, an exponential random variable with parameter $\lambda$ has cdf $F_Z(z) = 1 - e^{-\lambda z}$ and so we can construct an exponential from a uniform as
    $$U = F_Z(Z) = 1 - e^{-\lambda Z}$$
    $$\Rightarrow Z = \frac{-\ln(1 - U)}{\lambda} \quad (8)$$

- **Discrete-Time:**
  In discrete-time, a geometric random variable with heads probability $\lambda \in (0, 1)$ has cdf given by $F_Z(k) = 1 - (1 - \lambda)^k$ and so we can construct a geometric from a uniform as
    $$U = F_Z(Z) = 1 - (1 - \lambda)^Z$$
    $$\Rightarrow Z = \frac{\ln(1 - U)}{\ln(1 - \lambda)} \quad (9)$$

So if we can generate $n$ i.i.d. uniform $[0, 1]$ random variables, $\{U_i\}_{i=1}^n$, then we can generate the inter-arrival times $Z_i$ according to (8) or (9).

### B. Coding Algebraically with Inter-Arrival Times

So by first using the inverse CDF, we can collapse the encoding problem into constructing $n$ i.i.d. uniform $[0, 1]$ random variables. It is well known that random linear codes over finite fields suffice [8]. By shaping according to the method in the previous section, Gallager showed how using random linear coset codes over finite fields with maximum-likelihood decoding suffices to achieve capacity [8, p. 208] on arbitrary discrete memoryless channels. This has recently been shown to also be true when we specifically consider LDPC coset codes [13]. A similar approach was used in [14]. Other authors have considered using essentially the same inverse CDF idea for other continuous communication channels that require shaping [15], [16]. However, none of these methods addressed constructing a Poisson arrival process.

So we propose doing the following: consider some field size $Q = 2^s$. Then we force our $X_i$’s to lie in the finite field $\mathbb{F}_Q$. We consider a matrix $H$ with $m < n$ rows and $n$ columns defined over $\mathbb{F}_Q$. We will define our linear coset code $C$ as

$$C = \{x : \ H x = s \}$$

> From here, we interpret each $x_i \in \{0, \ldots, Q - 1\}$ as a member of $\mathbb{R}$ and define

$$U_i = \frac{X_i + \tau}{Q}. \quad (10)$$

where $\tau \in [0, 1)$ so that $\frac{\tau}{Q} \leq U_i \leq 1 - \frac{(1 - \tau)}{Q}$. Note that for the ensemble of random linear codes, the $X_i$’s will be uniformly distributed over $\{0, \ldots, Q - 1\}$ and thus the $U_i$’s will be uniformly distributed over $\{\frac{\tau}{Q}, \frac{1 + \tau}{Q}, \ldots, \frac{Q - (1 - \tau)}{Q}\}$. For large $Q$, this approximates a uniform distribution over $[0, 1]$. So altogether, we go from $X_i$ to $U_i$ via (10) and then from $U_i$ to $Z_i$ via (7) to arrive at:

$$s(x) = F_Z^{-1} \left( \frac{X_i + \tau}{Q} \right). \quad (11)$$

#### Examples:

- **Continuous-Time:**
  $$Z_i = s(X_i) \quad (12a)$$
  $$\Rightarrow Z_i = \frac{-\ln \left[ 1 - \frac{(X_i + \tau)}{Q} \right]}{\lambda} \quad (12b)$$
Since the rate for such a procedure is
\[ R = \frac{\log_2 Q^{n-m}}{n} \text{bits per packet}, \]
for an arrival Poisson process with rate \( \lambda \) and an
exponential-\( \mu \) server, we must have that
\[ R < C(\lambda) \iff \left( 1 - \frac{m}{n} \right) \log_2 Q < \log_2 \frac{\mu}{\lambda} \]
bits/packet.

- **Discrete-Time:**
\[ Z_i = s(X_i) \]
\[ = \left[ \ln \left( 1 - \frac{X_i + \gamma}{Q} \right) \right] \left( \frac{\ln(1 - \lambda)}{n} \right) \]
(13a)
\[ = \ln \left( 1 - \frac{X_i + \gamma}{Q} \right) \] \quad (13b)
Since the rate for such a procedure is
\[ R = \frac{\log_2 Q^{n-m}}{n} \text{bits per packet}, \]
for an arrival Bernoulli process with rate \( \lambda \) and a
geometric-\( \mu \) server, we must have that
\[ R < C(\lambda) \iff \left( 1 - \frac{m}{n} \right) \log_2 Q < \frac{h_2(\lambda)}{\lambda} - \frac{h_2(\mu)}{\mu} \]
bits/packet.

### C. The Arrival Process as a Simple First-Order Stochastic Dynamical System

We know that the actual arrival times of our input process satisfy
\[ a_i = a_{i-1} + z_i \]
\[ = a_{i-1} + s(x_i) \]
(14a)
(14b)
We denote the variable \( x_i \) to capture this as:
\[ \alpha_i(a_i, a_{i-1}, x_i) \triangleq 1_{\{a_i = a_{i-1} + s(x_i)\}}. \]

The formulation of the arrival process as a simple first-order stochastic dynamical system is illustrated in figure 2.

### D. The Departure Process as a Simple First-Order Stochastic Dynamical System

Now consider the departure process from the output of a
queue with a first-come, first-serve (FCFS) discipline, depicted in figure 3. Note that the service time \( s_i \) for the first packet
is given by \( s_1 = d_2 - d_1 \), because the server starts working on the second packet once the first packet departs. The second packet departs before the third arrival \( a_3 \). Thus the third service time is simply \( s_3 = d_3 - a_3 \). So in general, it follows that [6], [7]
\[ s_i = d_i - \max(a_i, d_{i-1}) \]
\[ = w(d_i, a_i, d_{i-1}) \] \quad (16)
where (16) reflects that \( s_i \) only depends on \( d_i, a_i, \) and \( d_{i-1} \).

- **Continuous-Time:**
  If we are in continuous time and the service times \( s_i \) are exponentially distributed with \( \mu \), i.e.
  \[ f_{S_i}(s) = \mu e^{-\mu s} \]
  then we have that [4]
  \[ P(d|a) = \prod_{i=1}^{n} 1_{\{d_i > a_i\}} \mu e^{-\mu w(d_i, a_i, d_{i-1})} \]
  where \( w(\cdot) \) is given by (16).

- **Discrete-Time:**
  If the service times \( S_i > 0 \) are geometrically distributed with parameter \( \mu \), i.e.
  \[ P(S_i = k) = \mu(1 - \mu)^{k-1}, \quad \text{for } k \geq 1, \]
  then we have that
  \[ P(d|a) = \prod_{i=1}^{n} 1_{\{d_i > a_i\}} \mu(1 - \mu)^{w(d_i, a_i, d_{i-1})-1} \]
  where \( w(\cdot) \) is given by (16).

Note that in both cases, we can say that
\[ P(d|a) = \prod_{i=1}^{n} \beta_i(d_i, d_{i-1}, a_i), \]
where an appropriate choice of \( \beta_i(d_i, d_{i-1}, a_i) \) for discrete-time geometric and the continuous exponential server are
\[ \beta_i(d_i, d_{i-1}, a_i) = \begin{cases} 
1_{\{d_i > a_i\}} \mu(1 - \mu)^{w(d_i, a_i, d_{i-1})-1} & \text{for DT} \\
1_{\{d_i > a_i\}} \mu e^{-\mu w(d_i, a_i, d_{i-1})} & \text{for CT} 
\end{cases} \]
(17)
Figure 4 depicts the formulation of departure process as a simple first-order stochastic dynamical system, with the arrival times as an exogenous input.
E. A State-Space Representation of the Aggregate System

Now that we have used a linear code to map input bits to code symbols $x_i$, viewed the arrival process as a simple first-order stochastic dynamical system with $x_i$ as an exogenous input, and viewed the departure process as a simple first-order stochastic dynamical system with $a_i$ as an exogenous input, we can characterize the joint likelihood of all observable and unobservable state variables:

\[
P(d, a, x) = P(d|a, x)P(a, x) = \frac{1}{|\mathcal{C}|} 1_{\mathcal{E} \in \mathcal{C}} P(d|a) \prod_{i=1}^{n} 1_{\{a_i = a_{i-1} + s(x_i)\}}
\]

\[
= \frac{1}{|\mathcal{C}|} 1_{\{H \mathcal{E} = 2\}} \prod_{i=1}^{n} \beta_i(d_i, d_{i-1}, a_i)
\]

\[
= \frac{1}{|\mathcal{C}|} 1_{\{H \mathcal{E} = 2\}} \prod_{i=1}^{n} \beta_i(d_i, d_{i-1}, a_i) \alpha_i(a_i, a_{i-1}, x_i)
\]

\[
= \frac{1}{|\mathcal{C}|} 1_{\{H \mathcal{E} = 2\}} \prod_{i=1}^{n} g_i(d_i, d_{i-1}, a_i, a_{i-1}, x_i)
\]

Now that we know the structure of the joint likelihood, we can consider finding a posteriori probabilities for decoding purposes. We would like to use the sum-product algorithm [9], [10] on the factor graph representation to get approximates of $P(x_i|d)$ to do decoding, as is done with LDPCs and Turbo Codes. Note that there are no cycles in the factor graph representing $\prod_{i=1}^{n} g_i(d_i, d_{i-1}, a_i, a_{i-1}, x_i)$. So if we have a good graphical representation for $1_{\{H \mathcal{E} = 2\}}$ of a sparse graph linear coset code (i.e. an LDPC coset code), then it is quite sensible to postulate that good approximate decoding should arise. By performing the following association,

- associate node $g_i$ with (18),
- associate node $r_{C,i}$ with $x_i \in F_Q$,
- associate node $p_{C,j}$ with $1_{\{h_i \mathcal{E} = s_j\}}$.

we arrive at the following message-passing rules which are standard direct manifestations of the sum-product algorithm.
G. Performance Results

We used this architecture and tested its performance using a $Q = 4$, $n = 250$ irregular LDPC coset code to encode messages and simulate them through a discrete-time geometric server with service rate $\mu = 0.5$. Such a system has a capacity of 1.6096 bits/packet, and a symbol-error-rate plot, shown in Figure 6, illustrates that at 82% of capacity, SER values on the order of $10^{-4}$ are achieved. Thus with these simple sparse graph codes, of short length, and without detailed optimizing the detailed structure of the sparse graph code, we were still able to achieve nontrivial and significant performance (already beyond half of capacity with low SER using a message-passing decoder). This thus illustrates a proof-of-concept of this approach, and many future directions can be explored to sharpen performance.

IV. Acknowledgements

The authors would like to thank G. David Forney, P. Vontobel, and R. Koetter for useful earlier comments and suggestions on improving the organization of this manuscript.

References