ACHIEVABLE RATES FOR QUEUE-BASED TIMING STEGOCODES

BY

IBTISSAM MOHAMMAD EZZEDDINE

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Adviser:

Professor Pierre Moulin

Abstract

This work studies stegocodes (data-hiding codes) for covert communication over timing channels. In a timing channel, the information resides in the packet interdeparture times as opposed to the packets themselves. The encoding procedure should preserve the statistics of the packet interarrival process. Our steganographic codes are based on two novel ideas, namely, queue-based codes and Shannon's encoding functions for channels with causal side information at the transmitter.

The embedding of information bits is done by a $Geo/Geo/1$ queue servicing the interarrival times. Shannon's theory of coding with causal side information at the transmitter is then used to study the queue-based stegocodes and their information-theoretic limits. Evaluating these limits for the $Geo/Geo/1$ queue is a formidable computational problem. An efficient computational approach is proposed to compute the maximum achievable rate of the queue-based stegocode and the optimal encoding functions. We also design simple practical codes, test them on a communication system while taking network noise into account, and compare their performance to the mappings under study.

To my family

Acknowledgments

I first wish to thank my adviser, Pierre Moulin, for his guidance and support. I also want to thank my family and my friends for their ongoing encouragement. Finally, this research would not have been possible without the financial support of NSF grant CCF 08-30776.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Steganography is the science of hiding messages in cover signals (*covertext*) such as image, video, audio, graphics, text or packet transmission times. The dual problem to steganography is *steganalysis*, that is, detection of hidden information within a dataset (*stegotext*). Steganography can be used to secretly transmit data over public and private networks. In this thesis, we develop new methods for covert communication over timing channels with side information, building on the work in [1]. It would be unsafe for the steganographer to ignore that his communication might be detectable by a steganalyzer. A secure strategy for the steganographer is to assume that the steganalyzer is capable of performing optimal detection tests. An example of a steganalyzer is a network administrator who is trying to discover any unauthorized use of the network.

Timing channels are covert channels in which information is embedded in the time intervals between the packets sent over a network from a transmitting node to a receiving node. A covert receiver observing packet timings decodes the embedded message; therefore, a channel is created between the stegocoder and the covert receiver. If the packet timings have the same statistics as regular packet timings, communication over the timing channel is undetectable. The timing channel can be used without altering the packet contents. The encryption of the packet contents provides no protection against the communication over the timing channel. Note that the stegocoder can use the timing channel along with the packet channel for a higher transmission rate. One instance of exploiting the packet channel is by modifying the packet headers.

Timing channels can be used, e.g., by the military and intelligence organizations to discreetly communicate over public networks. On the negative side, timing channels can be designed by a malicious intruder for the same purpose. The steganographer might be a hacker who has gained unautho-

rized access to a node on the network and tries to exfiltrate private data such as passwords and sensitive documents. In either scenario, the steganographer wants to remain undetected. Even if the network administrator cannot detect this unauthorized use of the network, he can still disrupt it by jamming the network. The packet timings could be altered to some extent by adding timing noise to the channel. This might not be desirable because normal network operations will be subject to extra latencies and transmission errors. Therefore, the actual detection of a covert channel is more efficient than jamming.

This thesis is organized as follows. In Chapter 2, we present background and a literature survey on timing channels. Chapter 3 provides an overview of the theoretical basis needed to construct the timing stegocodes. The codes are formally defined, and the complexity of practical implementation is pointed out. In Chapter 4, a feasible approach to compute the maximum achievable rates of the constructed stegocodes is adopted. Simple practical codes are designed in Chapter 5 and are tested over a timing channel while taking the network noise into account. Finally, Chapter 6 concludes the thesis.

Notation: Capital letters denote random variables, and lowercase letters represent their realizations. Boldface letters denote vectors, and calligraphic letters denote sets. For example, $\mathbf{X}^n \in \mathcal{X}^n$ represents a random vector $(X_1, \ldots, X_n)$, with each X_i taking values in $\mathcal{X}$. The probability distribution of $\mathbf{X}^n$ is denoted by $p_{\mathbf{X}^n}$, and the probability of its realization $\mathbf{x}^n$ is denoted by $p_{\mathbf{X}^n}(\mathbf{x}^n)$. The mutual information between two random variables X and Y with joint pmf p is $I_p(X;Y) = H(X) - H(X|Y)$, where $H(X)$ is the entropy of X and $H(X|Y)$ is the conditional entropy of X given Y. The Kullback-Leibler divergence between two pmf's p and q is denoted by $D(p||q)$. A matrix is denoted by a blackboard symbol (e.g, $\mathbb{A}$). The indicator function of a set $\mathcal{A}$ is denoted by $\mathbf{1}_{\mathbf{x}\in\mathcal{A}}$. The floor function $\lfloor x \rfloor$ denotes the largest integer $\leq x$. Finally, $\xrightarrow{w}$ denotes weak convergence.

Chapter 2

Timing Channels

2.1 Background

Timing channels first started with Lampson's program confinement problem [2]. The goal was to ensure that no software process can leak private data to any other process. An application of this problem arose in the 1990s in multi-level secure (MLS) systems that have multiple security clearance levels. There are HIGH and LOW levels. A HIGH can access any data available at a LOW but not vice versa. This is also known as the "No Read-Up/Write-Down" policy. In this scenario, a timing channel between the different levels can allow a software trojan horse to transfer information from HIGH to LOW [3, 4].

Timing channels exist in the following types of communication: (1) when data is transmitted over asynchronous networks, and (2) when data sources transmit data packets at somewhat irregular time instants. An important application which arises in interactive communication sessions is transmission of *keystroke packets.* In an interactive SSH session, each single keystroke typed by a user is transmitted to the remote machine in an individual IP packet immediately after the key is pressed. A trojan horse sitting in the vicinity of a network node involved in this communication can exploit the timing channel. The trojan horse can alter the inter-keystroke timings in such a way that the modulated timing sequence leaks private information. This problem is also known as *data exfiltration* [5].

Several codes have been proposed to exploit timing channels, including IPtime [6], ICMPtime [5], Keyboard Jitterbug [7], and TCPscript [8]. These codes are simple and the basic operation is to delay incoming packets in such a way that the time difference between consecutive transmitted packets is an even multiple of some δ if the hidden bit is 0, and an odd multiple of δ if the

hidden bit is 1. These simple codes are easily detectable, as will be explained later.

Research has been conducted on several information-theoretic aspects of timing channels as well. Giles and Hajek [9] studied game-theoretic aspects of communication over a timing channel with a jammer. Moulin and O'Sullivan's theory of steganography [10] applies to timing channels, as detailed by Wang and Moulin [11]. Servetto and Vetterli [12] analyzed timing channels for broadcast transmission systems. Anantharam and Verdú [13] studied the capacity of queueing systems and quantified the capacity of the underlying timing channel. There, a sender communicates bits through a queue, by controlling the packet interdeparture times.

In contrast, the problem presented in this thesis[1] is one of information transmission *with side information* in the form of *given* packet interarrival times. The stegocoder is able to perturb the packet timings but not the contents, *subject to causality and average delay constraints*. In this sense, the problem relates to Shannon's 1958 work on channel coding with causal side information available at the transmitter [15]. Moreover, there is an average cost constraint on the codewords, which takes the form of an average delay that covertext packets are subjected to.

2.2 Overview of Applications

When we talk about timing channels, two problems should be addressed. The problems are illustrated in Fig. 2.1 and Fig. 2.2. In both problems, the packet traffic is generated by a user engaged in an interactive communication session. In the first problem, the timings of the packets are generated by the steganographer, and therefore the steganographer has full control over the timing channel. In the second problem, the steganographer obtains a timing sequence and is constrained to it, which means he must adapt his transmission to an incoming packet stream. Our work only addresses the second problem, which is also more challenging.

Before the problems are presented, it is useful to give background on statistical analysis of keystrokes. Statistical analysis has shown that inter-keystroke timings follow a heavy-tailed distribution such as the Pareto dis-

[1] Part of this work was presented in our conference paper [14].

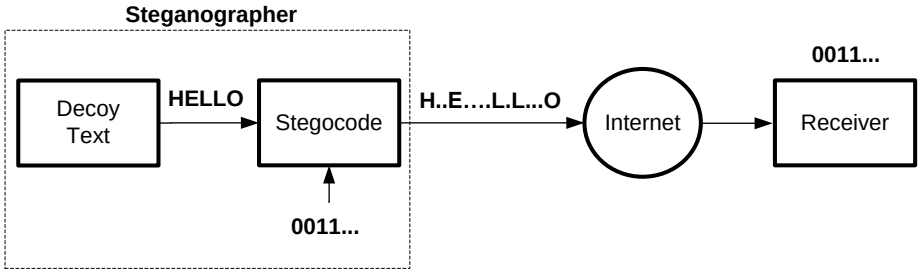

Figure 2.1: The decoy-to-stegotext problem. In this example one hidden bit is transmitted following each letter of the decoy word "HELLO."

tribution [16]. Large inter-keystroke timings are much more likely than those given by an exponential distribution. The shortest inter-keystroke timings are of the order of 30 ms, and the time it takes the operating system to send out the packet after a keystroke is generally negligible. If the packets are time-stamped, the time resolution can vary from 10 ms (in some Linux systems) to 100 ms (in Windows systems). Time stamping takes place at the application layer and has no effect on the steganographic operations, given that these operations take place below the application layer. Network transit time depends on the geographic distance and the number of hops between the sending and receiving nodes. Measurements have shown that roundtrip transit time typically ranges from a few ms (for short links) to 200 ms (for transmission across the globe) [17]. These transit times tend to fluctuate, with a typical jitter window of 10 ms [7]. These fluctuations may be thought of as noise that affects the transmission of information in the timing channel.

2.2.1 Decoy-to-stegotext generation

Referring to Fig. 2.1, the steganographer is allowed to select the packet contents as well as their interdeparture times. These interdeparture times will convey information to the receiver, analogous to a Morse code. For instance, if the timing takes two possible values, δ and 2δ, with equal probability, the steganographer could assign timing δ to bit 0, and timing 2δ to bit 1. The average time for transmission of a bit is $\frac{1}{2}(\delta + 2\delta) = \frac{3}{2}\delta$, resulting in a transmission rate of $\frac{2}{3\delta}$ bits per second of covert information. In an actual typing example, inter-keystroke timings have many more than just two values and follow a heavy-tailed distribution, e.g., a Pareto law. For the code illustrated

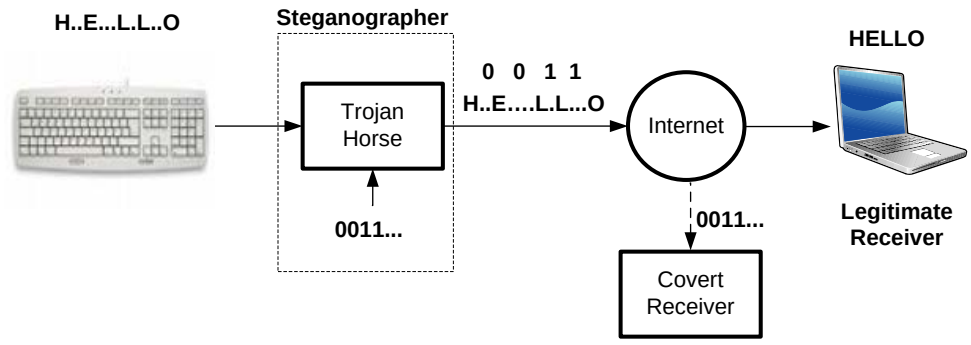

Figure 2.2: The trojan horse problem. Here an even number of time slots between two consecutive letters encodes bit 0, and an odd number encodes bit 1.

in Fig. 2.1, inter-keystroke times are represented by integer values. An even number represents bit 0, and an odd number represents bit 1. The stego-text is a one-to-one function of the decoy (packet contents) and the hidden message (packet timings).

2.2.2 Trojan horse

Referring to Fig. 2.2, the steganographer has gained unauthorized access to a host computer (by installing a Trojan horse) and is able to modify the timings of the packets transmitted by the computer. Note that the steganographer has no control over the packet contents here. More importantly, for the steganographer's operations to be discreet, the statistics of the timings should be preserved, and the average delay caused by his operations should be small. (In an interactive application, it is generally considered that the user feels the system is non-responsive if the response time exceeds 100 ms [7, 18].) The small-delay requirement is not straightforward to deal with. To see this, consider the example above where the timing takes two possible values, δ and 2δ, with equal probability, and the steganographer assigns these values to bits 0 and 1, respectively. Since the steganographer must process packets causally in time, he must introduce a large transmission delay to be able to transmit a bit sequence such as $000\ldots0$, which forces the departing packets to be transmitted in adjacent slots, while the incoming packets were not. All the techniques used in current literature [5, 6, 7, 8] suffer from this problem, and constrain the trojan horse to actively transmit only over brief periods

of time (to keep the overall delay unnoticeable). Therefore, the trojan horse fails to fully exploit the timing channel.

Interestingly, what we have called trojan horse could, in reality, be used by a "good guy." Indeed this person could use actual text that he is typing as a decoy, and have additional data embedded via the "trojan horse" algorithm.

In all cases, the steganographer does not modify, create, or delete packet contents; nor does he need to know the packet contents, which could be encrypted.

Chapter 3

Timing Stegocodes

Referring to Fig. 3.1, the covertext is modeled as a sequence of interarrival times $\mathbf{A}^n = (A_1, \ldots, A_n)$ of i.i.d. samples drawn from a pmf $\{p_A(a), a \in \mathcal{A}\}$ over a countable alphabet $\mathcal{A} \subseteq \mathbb{N}$. A message M uniformly distributed over $\mathcal{M}$ is to be embedded in $\mathbf{A}^n$ and transmitted to a decoder. The stegocoder produces a stegotext $\mathbf{D}^n$ through a function $\psi_n(\mathbf{A}^n, M)$, in order to convey the message M to the decoder reliably. The covertext and stegotext are required to be close according to some distortion metric. Moreover, a steganalyzer observing $\mathbf{D}^n$ can test whether it is drawn i.i.d. from p_A. Therefore, we restrict the divergence rate ($1/n$ the Kullback-Leibler distance) between $\mathbf{A}^n$ and $\mathbf{D}^n$ to be less than a positive number ε that can be made arbitrarily small. The decoder does not have access to the original covertext $\mathbf{A}^n$. The decoder produces an estimate $\widehat{M} = \phi_n(\mathbf{D}^n) \in \mathcal{M}$. The code (ψ_n, ϕ_n) is randomized using a random variable known only to the stegocoder and decoder. The expected latency introduced by the code at time n is $\tau_n = \mathbb{E}\sum_{j=1}^{n}(D_j - A_j)$ where the expectation is taken with respect to M and (ψ_n, ϕ_n).

Definition 1: A length-n, rate-R, latency-τ_n, ε-secure timing stegocode is defined as follows [1]:

$$D_i = f_i(\mathbf{A}^i, \mathbf{D}^{i-1}, M), \quad i = 1, 2, \ldots, n \tag{3.1}$$

$$\tau_n = \mathbb{E}\sum_{j=1}^{n}(D_j - A_j) \tag{3.2}$$

$$\frac{1}{n} D(p_{\mathbf{D}^n} || p_{\mathbf{A}^n}) \leq \varepsilon \tag{3.3}$$

where $M \in \mathcal{M} \triangleq \{1, \ldots, 2^{nR}\}$ is the secret message to be transmitted, $\mathbf{A}^i = (A_1, A_2, \ldots, A_i)$, and $f_i, 1 \leq i \leq n$, is the sequence of encoding functions. The decoder returns $\widehat{M} = \phi_n(\mathbf{D}^n)$.

Definition 2: A rate R is said to be *achievable* for a stationary, ergodic input process $\mathbf{A}^n$ if there exists a sequence of $(n, R, \tau_n, \varepsilon)$ timing stegocodes

with vanishing decoding error probability, $P_{e,n} = P_r[\widehat{M} \neq M]$ as $n \to \infty$.

Definition 3: The capacity of the timing channel is the supremum of all achievable rates.

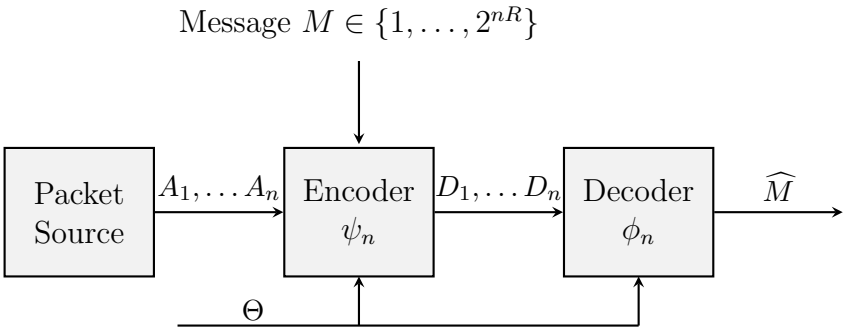

Figure 3.1: Timing channel with side information **A**.

3.1 Examples

3.1.1 Naive stegocode

Cybenko proposed a simple stochastic timing code in [5]. The code modifies each interarrival packet time A_i into an interdeparture time D_i to embed a binary bit $b_i \in \{0, 1\}$ according to the following formula:

$$D_i = \begin{cases} 2\lfloor A_i/2 \rfloor + S_i', & b_i = 0 \\ 2\lfloor A_i/2 \rfloor + S_i' + 1, & b_i = 1 \end{cases} \tag{3.4}$$

where S_i' is some odd (and possibly random) positive number. The expected value of D_i is strictly greater than the expected value of A_i, and this creates a lag with linearly increasing mean as the transmission time increases (see Fig. 3.2). This *drift* constrains the stegocoder to actively transmit only over brief periods of time, in order to keep the overall delay small. This is an inefficient exploitation of the covert channel and at the same time makes the communication easily detectable.

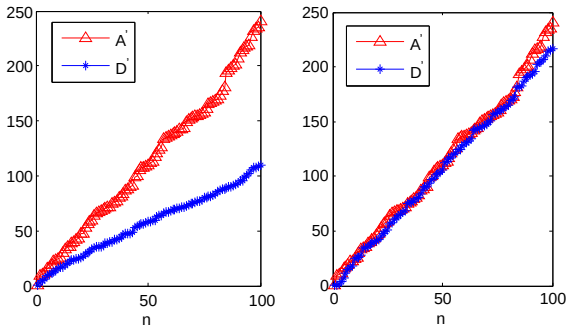

Figure 3.2: The arrival process $A'_n = \sum_{i=1}^{n} A_i$ and the departure process $D'_n = \sum_{i=1}^{n} D_i$ for (a) the simple code of (3.4), and (b) the queue-based code of (3.5).

3.1.2 Simple queue-based stegocode

To circumvent the drift problem of timing codes, a new approach was proposed in [1] which embeds the information bits via a queue (see Fig. 3.2). This code modifies the idle time W_i of a queue into an interdeparture time D_i to embed a binary bit $b_i \in \{0, 1\}$ according to the following formula:

$$D_i = \begin{cases} W_i + S'_i + 1, & W_i + b_i \quad \text{even} \\ W_i + S'_i, & W_i + b_i \quad \text{odd} \end{cases} \tag{3.5}$$

where $\{S'_i,\ i \in \mathbb{N}\}$ are i.i.d. random variables drawn from the following distribution on the set of odd integers:

$$p_{S'}(s') = \mu(2+\mu)(1-\mu)^{s'-1} \quad s' = 1, 3, 5, \ldots. \tag{3.6}$$

Referring to the definition of a timing stegocode given in (3.1), the encoding functions $f_i(\mathbf{A}^i, \mathbf{D}^{i-1}, M)$ are given by (3.5) where $W_i = \max(0, \sum_{j=1}^{i} A_j - \sum_{j=1}^{i-1} D_j)$.

Similarly to (3.4), an even interdeparture time conveys a 0 and an odd interdeparture time conveys a 1. Therefore, the message is reliably decoded by performing modulo 2 operations on the interdeparture times. Unlike (3.4) however, this code does not suffer from the drift effect. The code still has one drawback: it is not perfectly secure. The distribution of the interdeparture times is not geometric as illustrated in Fig. 3.3, and therefore the code is

still detectable. The divergence rate between $\mathbf{D}^n$ and $\mathbf{A}^n$ (referring to (3.3)) cannot be made arbitrarily small.

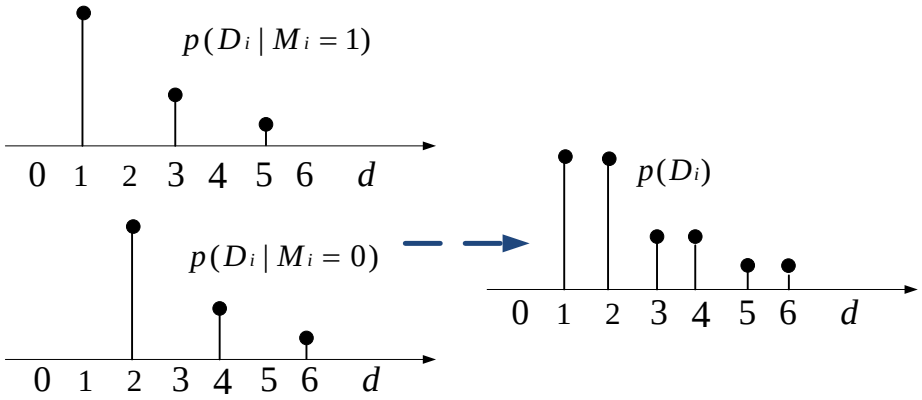

Figure 3.3: Simple queue based example.

3.1.3 Stochastic queue-based stegocode

A perfectly secure code is a code that matches the distribution of the interdeparture times to the distribution of the interarrival times and preserves the timing statistics of the channel. A simple stochastic code was proposed in [1] which still embeds the information bits via a queue but adds randomization to match the distribution of the interdeparture times to a geometric distribution. This randomization is modeled as follows:

$$D_i = \begin{cases} W_i + S_i', & : \text{ with probability } \frac{\mu}{2-\mu} \quad W_i + b_i \quad \text{even} \\ W_i + S_i' + 1, & : \text{ with probability } 1 - \frac{\mu}{2-\mu} \quad W_i + b_i \quad \text{even} \\ W_i + S_i', & : \text{ with probability } 1 \quad W_i + b_i \quad \text{odd} \end{cases} \tag{3.7}$$

where $\{S_i',\ i \in \mathbb{N}\}$ are i.i.d. random variables drawn from the following distribution on the set of odd integers:

$$p_{S'}(s') = \mu(2+\mu)(1-\mu)^{s'-1} \quad s' = 1, 3, 5, \ldots. \tag{3.8}$$

Similarly to (3.5), the encoding functions $f_i(\mathbf{A}^i, \mathbf{D}^{i-1}, M)$ are given by (3.7) where $W_i = \max(0, \sum_{j=1}^{i} A_j - \sum_{j=1}^{i-1} D_j)$.

Figure 3.4 illustrates that the code is perfectly secure in the sense that the divergence rate between $\mathbf{D}^n$ and $\mathbf{A}^n$ (referring to (3.3)) can be made

arbitrarily small. The disadvantage of this stegocode is that we can no longer reliably decode the message bits by performing modulo 2 operations on the interdeparture times. The performed randomization decreases the achievable code rate to less than 1 bit per transmission.

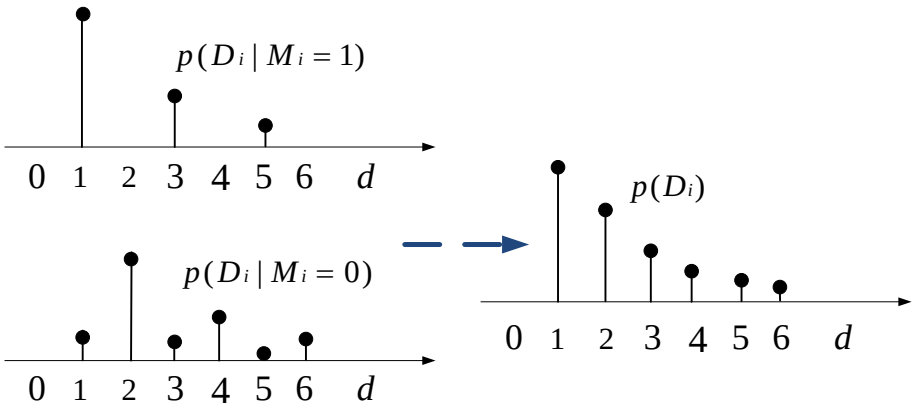

Figure 3.4: Stochastic queue based example.

3.2 Queueing Theory

A queue is a nonlinear system with memory. As illustrated in Fig. 3.5, packets with interarrival times, $\mathbf{A} = A_i, i \in \mathbb{N}$, are fed into the queue, which services them with a set of service times, $\mathbf{S} = S_i, i \in \mathbb{N}$. The interdeparture times are denoted by $\mathbf{D} = D_i, i \in \mathbb{N}$. The idle times, $\mathbf{W} = W_i, i \in \mathbb{N}$, correspond to the times when the server is not busy servicing any packet. These processes are related by Lindley's equations [19]:

$$D_i = W_i + S_i, \tag{3.9}$$

$$W_i = \left| \sum_{j=1}^{i} A_j - \sum_{j=1}^{i-1} D_j \right|^{+}, \; i = 1, 2, \ldots. \tag{3.10}$$

A special kind of queue is the $Geo/Geo/1$ queue, which has stationary i.i.d. geometric interarrival and service times (see Fig. 3.6). The $Geo/Geo/1$ queue is the discrete-time version of the classical $M/M/1$ queue with Poisson arrival process and exponential service times. Burke showed in [20] that for a queueing system with a Poisson input, single waiting line without defec-

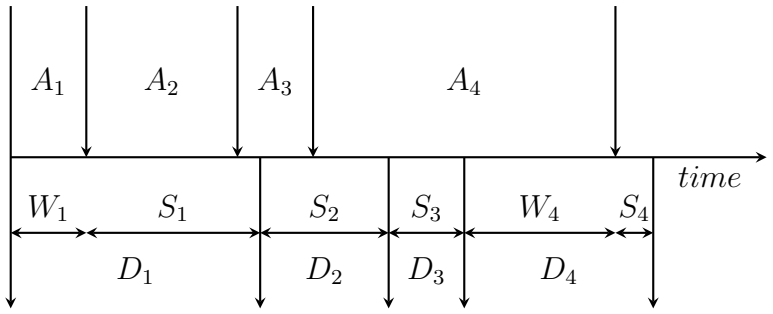

Figure 3.5: Interarrival (**A**), idle (**W**), service (**S**), and interdeparture (**D**) processes for the queue.

tions, and identically distributed independent exponential service times, the equilibrium distribution of the number of service completions in an arbitrary time interval is the same as the input distribution, for any number of servers. Burke's mathematical proof extends to the $Geo/Geo/1$ queue.

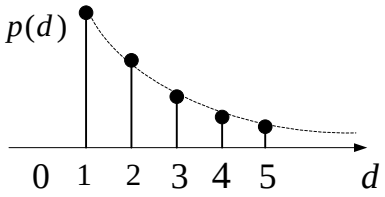

Figure 3.6: Geometric distribution.

The arrival rate λ and the service rate μ of the $Geo/Geo/1$ queue are given by

$$\lambda = \frac{1}{\mathbb{E}(A_1)}, \quad \mu = \frac{1}{\mathbb{E}(S_1)}. \tag{3.11}$$

The load factor μ/λ must be less than 1 for the queue to be stable. Then A_i and S_i are mutually independent and i.i.d. geometrically distributed with respective parameters λ and μ; D_i has the same stationary distribution as A_i; A_i, W_i, S_i, D_i are jointly stationary sequences. The marginal distributions for the service, interarrival and interdeparture times are given by the following geometric probability laws:

$$p_S(s) = (1-\mu)^{s-1}\mu, \quad s = 1, 2, \ldots \tag{3.12}$$

$$p_A(d) = p_D(d) = (1-\lambda)^{d-1}\lambda, \quad d = 1, 2, 3, \ldots \tag{3.13}$$

The marginal distribution p_W for the idle times is the mixture of an

impulse at 0 (with weight λ/μ) and a normalized geometric distribution with parameter λ. The queue introduces a mean delay:

$$\tau_n = \mathbb{E}\sum_{j=1}^{n}(D_j - A_j) \tag{3.14}$$

where $\lim_{n\to\infty}\tau_n = \frac{1}{\mu-\lambda}$.

More generally, the service could follow a renewal process (S_i is i.i.d. but not necessarily geometrically distributed) or a more general stationary ergodic process. For a queue operating in stationary mode, (A, D) is a jointly stationary process. The following properties apply:

- **Small Latency** For $\mu \gg \lambda$, we have $\tau \to 0$, $\mathbf{S} \xrightarrow{w} 0$, $\mathbf{W} \xrightarrow{w} \mathbf{A} \xrightarrow{w} \mathbf{D}$, and (**W**,**D**) $\xrightarrow{w}$ i.i.d. joint sequence, where $\xrightarrow{w}$ denotes weak convergence.
- **Large Latency** For $\mu \downarrow \lambda$, we have $\tau \to \infty$, $\mathbf{W} \xrightarrow{w} 0$, and (**W**,**D**) $\xrightarrow{w}$ i.i.d. joint sequence.

Bambos and Walrand showed in [21] that, for arbitrarily fixed statistics of service times, there exists an interarrival time distribution that induces an identical distribution on the interdeparture times. The existence of one such distribution is proved by construction. The paper considers an infinite cascade of queues with identical service time distribution p_S. The paper lets $A_i^k = D_i^{k-1}$, $i \in \mathbb{N}$, be the interarrival times for the $k-th$ queue when $k = 1, 2, \ldots$. Queue 1 is initialized with A_i^1, $i \in \mathbb{N}$, that have an arbitrary distribution, with rate λ. If an invariant distribution exists, this will be the limiting distribution for D_i^k, $i \in \mathbb{N}$ as $k \to \infty$, under the above iterative process.

3.3 Queue-Based Stegocodes

The $Geo/Geo/1$ queue is particularly relevant to this work because it preserves the distribution and the rate of the interarrival times, and the latency is determined by the load factor. This implies that if the information bits are embedded via the service times added by the queue while maintaining the stochastic properties of the $Geo/Geo/1$ queue, then the steganographic requirements can be met. In essence, this brings up the idea of perturbing the

timings of the codewords via queue-based encoding functions. A toy example for such a code was presented in [1]. Here we ask: *What is the maximum rate of reliable transmission $R_{\max}$ over this queue*? While it is reasonable to conjecture that $R_{\max}$ is the capacity in Definition 3, we do not know whether this conjecture is true. To analyze the queue model's information-theoretic limits and construct practical queue-based stegocodes, Shannon's framework for channels with causal side information at the transmitter (subject to additional steganographic constraints) is considered next.

3.4 Channels with Side Information at the Transmitter

A summary of Shannon's work on channels with side information at the transmitter [15] is provided below. Referring to Fig. 3.7, Shannon considered a discrete memoryless channel (DMC) $p_{Y|XS}$ with i.i.d. random states with pmf p_S over an alphabet $\mathcal{S}$. State information is made causally available to the transmitter.

A length n block code with M messages is a sequence of functions $\{f_i,\ 1 \leq i \leq\ n\}$:

$$x_i = f_i(s_1, \ldots, s_i; m),\ i = 1, \ldots, n,\ m = 1, \ldots, M. \tag{3.15}$$

Shannon showed that there is a one-to-one correspondence between this channel with side information and another channel without side information, which has the same output alphabet and an expanded input alphabet, $\mathcal{U}$ defined by the following theorem.

Theorem 1 *The capacity of $p_{Y|XS}$ with input alphabet $\mathcal{X}$, state alphabet $\mathcal{S}$, and state pmf p_S is equal to the capacity of the memoryless channel $p_{Y|U}$*

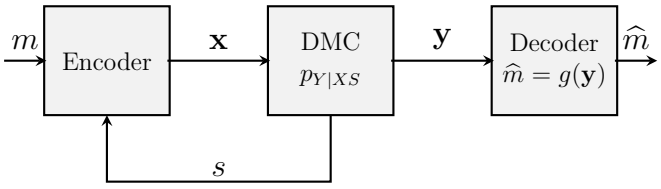

Figure 3.7: DMC with side information at the transmitter.

(without side information) with the same output alphabet $\mathcal{Y}$ *and an input alphabet* $\mathcal{U}$ *of size* $|\mathcal{U}| = |\mathcal{X}|^{|\mathcal{S}|}$*. Letting* $u = (x_1, x_2, \ldots, x_{|\mathcal{S}|}) \in \mathcal{U}$*, the transition probabilities* $p_{Y|U}$ *for the channel* $p_{Y|U}$ *are given by*

$$p_{Y|U}(y|x_1, x_2, \ldots, x_{|\mathcal{S}|}) = \sum_{s_i \in \mathcal{S}} p_S(s_i) p_{Y|XS}(y|x_i, s_i) \tag{3.16}$$

and the capacity is

$$C = \max_{p_U} I_{p_U}(U; Y). \tag{3.17}$$

A codeword for the memoryless channel $p_{Y|U}$ consists of a sequence $\mathbf{U}^n$ of n letters from $\mathcal{U}$. An input letter u corresponds to a particular function mapping the states to the input letters of the original channel. Each input letter $u \in \mathcal{U}$ maps every state $s \in \mathcal{S}$ into some $x \in \mathcal{X}$, and therefore $|\mathcal{U}| = |\mathcal{X}|^{|\mathcal{S}|}$. In other words, a mapping $u = (x_1, x_2, \ldots, x_{|\mathcal{S}|}) \in \mathcal{U}$ generates x_s when state s is observed. These mappings and the transition probabilities of (3.16) are used to prove the equivalence of the two channels. Similar mappings will be constructed next.

3.5 Information-Theoretic Analysis of Queue-Based Stegocodes

Denote by $\mathcal{W}$ and $\mathcal{D} \subseteq \mathbb{N}$ the alphabets for W and D, and consider the following framework. Let p^*_{WD} be the joint distribution of (W, D) over $\mathcal{W} \times \mathcal{D}$ for the *Geo*/*Geo*/1 queue with expected service time $1/\mu$. Denote by p^*_W and p^*_D the corresponding marginals. Denote by $\mathcal{U}$ the set of mappings from the idle time space $\mathcal{W}$ to the interdeparture time space $\mathcal{D}$ (analogous to Shannon's problem above, with U denoting the mappings, and W and D respectively playing the roles of channel state S and input X.) Let p_U be a distribution on the mappings, and $p_{D|U,W}(D|u, w)$ be the conditional probability with which a mapping $u \in \mathcal{U}$ maps an idle time $w \in \mathcal{W}$ into an interdeparture time $D \in \mathcal{D}$. These two distributions are subject to the steganographic constraints given by

$$p^*_{WD}(w, d) = p^*_W(w) \sum_{u \in \mathcal{U}} p_U(u) p_{D|U,W}(d|u, w), \quad \forall d \in \mathcal{D},\ w \in \mathcal{W}, \tag{3.18}$$

i.e., the resulting marginal on (W, D) matches p^*_{WD} exactly.

A deterministic mapping would map each $w \in \mathcal{W}$ into a single $d \in \mathcal{D}$ with probability 1. Therefore, the conditional pmf $p_{D|U,W}$ corresponding to a deterministic mapping consists of zeroes and ones such that $p_{D|U,W}(d|u, w) = \mathbf{1}_{d=u(w)}$. Consequently, the steganographic constraints given in (3.18) are given by

$$p^*_{WD}(w, d) = p^*_W(w) \sum_{u \in \mathcal{U}} p_U(u) \mathbf{1}_{d=u(w)}, \quad \forall d \in \mathcal{D},\ w \in \mathcal{W}. \tag{3.19}$$

A stochastic mapping maps each $w \in \mathcal{W}$ into some $d \in \mathcal{D}$ according to a conditional pmf, $p_{D|U,W}$. The steganographic constraints are given in (3.18). We have a countable infinite set of deterministic mappings that can be approximated by a finite set (to be discussed later). On the other hand, we have a continuum of stochastic mappings out of which a small subset suffices to meet the constraints of (3.18). Unfortunately, finding a set of stochastic mappings that achieve a relatively high rate, while meeting the steganographic requirements, is an unresolved problem. For the deterministic case, a much larger number of mappings will be needed to meet the constraints of (3.19). This thesis will mainly consider the class of deterministic mappings.

Given a message m, a packet i, and the current waiting time W_i, the stegocoder selects the mapping $u_{m,i}$ and outputs $D_i = u_{m_i}(W_i)$. Now let $\mathcal{C}$ be a randomized code, where the letters U_{M_i} are drawn i.i.d. from a pmf p_U. There are $|\mathcal{U}| = |\mathcal{D}|^{|\mathcal{W}|}$ deterministic mappings and $|\mathcal{W}||\mathcal{D}|$ matching constraints in (3.19). Denote by $\mathcal{P}_U(p_{WD})$ the set of feasible probability distributions over $\mathcal{U}$. We seek p_U that maximizes $I(U; D)$ over $\mathcal{P}_U(p_{WD})$ and satisfies the steganographic constraints of (3.19).

The ideal geometric distributions p_S and p_A given in (3.12) and (3.13) have unbounded supports. To evaluate $R_{\max}$, we consider a sequence of problems (indexed by $t \in \mathbb{N}$) involving perturbed truncated and normalized distributions. The pmf's p_S and p_A are truncated at a point t, i.e. the interarrival and service processes follow truncated geometric distributions. The corresponding distributions are denoted by $p^{(t)}_A$, $p^{(t)}_S$, $p^{(t)}_U$, $p^{*(t)}_D$, and $p^{*(t)}_{WD}$. The corresponding alphabets are given by $\mathcal{A}^{(t)} = \mathcal{S}^{(t)} = \{1, 2, \ldots, t\}$, $\mathcal{W}^{(t)} = \{0, 2, \ldots, t-1\}$, and $\mathcal{D}^{(t)} = \{1, 2, \ldots, 2t-1\}$. The alphabet for the mappings is $\mathcal{U}^{(t)} \triangleq (\mathcal{D}^{(t)})^{\mathcal{W}^{(t)}}$, and its cardinality $(2t-1)^t$ is exponential in t. The interarrival process $\mathbf{A}^{(t)}$ for the perturbed problem is i.i.d. with marginal

pmf $p_A^{(t)}$ over $\mathcal{A}^{(t)}$. Clearly $\lim_{t\to\infty} D(p_A^{(t)}||p_A) = 0$. The following equalities hold:

$$W_i^{(t)} = \left|\sum_{j=1}^{i} A_j^{(t)} - \sum_{j=1}^{i-1} D_j^{(t)}\right|^+, \; i = 1, 2, \ldots \tag{3.20}$$

$$D_i^{(t)} = U_{m,i}^{(t)}(W_i^{(t)}). \tag{3.21}$$

The maximum achievable rate $R_{\max}$ of the queue-based stegocode with causal side information is

$$R_{\max} = \sup_{t\geq 2} R_t \tag{3.22}$$

where

$$R_t = \max_{p_U^{(t)} \in \mathcal{P}_U^{(t)}(p_{WD}^{*(t)})} I_{p_U^{(t)}}(U^{(t)}; D^{(t)}) \tag{3.23}$$

and $\mathcal{P}_{U^{(t)}}(p_{WD}^{*(t)})$ is the feasible set of probabilities over $\mathcal{U}^{(t)}$, defined analogously to (3.19):

$$\mathcal{P}_{U^{(t)}}(p_{WD}^{*(t)}) = \{p_U^{(t)} : p_{WD}^{*(t)}(w,d) = p_W^{(t)}(w) \sum_{u\in\mathcal{U}^{(t)}} p_U^{(t)}(u)\mathbf{1}_{d=u(w)}\}. \tag{3.24}$$

Consequently

$$\sum_{w\in\mathcal{W}^{(t)}} p_W^{*(t)}(w) \sum_{u\in\mathcal{U}^{(t)}} p_U^{(t)}(u)\mathbf{1}_{\mathbf{d=u(w)}} = p_D^{*(t)}(d) \; \forall d \in \mathcal{D}^{(t)}. \tag{3.25}$$

We can show that R_t is increasing in t and is bounded from above by the entropy $H(A^{(t)})$; therefore, R_t is finite and the supremum in (3.22) is equal to the limit. $H(A^{(\infty)})$ is the entropy of an ideal geometric distribution. The matching constraints in (3.19) are represented in matrix form as

$$\mathbb{A}p_U^{(t)} = \mathbf{b} \tag{3.26}$$

where $\mathbb{A}$ is a $t(2t-1) \times (2t-1)^t$ binary matrix. Its columns correspond to the mappings, and the rows correspond to the constraints (the structure of the matrix will be explained later). The $t(2t-1)$ dimensional vector $\mathbf{b}$ corresponds to the right-hand side of the equalities given in (3.19).

3.6 Toy Stegocodes Revisited

Before moving on to the information-theoretic analysis of stegocodes, it is worth revisiting the simple stegocodes presented in (3.4) and (3.7) and relating them to particular mappings. Moreover, these simple stegocodes will illustrate some advantages and disadvantages of stochastic and deterministic mappings. Each stegocode considers two mappings: one for the case when $W_i + b_i$ is even, and one for the case when $W_i + b_i$ is odd.

In the toy stegocodes, at each time instance i, a random variable S'_i is generated. The stegocodes define two functions that map the idle time W_i and the message bit b_i into an interdeparture time D_i by adding the random service time S'_i or $S'_i + 1$ to W_i. The simple stegocode of (3.5) defines two stochastic mappings u_1 and u_2 with the following conditional distributions:

$$p(d|u_1, w) = p_{S'}(d - w - 1)$$
$$p(d|u_2, w) = p_{S'}(d - w)$$

where $p_{S'}$ is given in (3.6). Mapping u_1 is picked when $W_i + b_i$ is even and u_2 is picked when $W_i + b_i$ is odd. This code does not match the interdeparture times' distribution to a geometric distribution. The stochastic queue-based code of (3.7) using the same mapping u_2 but a different stochastic mapping u_1 can match the interarrival and interdeparture times' distributions. These mappings are governed by the following conditional distributions:

$$\begin{aligned} p(d|u_1, w) &= p_{S'}(d - w - 1) \times (1 - \frac{\mu}{2 - \mu}) + p_{S'}(d - w) \times \frac{\mu}{2 - \mu} \\ p(d|u_2, w) &= p_{S'}(d - w) \end{aligned}$$

where $p_{S'}$ is given in (3.6). Using only two stochastic mappings, we are able to meet the steganographic requirements but achieve a rate lower than 1. We cannot construct a secure code consisting of only two deterministic mappings. On the other hand, if a larger set of mappings can be accounted for, a higher rate can be achieved by using deterministic mappings.

3.7 Matrix Representation of Steganographic Constraints

The conditional distribution of the interdeparture times, given the idle times and the deterministic set of mappings, can be represented by a $t(2t-1) \times (2t-1)^t$ binary matrix $\mathbb{A}$. Its columns correspond to the mappings, and the rows correspond to the steganographic constraints. Equation (3.26) is a matrix representation of the equations given in (3.25). Each column of $\mathbb{A}$ corresponds to the probabilities with which a mapping maps the idle times into interdeparture times. The first $2t-1$ entries of a column represent the probabilities with which the mapping maps idle time 0 into the $2t-1$ interdeparture times respectively. The second $2t-1$ entries of a column represent the probabilities with which the mapping maps idle time 1 into the $2t-1$ interdeparture times respectively, and so on. Every $2t-1$ entries corresponding to an idle time add up to 1. With the causality condition a mapping can only map an idle time into a strictly greater interdeparture time (the queue service time is strictly greater than 1). Therefore, the probability of mapping idle time 1 into interdeparture time 1 is 0. Therefore, $\mathbb{A}$ has rows $2t$, $4t-1$, $4t, \ldots$ bound to 0. For a deterministic mapping, every idle time is mapped into a single interdeparture time with probability 1. This is represented in a column of the matrix $\mathbb{A}$ by an entry equal to 1 in every $2t-1$ entries corresponding to an idle time and the remaining entries equal to 0.

Matrix (3.27) shows an example of a matrix made up of 3 stochastic mappings for $t = 2$. Matrix (3.28) shows the matrix corresponding to the whole set of deterministic mappings for the simple case when $t = 2$. For $t = 2$, idle time 0 and 1 are mapped into the interdeparture times 1, 2, or 3.

$$\mathbf{A} = \begin{bmatrix} 0.1 & 0.25 & 0.3 \\ 0.4 & 0.2 & 0.5 \\ 0.5 & 0.55 & 0.2 \\ 0 & 0 & 0 \\ 0.9 & 0.75 & 0.4 \\ 0.1 & 0.25 & 0.6 \end{bmatrix} \tag{3.27}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{3.28}$$

The conditional probability distribution of a stochastic mapping belongs to the convex hull of the conditional distributions of the deterministic mappings. Shannon showed in [22] that any input letter interior to the convex hull defined by the other input letters can be deleted without affecting channel capacity. Therefore stochastic mappings can be discarded when the whole family of deterministic mappings is considered. Consequently, if a set of stochastic mappings meets the steganographic constraints then there exists a corresponding set of deterministic mappings also meeting the steganographic constraints while achieving an equal or higher rate. Therefore, only the family of deterministic mappings will be studied.

3.8 Dimensionality of the Optimization Problem

Evaluating (3.23) is apparently a simple problem since it involves maximizing a concave function over a convex set defined by $|\mathcal{W}^{(t)}||\mathcal{D}^{(t)}|$ linear equalities. The difficulty of this optimization problem lies in the high dimension of the input and constraint spaces. There are of the order t^t deterministic mappings and t^2 constraints. Even for moderately large t, the size of $\mathcal{U}^{(t)}$ is astronomical. As $t \to \infty$, the support set of $W^{(t)}$ and $D^{(t)}$ is unbounded and $p_W^{*\infty}$ and $p_D^{*\infty}$ are the ideal geometric distributions.

The size of the support set of $p_U^{(t)}$ that achieves R_t is at most $|\mathcal{D}^{(t)}| + |\mathcal{D}^{(t)}||\mathcal{W}^{(t)}| = 2t^2 + t - 1$. Therefore, only a small subset of the entire set $\mathcal{U}^{(t)}$ is needed to construct encoding functions achieving R_t. Also, note that since t is finite, it is not possible to perfectly match the interdeparture distribution, $p_D^{*(t)}$ to the interarrival distribution, $p_A^{(t)}$. The truncated processes $\mathbf{A}^{(t)}$ and $\mathbf{D}^{(t)}$ are jointly i.i.d. only in the limit as $t \to \infty$; therefore, $\frac{1}{n} D(p_{\mathbf{D}^n}^{*(t)} || p_{\mathbf{A}^n}^{(t)}) \leq D(p_D^{*(t)} || p_A^{(t)}) \triangleq \varepsilon^{(t)}$. The timing stegocode in (3.1) is $\varepsilon^{(t)}$-secure. As $t \to \infty$,

$\varepsilon^{(t)} \to 0$ and $R_t = \max_{p_U^{(t)}} I_{p_U^{(t)}}(U^{(t)}; D^{(t)}) \to R_{\max}$.

Chapter 4

Computation of Achievable Rates

Evaluation of (3.22) involves the solution of a sequence of convex programs indexed by t. Apparently an analytic solution cannot be found. Standard optimization algorithms are computationally expensive and fail for large alphabet sizes. Blahut [23] proposed a computationally feasible iterative algorithm to compute the capacity of an unconstrained channel. He also generalized this algorithm to channels with two constraints using a Lagrangian method [23]. A closed-form solution for the Lagrangian parameter was given, and the complexity of the algorithm with cost constraints is essentially the same as that of the basic algorithm without cost constraints (see appendix). To run Blahut's algorithm with constraints, the vector of multipliers needs to be known. For the simple cases of one or two constraints, the multipliers are easily obtained. Unfortunately, we have $t(2t-1)$ multipliers corresponding to the constraints of (3.26). Computing these multipliers appears to be a formidable task even for a moderately large t, so a computationally efficient method is needed. We first give necessary background on linear programming.

4.1 Linear Programming Background

A linear programming problem may be defined as the problem of *maximizing or minimizing a linear function subject to linear constraints.* The constraints may be equalities or inequalities. The function to be maximized (or minimized) is called the *objective function.* In general, the objective function, being linear, always takes on its maximum (or minimum) value at a corner point of the constraint set, provided the constraint set is bounded. Occasionally, the maximum occurs along an entire edge or face of the constraint set, but then the maximum occurs at a corner point as well.

Not all linear programming problems are so easily solved. There may be many variables and many constraints. In general, we are given an m-vector, $\mathbf{b}= (b_1, \ldots, b_m)^T$, an n-vector, $\mathbf{c}= (c_1, \ldots, c_n)^T$, and an $m\times\ n$ matrix,

$$\mathbb{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ldots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \tag{4.1}$$

of real numbers. Define the following Linear Program:

$$\max_{\mathbf{x}} \mathbf{c^T x} = c_1x_1 + \ldots + c_nx_n \tag{4.2}$$

over an n-vector, $\mathbf{x}= (x_1, \ldots, x_n)^T$, subject to the constraints

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \vdots \qquad & \qquad (or \quad \mathbb{A}\mathbf{x}=\mathbf{b}) \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned}$$

and $\mathbf{x}\geq 0$.

To every linear program (primal LP) there is a dual linear program (dual LP) with which it is intimately connected. We will state the duality for the linear program defined in (4.2). As previously stated, $\mathbf{c}$ and $\mathbf{x}$ are n-vectors, $\mathbf{b}$ and $\mathbf{y}$ are m-vectors, and $\mathbb{A}$ is an $m\times\ n$ matrix. The dual of the Linear Program defined in (4.2) finds an m-vector, $\mathbf{y}= (y_1, \ldots, y_n)^T$, to minimize

$$\max_{\mathbf{y}} \mathbf{y^T b} = b_1y_1 + \ldots + b_my_m \tag{4.3}$$

subject to the constraints

$$\mathbf{y^T}\mathbb{A} \geq \mathbf{c^T}.$$

A Linear Program and its dual are connected by the following two theorems.

Theorem 2 ***Strong Duality Theorem*** *If a linear programming problem has an optimal solution, so does its dual, and the respective optimal costs are equal. This means that given* $\boldsymbol{x^*}$*, the optimal solution of the Linear Program,*

then $\boldsymbol{y^*}$*, the optimal solution of the dual, exists. Moreover,*

$$\mathbf{y^{*^T}b} = \mathbf{c^T x^*}.$$

Theorem 3 ***Complementary Slackness*** *Let* $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *be feasible solutions to the primal and the dual programs, respectively. The vectors* $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *are optimal solutions for the two respective programs iff*

$$\begin{aligned} y_i(\mathbf{a_i^T x} - b_i) = 0, \quad i = 1, \ldots, m \\ (c_j - \mathbf{y^T A_j})x_j = 0, \quad j = 1, \ldots, n \end{aligned}$$

where a_i *and* A_j *are the row and column vectors of the matrix* $\mathbb{A}$*, respectively.*

4.2 Our Linear Program

4.2.1 Formulating the primal

For notational convenience we denote by $I(p_U^{(t)})$ the mutual information between $U^{(t)}$ and $D^{(t)}$. The functional is concave in $p_U^{(t)}$ over the $(2t-1)^t$ dimensional probability simplex. The feasibility set $\mathcal{P}_{U^{(t)}}(p_{WD}^{*(t)})$ given by (3.26) is a linear subspace of $\mathbb{R}^{(2t-1)^t}$. Observe that $I(p_U^{(t)})$ is linear in $p_U^{(t)}$ over $\mathcal{P}_{U^{(t)}}(p_{WD}^{*(t)})$. Indeed

$$I(p_U^{(t)}) = \mathbf{c_U}' p_U^{(t)}$$

where

$$\begin{aligned} \mathbf{c_U}(u) &= \sum_{d \in \mathcal{D}^{(t)}} p_{D|U}^{(t)}(d|u) \log \frac{p_{D|U}^{(t)}(d|u)}{p_D^{*(t)}(d)} \\ p_{D|U}^{(t)}(d|u) &= \sum_{w \in \mathcal{W}^{(t)}} p_W^{(t)}(w) \mathbf{1_d} = \mathbf{u(w)}. \end{aligned}$$

We note that the $(2t-1)^t$ dimensional vector, $\mathbf{c_U}$, is independent of $p_U^{(t)}$ because $p_D^{*(t)}$ and $p_{D|U}^{(t)}$ are independent of $p_U^{(t)}$. Consequently, the computation

of R_t in (3.23) can be reduced to a linear program of the form:

$$\begin{aligned} &\max_{p_U^{(t)}} \mathbf{c_U}' p_U^{(t)} \\ &s.t \quad \mathbb{A} p_U^{(t)} = \mathbf{b} \end{aligned} \tag{4.4}$$

where $\mathbb{A}$ and $\mathbf{b}$ are defined in (3.26). Note that the constraint, $\sum_{u \in \mathcal{U}^{(t)}} p_U^{(t)}(u) = 1$, is included in (3.26).

4.2.2 Formulating the dual

This reformulation of the optimization problem does not reduce the complexity of (3.23) but presents us with a useful dual problem. In fact, the Lagrangian multipliers needed to run Blahut's algorithm with multiple constraints constitute the dual vector of the linear program in (4.4). The dual of the linear program in (4.4) is an optimization in a $t(2t-1)$ dimensional space. The complexity of the optimization is reduced from working in a space of a dimension exponential in t to a space of a dimension quadratic in t. Let $\mathbf{x}$ be the dual vector or the vector of Lagrangian multipliers needed in Blahut's algorithm. The dual program is of the form

$$\begin{aligned} &\min \mathbf{b}'\mathbf{x} \\ &s.t \quad \mathbb{A}'\mathbf{x} \geq \mathbf{c_U}. \end{aligned} \tag{4.5}$$

4.2.3 From the dual to the primal

After solving the dual in (4.5), the optimal vector $\mathbf{x}^*$ is obtained. Two approaches can be taken to find the solution of the original problem. First, the resulting vector $\mathbf{x}^*$ constitutes the vector of multipliers in Blahut's iterative algorithm. This results in R_t and $p_U^{*(t)}$ over $\mathcal{U}^{(t)}$. Another way of solving the primal in (4.4) from the dual in (4.5) is by making use of complementary slackness. The vector $\mathbb{A}'\mathbf{x}^*$ is evaluated and compared to $\mathbf{b}$. The slack inequalities correspond to zero entries in the primal solution. Fortunately, a large subset of the constraints are slack since the subset of mappings that achieve R_t consists of no more than $2t^2 + t - 1$ mappings. The primal in (4.4) is easily solvable with this reduced subset, and $p_U^{*(t)}$ that achieves R_t is

obtained.

4.3 Numerical Results

Table 4.1 gives R_t of (3.23) for fixed service and arrival rates, $\mu = 0.5$ and $\lambda = 0.3$, and increasing values of the truncation point, t. The larger t, the higher the maximum achievable rate R_t, and the higher the computational complexity of the problem. Recall that a linear increase in t leads to an exponential increase in the space of mappings $\mathcal{U}^{(t)}$. Fortunately R_t converges fairly rapidly, as seen in Table 4.1.

To study the tradeoff between achievable rate and latency, we varied the arrival and service rates and computed R_7. Rate curves for fixed arrival rates were plotted in Fig. 4.1 against the expected normalized delay, $\overline{\tau} = \lambda/(\mu - \lambda) = 1/(\mu/\lambda - 1)$. The achievable rates decrease with an increase in λ, as does the entropy of the interarrival process. Referring to Table 4.2 and Fig. 4.1, we see that the rate curves saturate and converge to the upper bound $H(A^{(t)})$ in the limit as $\overline{\tau} \to \infty$. Moreover $H(A^{(t)})$ is itself upper-bounded by the entropy $H(A^{(\infty)})$ of the ideal i.i.d. geometric process $\mathbf{A}$. The stegocoder has no control over the arrival process and its rate. On the other hand, the service rate of the queue is a design parameter that controls the latency.

Table 4.1: R_t of our timing stegocode for $\lambda = 0.3$, $\mu = 0.5$ and for different values of the truncation point t.

t	R_t	$\lvert\mathcal{U}^{(t)}\rvert$
2	0.8360	9
3	1.1983	125
4	1.3756	2401
5	1.4614	59049
6	1.4724	1771561
7	1.4812	62748517

Table 4.2: Comparison of R_7 for $\overline{\tau} = 35$ to $H(A^{(7)})$ for different values of λ.

λ	R_7	$H(A^{(7)})$	$H(A^{(\infty)})$
0.3	2.4736	2.4904	2.9376
0.5	1.9192	1.9336	2
0.7	1.2516	1.256	1.259

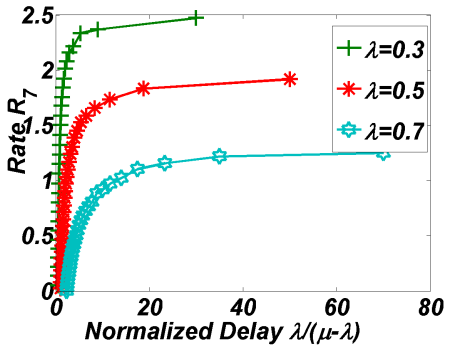

Figure 4.1: Rate curves for increasing λ plotted against the normalized delay, $\overline{\tau} = \lambda/(\mu - \lambda) = 1/(\mu/\lambda - 1)$, for $t = 7$.

Chapter 5

Practical Timing Stegocodes

It is not an easy task to find near-capacity achieving practical codes. In this section we propose a simple practical code and compare its rate to the achievable rates computed in the previous section. We also test this code over a real network. A real network perturbs the timings and adds noise to the channel. Therefore we need to consider an appropriate noise model and design a decoder. The building blocks of the communication system are depicted in Fig. 5.1 and detailed below. Coding schemes involving a pseudo-random interleaver (to mask the dependencies between the bits) known to the stegocoder and decoder need to be used to achieve reliable, and undetectable communication.

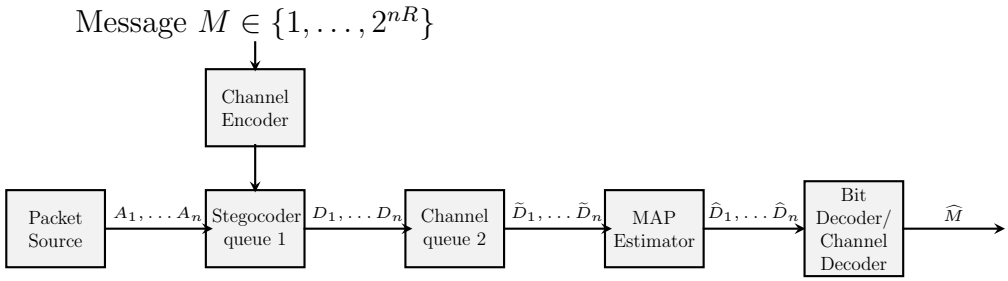

Figure 5.1: Communication model of a timing channel.

5.1 The Encoder

The stegocoder maps the stream of interarrival times $\mathbf{A}^n$ and the length-n message M into a stream of interdeparture times $\mathbf{D}^n$. The stegocoder uses a code with a layered structure. The bits are first mapped to a Q-array alphabet, then the Q-array message and the input stream are mapped into interdeparture times while preserving the interarrival times' statistics (we are

still working under a $Geo/Geo/1$ queue setup to ensure the steganographic constraints). We will first extend the toy example of Equation (3.5) to a Q-array alphabet; then we will present how a Q-array encoder can be used on a binary input.

5.1.1 Q-array encoders

The simple toy stegocode of (3.5) can be extended to a Q-array alphabet using modulo Q operations on the idle times and the message bits. Assume we want to embed a ternary alphabet message consisting of the set of symbols or bits $b_i \in \{1, 2, 3\}$ instead of a binary message in the packets' interarrival times. The encoding is based on the following formula:

$$D_i = \begin{cases} W_i + S'_i + 2, & (2W_i + b_i) \bmod 3 = 0 \\ W_i + S'_i, & (2W_i + b_i) \bmod 3 = 1 \\ W_i + S'_i + 1, & (2W_i + b_i) \bmod 3 = 2 \end{cases} \tag{5.1}$$

where W_i is the i^{th} idle time and $\{S'_i\}$ are i.i.d. random variables following a geometric distribution given by

$$p_{S'}(s') = \frac{\mu(1-\mu)^{s'-1}}{\mu/(1-(1-\mu)^3)} = (1-(1-\mu)^3)(1-\mu)^{s'-1} \quad s' = 1, 4, 7 \ldots. \tag{5.2}$$

The i^{th} service time of the queue is $S_i \triangleq D_i - W_i$.

It is easy to verify that the resulting interdeparture times satisfy $d_i \bmod 3 = b_i \bmod 3$. Therefore, we obtain the correct ternary message bits by performing modulo 3 operations on the observed interdeparture times. If the ternary bit takes on the values $1, 2$, and 3 with equal probabilities, then the service time S_i takes on the values S'_i, $S'_i + 1$, and $S'_i + 2$ with probability $1/3$. Therefore for $S_i \bmod 3 = 1$, the probability of S_i is the probability that $S'_i = S_i$ times the probability that $S_i \bmod 3 = 1$, which is $1/3$. Similarly for $S_i \bmod 3 = 2$, the probability of S_i is the probability that $S'_i = S_i - 1$ times the probability that $S_i \bmod 3 = 2$, which is $1/3$. Finally for $S_i \bmod 3 = 0$, the probability of S_i is the probability that $S'_i = S_i - 2$ times the probability that $S_i \bmod 3 = 0$, which is $1/3$. Consequently, the service time distribution

of our queue is not geometric and is given by the following formula:

$$\begin{aligned} p_S(s) &= \begin{cases} \frac{(1-(1-\mu)^3)(1-\mu)^{s-1}}{3(1-\mu)^2}, & s \bmod 3 = 0 \\ \frac{(1-(1-\mu)^3)(1-\mu)^{s-1}}{3}, & s \bmod 3 = 1 \\ \frac{(1-(1-\mu)^3)(1-\mu)^{s-1}}{3(1-\mu)}, & s \bmod 3 = 2 \end{cases} \\ &= \frac{(1-(1-\mu)^3)(1-\mu)^{s-1}}{3(1-\mu)^{(s-1) \bmod 3}}. \end{aligned} \tag{5.3}$$

This method can be extended in a similar way to to a larger input alphabet. Suppose the encoder wants to embed a message consisting of bits $b_i \in \{1, 2, \ldots, Q\}$ in the input stream. Similarly to (5.1), the encoding scheme is extended to the following:

$$D_i = \begin{cases} W_i + S_i' + Q - 1, & ((Q-1)W_i + b_i) \bmod Q = 0 \\ W_i + S_i', & ((Q-1)W_i + b_i) \bmod Q = 1 \\ \quad \vdots & \quad \vdots \\ W_i + S_i' + Q - 3, & ((Q-1)W_i + b_i) \bmod Q = Q-2 \\ W_i + S_i' + Q - 2, & ((Q-1)W_i + b_i) \bmod Q = Q-1 \end{cases} \tag{5.4}$$

where $S_i' \bmod Q = 1$ and $\{S_i'\}$ are i.i.d. random variables following a geometric distribution given by

$$p_{S'}(s') = \frac{\mu(1-\mu)^{s'-1}}{\mu/(1-(1-\mu)^Q)} = (1-(1-\mu)^Q)(1-\mu)^{s'-1} \quad s' = 1, Q+1, 2Q+1 \ldots.$$

The i^{th} service time of the queue is $S_i \triangleq D_i - W_i$.

It follows that $D_i \bmod Q = b_i \bmod Q$. Therefore, we obtain the correct Q-array message bits by performing modulo Q operations on the observed interdeparture times. Note that the larger the Q, the larger the mismatch between the service time distribution and the geometric distribution given that the bits are equiprobable. Similarly to the ternary alphabet case, for a general Q the service time distribution of our queue is given by the following

formula:

$$
\begin{aligned}
p_S(s) &= \begin{cases}
\frac{(1-(1-\mu)^Q)(1-\mu)^{s-1}}{Q(1-\mu)^{Q-1}}, & s \bmod Q = 0 \\
\frac{(1-(1-\mu)^Q)(1-\mu)^{s-1}}{Q}, & s \bmod Q = 1 \\
\frac{(1-(1-\mu)^Q)(1-\mu)^{s-1}}{Q(1-\mu)}, & s \bmod Q = 2 \\
\quad\vdots & \quad\vdots \\
\frac{(1-(1-\mu)^Q)(1-\mu)^{s-1}}{Q(1-\mu)^{Q-2}}, & s \bmod Q = Q-1
\end{cases} \\
&= \frac{(1-(1-\mu)^Q)(1-\mu)^{s-1}}{Q(1-\mu)^{(s-1) \bmod Q}}.
\end{aligned}
$$

5.1.2 A practical code using Q-array encoders

We need a practical encoder mapping the message consisting of zeroes and ones to interdeparture times. The simple toy stegocode of (3.5) transmits the message reliably at a rate of one bit per transmission, but the interarrival times' statistics are not preserved and in fact, for high service rates, the distribution mismatch is considerable and easily detectable. On the other hand, the stochastic toy stegocode of (3.7) takes care of this problem and matches the interarrival and interdeparture times' distributions but the rate of the code is lower than 1. For a service rate of μ, the rate is at most $1 - \frac{\mu}{2(2-\mu)}$. As an illustration, for a service rate $\mu = 0.5$, the rate is at most 0.83 bits per transmission.

The distribution mismatch in the toy example is one result of assigning the equally likely bits 0 and 1 to the even and odd supports of the geometric distribution which do not each add up to 0.5. For a geometric distribution with rate μ, the probability of an even integer is $\frac{1-\mu}{2}$ and that of an odd integer is $\frac{1+\mu}{2}$. For instance, for a rate $\mu = 0.5$, the ratio of odd to even is 2 to 1. To get around this problem we can take one of two approaches. The first is to design good probability shaping codes that generate zeroes and ones matching the even and odd probabilities of a geometric distribution. The second and more feasible approach is to map the zeroes and ones to supports of the geometric distribution with almost equal probabilities. This is possible only if the service rate is less than 0.5. This can be done using the previously discussed Q-array encoders.

Bit 1 is first mapped to the set consisting of the integers $\{1, Q+2, Q+$

$3, \ldots, 2Q\}$ according to a normalized geometric distribution supporting these integers. Similarly, bit 0 is mapped to a number between 2 and $Q+1$. Then, using a $2Q$-array encoder, bit 1 corresponds to the interdeparture times $\{1, Q+2, \ldots, 2Q, 2Q+1, 3Q+2, \ldots, 4Q \ldots\}$, and bit 0 corresponds to the rest. Q is chosen such that bits 1 and 0 are mapped onto two distinct supports of the geometric distribution with almost equal probabilities.

Figure 5.2 depicts a geometric distribution that has been divided into two supports of almost equal probabilities. The first support consists of the integers $\{1, Q+2, Q+3, \ldots, 2Q, 2Q+1, 3Q+2, \ldots, 4Q, \ldots\}$ and the second support consists of the integers $\{2, \ldots, Q+1, 2Q+2, \ldots, 3Q+1, \ldots\}$. Solving for Q is done by equating the probabilities of these two sets:

$$\mu + \sum_{i=Q+2}^{2Q} \mu(1-\mu)^{i-1} = \sum_{i=2}^{Q+1} \mu(1-\mu)^{i-1}. \tag{5.5}$$

An integer solution for equation (5.5) generally does not exist. For a service rate μ larger than 0.5, it is not possible to equate any two supports of the geometric distribution because the probability of 1 is μ. For a service rate less than 0.5, solving (5.5) generally does not yield an integer solution, and therefore we should consider solving for the nearest integer estimate solution. This approximation will result in a mismatch between the interdeparture times' distribution and the interarrival times' distribution. To ensure perfect security we can use a stochastic encoder similar to the one presented in (3.7) but adapted to a Q-array encoder. This stochastic encoder achieves a rate higher than that of (3.7).

5.1.3 Rate evaluation of the Q-array stegocode

The practical code of section 5.1.2 is not perfectly secure in the sense that it does not match the distribution of the interarrival and interdeparture times. Nonetheless, it serves as a good approximation. The maximum achievable rate of mappings satisfying the steganographic constraints was computed in Chapter 4 while assuming that the network does not perturb the generated interdeparture times, and therefore the timing channel is "noiseless." Hence, we can only evaluate the performance of our $2Q$-array code as compared to the optimal rate-achieving mappings before applying our noise model that

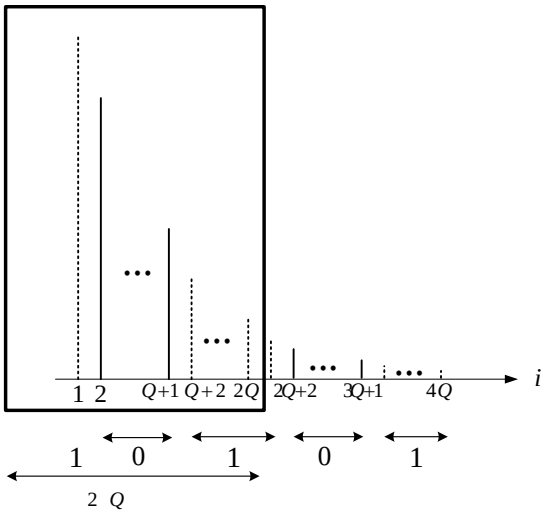

Figure 5.2: Geometric distribution divided into two distinct supports of equal probabilities, each corresponding to the bits 0 and 1.

will be discussed next. The practical code achieves a reliable rate of one bit per transmission. If we disregard the delay constraint and tolerate a large latency, the maximum achievable rate is 2.47 bits per transmission for an arrival rate $\lambda = 0.3$, as can be seen in Table 4.2. For our experiment, we have used an arrival rate $\lambda = 0.3$ and a service rate $\mu = 0.4$. Working under this setup, the maximum achievable rate is 1.53. Therefore, our practical code achieves 65% of the maximum achievable rate.

5.2 Noisy Channel

A network perturbs the timings, for example, by buffering some packets at routers due to congestion. A reasonable way to model the network noise that alters the timings between the packets is by using a queue. Anantharam and Verdú [13] characterized this model and gave lower bounds on the capacity of the queue which is the source of noise in continuous time. Gallager and Prabhakar [24], and Bedekar and Azizoglu [25] studied this channel in discrete time. These papers show that analogous to Gaussian noise, among queues with a fixed service rate (mean), queues with exponential (in continuous time) or geometric (in discrete time) service times have the lowest capacity. Moreover, for queues with geometric service times, the capacity-achieving interarrival times are also geometric. Therefore, for a given expected ser-

vice time, queues with geometric service times are considered the noisiest (introduce the worst case network noise).

At the output of the stegocoder, we have a stream of i.i.d. geometric interdeparture times $\mathbf{D}^n$ with mean $1/\lambda$. These timings are corrupted by noise modeled by a second $Geo/Geo/1$ queue (represents a source of noise) with service rate μ_N (the lower μ_N the more noisy the channel is). Therefore, our model of the network noise consists of a queue with geometric service times.

5.3 The Decoder

The decoder maps the stream of noisy interdeparture times $\widetilde{\mathbf{D}}^n$ into an estimated message $\widehat{M}$. The decoder is assumed to know the noise model and the service rate μ_N as well as the encoding function and the probability model of the interdeparture times $\mathbf{D}^n$. The ideal MAP decoder outputs

$$\widehat{M} = \arg\max_{M \in \mathcal{M}} p(M|\widetilde{\mathbf{D}}^n).$$

This decoder is very complex, although the stegocoder is simple. The complexity stems from the first queue (the stegocoder) and not from the channel encoder or the noisy queue. The first queue maps the current and past interarrival times (the idle times) and the current bit into an interdeparture time that belongs to a support set of the geometric distribution. The encoding is relatively simple but decoding is an involved task. Therefore, we implemented a practical two-stage decoder. The first stage consists of a MAP estimator that estimates the noiseless interdeparture times $\widehat{\mathbf{D}}^n$. Then, because a Q-array encoder was used to generate the interdeparture time $\mathbf{D}^n$, a suitable modulo operator is designed to map $\widehat{\mathbf{D}}^n$ into a sequence of binary bits that are finally mapped to a message $\widehat{M} \in \mathcal{M}$ using the appropriate channel decoder.

5.3.1 The MAP estimator

The MAP estimator takes as an input the stream of noisy interdeparture times $\widetilde{\mathbf{D}}^n = \{\widetilde{D}_2, \widetilde{D}_2, \ldots \widetilde{D}_n\}$ that have been subject to geometric service

times with mean $1/\mu_N$. The estimator outputs a stream of interdeparture times $\widehat{\mathbf{D}}^{n^*} = \{\widehat{D}_1^*, \widehat{D}_2^*, \ldots \widehat{D}_n^*\}$. The ideal MAP estimator outputs the sequence of interdeparture times that maximize the a posteriori probability $p(\widehat{\mathbf{D}}^n|\widetilde{\mathbf{D}}^n)$ over the feasible set of interdeparture times $\mathcal{F}^n \triangleq \{\widehat{\mathbf{D}}^n : \widehat{D}_i \in \mathcal{F}_i(\widetilde{\mathbf{D}}^i, \widehat{\mathbf{D}}^{i-1})\}$ where $\mathcal{F}_i(\widetilde{\mathbf{D}}^i, \widehat{\mathbf{D}}^{i-1}) \triangleq \{1, \ldots, \sum_{j=1}^{i} \widetilde{D}_j - \sum_{j=1}^{i-1} \widehat{D}_j\}$:

$$\begin{aligned}
\widehat{\mathbf{D}}^{n^*} &= \underset{\widehat{\mathbf{D}}^n \in \mathcal{F}^n}{\arg\max}\ p(\widehat{\mathbf{D}}^n|\widetilde{\mathbf{D}}^n) \\
&= \underset{\widehat{\mathbf{D}}^n \in \mathcal{F}^n}{\arg\max}\ \frac{p(\widehat{\mathbf{D}}^n)p(\widetilde{\mathbf{D}}^n|\widehat{\mathbf{D}}^n)}{p(\widetilde{\mathbf{D}}^n)} \\
&= \underset{\widehat{\mathbf{D}}^n \in \mathcal{F}^n}{\arg\max}\ p(\widehat{\mathbf{D}}^n)p(\widetilde{\mathbf{D}}^n|\widehat{\mathbf{D}}^n) \\
&\overset{(a)}{=} \underset{\widehat{\mathbf{D}}^n \in \mathcal{F}^n}{\arg\max}\ \prod_{i=1}^{n} p(\widehat{D}_i)p(\widetilde{D}_i|\widehat{\mathbf{D}}^i, \widetilde{\mathbf{D}}^{i-1}) \\
&\overset{(b)}{=} \underset{\widehat{\mathbf{D}}^n \in \mathcal{F}^n}{\arg\max}\ \prod_{i=1}^{n} p(\widehat{D}_i)p_{S_N}(\widetilde{D}_i - W_i)
\end{aligned}$$

where $W_i = \max(0, \sum_{j=1}^{i} \widehat{D}_j - \sum_{j=1}^{i-1} \widetilde{D}_j)$, and p_{S_N} is the noise distribution given by a geometric distribution with mean $1/\mu_N$. The equalities (a) and (b) follow from the properties of the $Geo/Geo/1$ queue. Equality (a) follows from the statistical independence of the interdeparture times $\widehat{\mathbf{D}}^n$ and the chain rule that also uses the fact that the current noisy interdeparture time $\widetilde{D}_i$ only depends on the current and past state of the system (causality condition). The second equality (b) follows from Lindley's queue equations presented in (3.10).

The size of the support set of $\widehat{D}_i$ $(\sum_{j=1}^{i} \widetilde{D}_j - \sum_{j=1}^{i-1} \widehat{D}_j)$ is large for relatively low service rates μ_N (noisy system). Fortunately, it is unnecessary to search over the whole space $\mathcal{F}^n$ of feasible interdeparture times $\widehat{\mathbf{D}}^n$. By exploiting the dependencies between the processes of the queue, we are able to construct a hidden Markov model and design a globally optimal Viterbi decoder to estimate $\widehat{\mathbf{D}}^n$. The Viterbi decoder admits a special trellis structure. The structure is that of a simple iterative step-by-step decoder.

The hidden Markov model

Figure 5.3 depicts our hidden Markov model with the hidden states and the observations. To understand the model we need to take a closer look at the processes of a queue. Our channel takes at its input the stream of noiseless interdeparture times $\mathbf{D}^n$ and outputs the stream of noisy interdeparture times $\widetilde{\mathbf{D}}^n$. Let $\widetilde{W}_i'$ and $\widetilde{W}_i$ be the waiting time and the idle time respectively of the i^{th} packet in the noisy queue. $\widetilde{W}_{i+1}'$ and $\widetilde{W}_{i+1}$ are given by the following recursion in [19]:

$$\widetilde{W}_{i+1}' = \max(0, \widetilde{W}_i' + \widetilde{S}_i - D_{i+1}) \quad (5.6)$$

$$\widetilde{W}_{i+1} = \max(0, -\widetilde{W}_i' - \widetilde{S}_i + D_{i+1}) \quad (5.7)$$

where $\widetilde{S}_i$ is the service time of the i^{th} packet in the noisy queue.

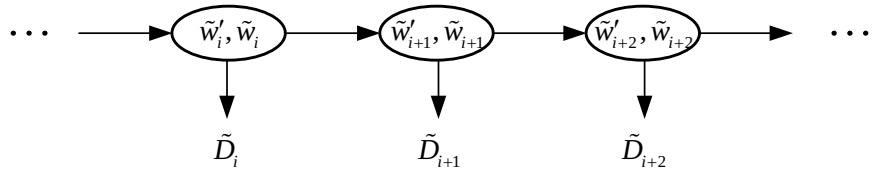

Figure 5.3: The hidden Markov model with the states and the observations.

We define the i^{th} state of the system to be $H_i \triangleq (\widetilde{W}_i', \widetilde{W}_i)$. The recursions of (5.6) and (5.7) show that state H_{i+1} is independent of the past states $\mathbf{H}^{i-1}$ given H_i. The observations consist of the noisy interdeparture times $\widetilde{\mathbf{D}}^n$. The i^{th} observation $\widetilde{D}_i$ depends only on H_i, and the output probabilities are given by

$$p(\widetilde{D}_i|H_i) = p(\widetilde{D}_i|\widetilde{W}_i) = p_{\widetilde{S}_i}(\widetilde{D}_i - \widetilde{W}_i).$$

To complete the hidden Markov model, we still need to compute the state transition probabilities. For a given state $H_i = (\widetilde{W}_i', \widetilde{W}_i)$ and an observation $\widetilde{D}_i$, the service time of the noisy queue is $\widetilde{S}_i = \widetilde{D}_i - \widetilde{W}_i$. Not all possible states are feasible given the observations and the recursions of (5.6) and (5.7) that govern the relationship between the idle and waiting times of a queue. The feasible set of states at time i is $\mathcal{H}_i(\widetilde{\mathbf{D}}^i) \triangleq \{(\widetilde{w}_i', 0), (0, \widetilde{w}_i) : \widetilde{w}_i' \in \{0, 1 \ldots, \sum_{j=1}^{i-1} \widetilde{D}_j - i\},\ \widetilde{w}_i \in \{0, 1 \ldots, \widetilde{D}_i - 1\}\}$. Let $h_i = (\widetilde{w}_i', \widetilde{w}_i) \in \mathcal{H}_i(\widetilde{\mathbf{D}}^i)$; then using the recursions of (5.6) and (5.7), the transition probabilities of

the hidden Markov model are given by

$$\begin{aligned}
p(h_{i+1}=(0,0)|h_i=(\widetilde{w}'_i,\widetilde{w}_i),\widetilde{D}_i) &= p_{D_{i+1}}(\widetilde{D}_i-\widetilde{w}_i+\widetilde{w}'_i)\\
&= p_{D_{i+1}}(\widetilde{S}_i+\widetilde{w}'_i)\\
p(h_{i+1}=(0,c)|h_i=(\widetilde{w}'_i,\widetilde{w}_i),\widetilde{D}_i) &= p_{D_{i+1}}(\widetilde{D}_i-\widetilde{w}_i+\widetilde{w}'_i+c)\\
&= p_{D_{i+1}}(\widetilde{S}_i+\widetilde{w}'_i+c)\\
p(h_{i+1}=(c,0)|h_i=(\widetilde{w}'_i,\widetilde{w}_i),\widetilde{D}_i) &= p_{D_{i+1}}(\widetilde{D}_i-\widetilde{w}_i+\widetilde{w}'_i-c)\\
&= p_{D_{i+1}}(\widetilde{S}_i+\widetilde{w}'_i-c)
\end{aligned}$$

where c corresponds to $\widetilde{w}'_{i+1}$ or $\widetilde{w}_{i+1}$, such that $h_{i+1}\in\mathcal{H}_{i+1}(\widetilde{\mathbf{D}}^{i+1})$.

The Viterbi decoder and the iterative estimator

The Viterbi decoder is optimal and outputs the most probable sequence of states $\mathbf{h}^n$ given the observed noisy interdepartures $\widetilde{\mathbf{D}}^n$. We did not estimate $\widehat{\mathbf{D}}^n$, but the states $\mathbf{h^n}$ and $\widetilde{\mathbf{D}}^n$ are sufficient to compute $\widehat{\mathbf{D}}^n$. Figure 5.4 shows the first four stages of the Viterbi decoder and the surviving paths. The most probable state at each stage is boxed. The surviving path to the most probable state at time $i+1$ stems from the most probable state at time i which is the result of the optimization up to time i ($\mathbf{H}^{i^*}=\arg\max p(\mathbf{H}^n|\widetilde{\mathbf{D}}^n)$). Therefore, the Viterbi decoder boils down to an iterative stage-by-stage decoder. In other words, the Viterbi decoder is equivalent to an iterative estimation of $\widehat{\mathbf{D}}^n$.

We estimate $\widehat{D}_1$ then use this estimate to estimate $\widehat{D}_2$ and so on. We incur no loss in optimality by using iterative estimation. We first estimate $\widehat{D}_1$ as follows:

$$\begin{aligned}
\widehat{D}_1^* &= \underset{\widehat{D}_1\in\mathcal{F}_1(\widetilde{D}_1)}{\arg\max}\ p(\widehat{D}_1|\widetilde{D}_1)\\
&= \underset{\widehat{D}_1\in\mathcal{F}_1(\widetilde{D}_1)}{\arg\max}\ \frac{p(\widehat{D}_1)p(\widetilde{D}_1|\widehat{D}_1)}{p(\widetilde{D}_1)}\\
&= \underset{\widehat{D}_1\in\mathcal{F}_1(\widetilde{D}_1)}{\arg\max}\ p(\widehat{D}_1)p_{S_N}(\widetilde{D}_1-\widetilde{W}_1)
\end{aligned}$$

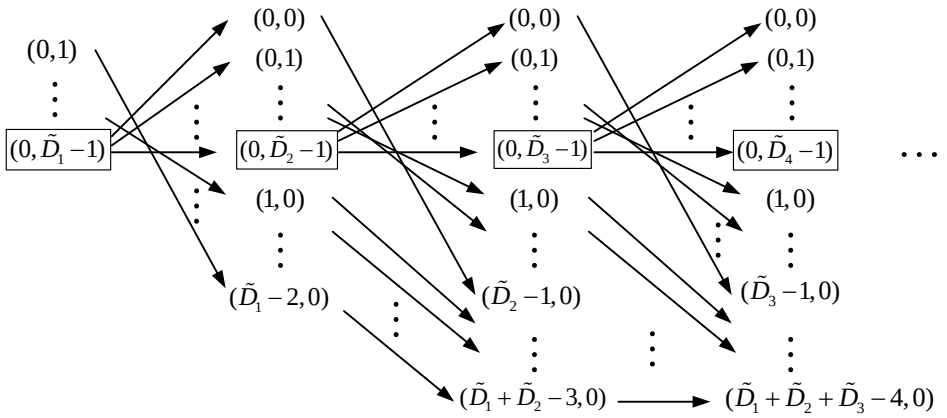

Figure 5.4: The first four stages and the surviving paths of the Viterbi decoder.

where $\widetilde{W}_1 = \widehat{D}_1$. Then $\widehat{D}_2$ is estimated as follows:

$$\begin{aligned}
\widehat{D}_2^* &= \underset{\widehat{D}_2 \in \mathcal{F}_2(\widetilde{\mathbf{D}}^2, \widehat{D}_1^*)}{\arg\max} \; p(\widehat{D}_2 | \widetilde{D}_1, \widetilde{D}_2, \widehat{D}_1^*) \\
&= \underset{\widehat{D}_2 \in \mathcal{F}_2(\widetilde{\mathbf{D}}^2, \widehat{D}_1^*)}{\arg\max} \; \frac{p(\widetilde{D}_2 | \widehat{D}_1^*, \widehat{D}_2, \widetilde{D}_1) p(\widehat{D}_2 | \widehat{D}_1^*, \widetilde{D}_1) p(\widehat{D}_1^*, \widetilde{D}_1)}{p(\widetilde{D}_1, \widetilde{D}_2, \widehat{D}_1^*)} \\
&= \underset{\widehat{D}_2 \in \mathcal{F}_2(\widetilde{\mathbf{D}}^2, \widehat{D}_1^*)}{\arg\max} \; p(\widehat{D}_2) p_{S_N}(\widetilde{D}_2 - \widetilde{W}_2)
\end{aligned}$$

where $\widetilde{W}_2 = \max(0, \widehat{D}_1^* + \widehat{D}_2 - \widetilde{D}_1)$. In a similar iterative fashion $\widehat{D}_i$ is estimated as follows:

$$\begin{aligned}
\widehat{D}_i^* &= \underset{\widehat{D}_i \in \mathcal{F}_i(\widetilde{\mathbf{D}}^i, \widehat{\mathbf{D}}^{i-1^*})}{\arg\max} \; p(\widehat{D}_i | \widetilde{\mathbf{D}}^i, \widehat{\mathbf{D}}^{i-1^*}) \\
&= \underset{\widehat{D}_i \in \mathcal{F}_i(\widetilde{\mathbf{D}}^i, \widehat{\mathbf{D}}^{i-1^*})}{\arg\max} \; \frac{p(\widetilde{D}_i | \widetilde{\mathbf{D}}^{i-1}, \widehat{\mathbf{D}}^{i-1^*}, \widehat{D}_i) p(\widehat{D}_i | \widetilde{\mathbf{D}}^{i-1}, \widehat{\mathbf{D}}^{i-1^*}) p(\widetilde{\mathbf{D}}^{i-1}, \widehat{\mathbf{D}}^{i-1^*})}{p(\widetilde{\mathbf{D}}^i, \widehat{\mathbf{D}}^{i-1^*})} \\
&= \underset{\widehat{D}_i \in \mathcal{F}_i(\widetilde{\mathbf{D}}^i, \widehat{\mathbf{D}}^{i-1^*})}{\arg\max} \; p(\widehat{D}_i) p_{S_N}(\widetilde{D}_i - \widetilde{W}_i)
\end{aligned}$$

where $\widetilde{W}_i = \max(0, \sum_{j=1}^{i-1} \widehat{D}_j^* + \widehat{D}_i - \sum_{j=1}^{i-1} \widetilde{D}_j)$. The complexity of this iterative estimator is quadratic in n. This is in fact optimal and no improvement is achieved by using the ideal MAP estimator due to the hidden Markov model of the queue.

5.3.2 The bit decoder

The final block in our communication system is the bit decoder that takes as an input the stream of estimated interdeparture times $\widehat{\mathbf{D}}^{n^*}$ and performs a bit by bit decoding on each interdeparture time $\widehat{D}_i$ as follows:

$$b_i \;=\; \begin{cases} 1, & \widehat{D}_i^* \bmod 2Q = 0, 1, Q+2, \ldots, 2Q-1 \\ 0, & \text{otherwise.} \end{cases}$$

The bit decoder can be followed by a deinterleaver and a channel decoder if channel encoding and interleaving were applied on the bits before being embedded in the stream of interarrival times. It is intuitive that for a higher service rate μ_N (less noisy) we can achieve higher rates of reliable bit transmission (smaller bit error probability). We do not have control over Q because it is obtained by solving (5.2), but for a larger Q the bits 0 and 1 are more protected against errors. This follows from the encoding scheme that assigns every Q consecutive integers to bit 0 or bit 1 (disregarding the integer 1). Therefore, for a larger Q it is less likely to estimate a D_i in a set of Q consecutive integers by a $\widehat{D}_i$ that belongs to a different set and consequently flipping the bit.

5.4 Simulation Results

We set up a timing channel between three servers and studied the performance of our practical code for various parameter values. The stegocoder embeds a message M of fixed length $n = 10^6$. The arrival rate λ is fixed and equal to 0.3. Machine A generates n packets by sampling the timings from a geometric distribution with mean $1/\lambda$ and releasing the packets accordingly. It sends them to machine B which hosts the decoder of the communication system. Machine C represents the stegocoder. It intercepts the incoming packets and embeds the message using the encoding scheme. We observed the distribution of the incoming and outgoing timing stream from machine C and decoded the message at machine B.

We compared the interarrival and interdeparture times (at the input and output of machine C) by plotting their histograms. A mismatch in the empirical distributions is observed. We used a service rate $\mu = 0.4$ and compared

the plots obtained from the Q-array encoder and the simple toy stegocode. Figure 5.5 plots the empirical distributions of the interarrival and interdeparture times for $Q = 3$ (integer estimate solution of equation (5.5)). On the other hand, Fig. 5.6 plots the distribution for $Q = 1$ or, in other words, the distribution for the even-odd simple encoder. For $Q = 1$ the distribution mismatch is more visible.

To numerically evaluate the distribution mismatch, we computed the divergence rates based on the obtained histograms. It is hard to accurately estimate the divergence because it is hard to obtain an accurate estimate of the low probabilities. These faulty low probability terms can blow up the divergence. Therefore, we only computed the divergence on a truncated support set. Upon truncation at the integer value 20, the Kullback-Leibler distance between the interdeparture and interarrival times' distributions $D(p_A||p_D)$ for $Q = 1$ equals 0.0832 while for $Q = 3$ equals 0.0351. We also computed the L_1-distance

$$L_1(p_A, p_D) = \sum_{i=1}^{n} |p_A(i) - p_D(i)|. \tag{5.8}$$

For $Q = 1$, we have $L_1(p_A, p_D) = 0.183$, and for $Q = 3$, we have $L_1(p_A, p_D) = 0.136$.

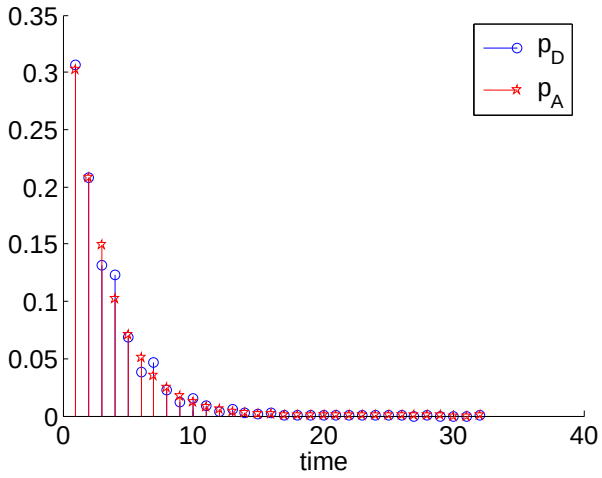

Figure 5.5: Distributions p_A and p_D for $\lambda = 0.3$, $\mu = 0.4$ and $Q = 3$.

For the channel encoder block, we used a simple repetition code (parameterized by a rate R) with random interleaving of the bits. The corresponding channel decoder is a simple majority voter. The model is complete now and we estimated different error probabilities as follows.

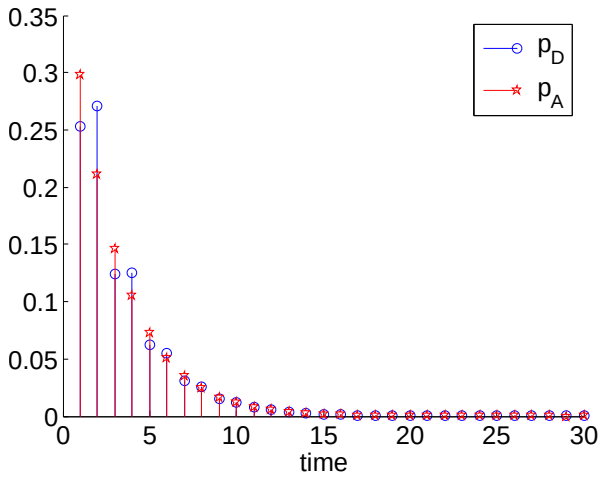

Figure 5.6: Distributions p_A and p_D for $\lambda = 0.3$, $\mu = 0.4$ and $Q = 1$.

Our stegocoder has a special structure mapping bits 0 and 1 onto two distinct sets of positive integers. Moreover, disregarding the integer 1, every set of consecutive Q integers corresponds to bit 0 or 1. In a geometric distribution, the probability of 1 is the highest and equals μ (the rate). Therefore, bit 0 is protected more against errors than bit 1. We are therefore interested in the bit error probability pe as well as the error probability of bit 0, pe_0, and bit 1, pe_1. We computed the error probabilities for different service rates μ_N of the noisy channel and repetition codes. Figure 5.7 shows the error probabilities as a function of service rate μ_N and code rate R for $\lambda = 0.3$, $\mu = 0.4$ and $Q = 3$ (integer estimate solution of equation (5.5)).

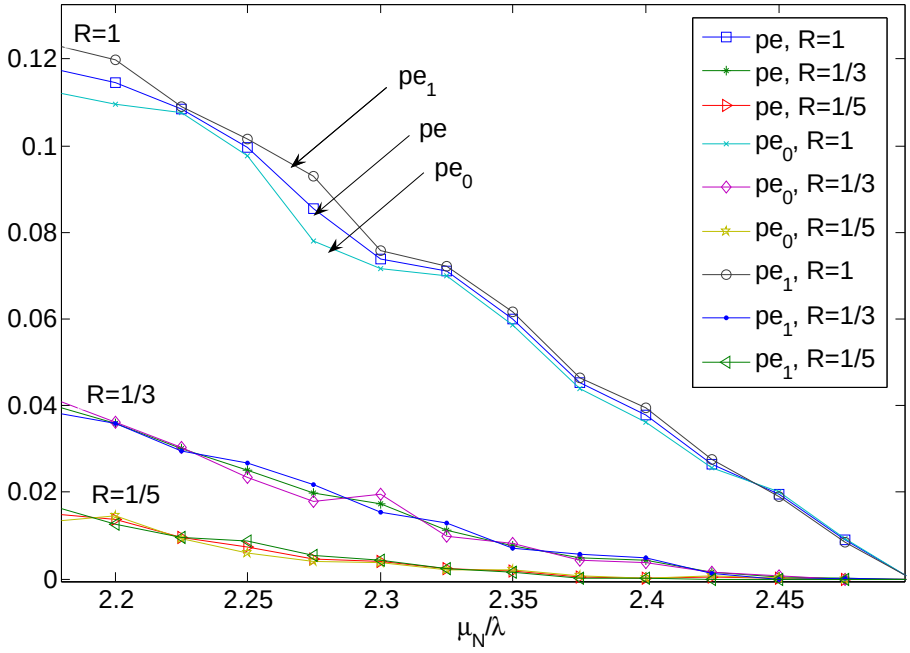

Figure 5.7: Bit error probabilities, pe, pe_0 and pe_1 for different service rates μ_N of the noisy channel and repetition codes. The stegocode's parameters are $\lambda = 0.3$, $\mu = 0.4$ and $Q = 3$.

Chapter 6

Conclusion

In this thesis, we have studied codes for timing channels with causality, latency, and steganographic constraints. Shannon's encoding functions for channels with causal side information at the transmitter were used to analyze the information-theoretic limits of the queue-based timing stegocodes. Computing the maximum achievable rate of the code is a computationally complex task. The adopted computational approach consisted of structuring a linear program and solving its dual. Blahut's iterative algorithm for finding the capacity of channels with multiple constraints was used to find the optimal input distribution and the maximum achievable rate of the timing stegocodes.

Then we set up a network simulation. To do that we developed simple practical codes and tested them over a timing channel in a network consisting of three servers. We modeled the network noise and used a practical decoder to account for the noise. Finally, we ran the experiment on the servers and monitored the performance of the practical code in terms of rate (how close they are to the mappings), probability of error and security (meeting the steganographic constraints).

Appendix A

Computation of the Capacity of a Channel with Multiple Constraints

Theorem 4 *The capacity of a constrained channel is given by*

$$\begin{aligned} C(\mathbf{E}) &= \max_{p_X \in \mathscr{P}_{\mathbf{X}}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p_X(x) p_{Y|X}(y|x) \log \frac{p_{X|Y}(x|y)}{p_X(x)} \\ \mathscr{P}_{\mathbf{X}} &= \{p_X : \mathbb{A} p_X = \mathbf{E}\}. \end{aligned}$$

In order to account for the constraints in the objective function, Blahut used a vector of multipliers. The vector of multipliers is fixed to a constant vector $\mathbf{s}$. The system is parameterized by $\mathbf{s}$. For a different vector $\mathbf{s}$, a different cost vector $\mathbf{E}$ is obtained.

Definition: The average cost vector for using a given input distribution p_X is given by

$$\mathbf{E}(p_X) = \mathbb{A} p_X.$$

Theorem 5 *Suppose E is achievable i.e. there exists a capacity achieving p_X^* s.t $\mathbb{A} p_x^* = E$. Then $C(E)$ can be expressed parameterically in terms of a vector $\mathbf{s} \in \mathbb{R}^{m+}$ by*

$$\begin{aligned} C(\mathbf{E_s}) &= \mathbf{s}' \mathbf{E}_s + V_s \\ \mathbf{E_s} &= \mathbb{A} p_X^* \end{aligned}$$

where

$$V_s = \max_{p_X} \sum_{x,y} p_X(x) p_{Y|X}(y|x) \log \frac{p_{Y|X}(y|x)}{\sum_{x'} p_X(x') p_{Y|X}(y|x')} - \mathbf{s}' \mathbb{A} p_X$$

and p_X^* achieves this maximum.

Blahut computed the channel capacity using an iterative algorithm:

- At each iteration, for each $x \in \mathcal{X}$ compute:

$$c(x) = \exp \quad \sum_y p_{Y|X}(y|x) \log \frac{p_{Y|X}(y|x)}{\sum_{x'} p_X(x') p_{Y|X}(y|x')} - \mathbf{s}' \mathbb{A} p_X.$$

- Then if $p_X^{(0)}$ has all components strictly positive, the sequence of probability vectors defined by

$$p_X(x)^{(r+1)} = p_X(x)^{(r)} \frac{c(x)^{(r)}}{\sum_{x'} p_X(x')^{(r)} c(x')^{(r)}}$$

is such that

$$I(p_X^{(r)}, Q_{X|Y}^{(r)}) \to C(\mathbf{E_s}),\ r \to \infty$$
$$\mathbf{E}(p_X^{(r)}) \to \mathbf{E_s},\ r \to \infty.$$

References

[1] P. Moulin, "Steganographic methods for covert communication over timing channels," presented at the 2008 Information Theory and Applications Workshop, San Diego, CA, Jan 2008 and at the DARPA Workshop, Cambridge, MA, July 2007.

[2] B. Lampson, "A note on the confinement problem," *Communications of the ACM*, vol. 16, no. 10, pp. 613–615, 1973.

[3] M. Kang, I. Moskowitz, and D. Lee, "A network pump," *IEEE Transactions on Software Engineering*, vol. 22, no. 5, pp. 329–338, 1996.

[4] I. Moskowitz and A. Miller, "The channel capacity of a certain noisy timing channel," *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1339–1344, 1992.

[5] V. Berk, A. Giani, G. Cybenko, and N. Hanover, "Detection of covert channel encoding in network packet delays," Department of Computer Science, Dartmouth College, Tech. Rep. TR2005536, 2005.

[6] S. Cabuk, C. Brodley, and C. Shields, "IP covert timing channels: design and detection," in *Proceedings of the 11th ACM conference on Computer and Communications Security*. ACM New York, NY, USA, 2004, pp. 178–187.

[7] G. Shah, A. Molina, and M. Blaze, "Keyboards and covert channels," in *Proceedings of the 15th conference on USENIX Security Symposium*, 2006, pp. 5–5.

[8] X. Luo, E. Chan, and R. Chang, "TCP covert timing channels: Design and detection," in *IEEE International Conference on Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008*, 2008, pp. 420–429.

[9] J. Giles and B. Hajek, "An information-theoretic and game-theoretic study of timing channels," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2455–2477, 2002.

[10] P. Moulin and J. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 563–593, 2003.

[11] Y. Wang and P. Moulin, "Perfectly secure steganography: Capacity, error exponents, and code constructions," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2706–2722, 2008.

[12] S. Servetto and M. Vetterli, "Communication using phantoms: Covert channels in the internet," in *2001 IEEE International Symposium on Information Theory, 2001. Proceedings*, 2001, p. 229.

[13] V. Anantharam and S. Verdú, "Bits through queues," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 4–18, 1996.

[14] I. Ezzeddine and P. Moulin, "Achievable rates for queue-based timing stegocodes," in *2009 Info. Theory Workshop*, Sicily, Italy, Oct. 2009, pp. 379–383.

[15] C. Shannon, "Channels with side information at the transmitter," *IBM Journal of Research and Development*, vol. 2, no. 4, pp. 289–293, 1958.

[16] V. Paxson and S. Floyd, "Wide area traffic: the failure of poisson modeling," *IEEE/ACM Transactions on Networking (TON)*, vol. 3, no. 3, pp. 226–244, 1995.

[17] D. Song, D. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on ssh," in *2001 USENIX Security Symposium, 2001. Proceedings*, vol. 2, 2001, p. 3.

[18] R. Miller, "Response time in man-computer conversational transactions," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I.* ACM New York, NY, USA, 1968, pp. 267–277.

[19] D. Lindley, "The theory of queues with a single server," in *Proceedings of the Cambridge Philosophical Society, vol. 48, issue 02*, 1952, p. 277.

[20] P. Burke, "The output of a queuing system," *Operations Research*, vol. 4, no. 6, pp. 699–704, 1956.

[21] N. Bambos and J. Walrand, "An invariant distribution for the G/G/1 queueing operator," *Advances in Applied Probability*, vol. 22, no. 1, pp. 254–256, 1990.

[22] C. Shannon, "Some geometrical results in channel capacity," *Claude Elwood Shannon Collected Papers*, vol. 1, no. 1, pp. 259–264, 1992.

[23] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 460–473, 1972.

[24] B. Prabhakar and R. Gallager, "Entropy and the timing capacity of discrete queues," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2706–2722, 2003.

[25] A. Bedekar and M. Azizoglu, "The information-theoretic capacity of discrete-time queues," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2706–2722, 1998.