ABSTRACT

Title of dissertation: CROSS-LAYER DISTORTION CONTROL FOR DELAY SENSITIVE SOURCES

Azadeh Faridi
Doctor of Philosophy, 2007

Dissertation directed by: Professor Anthony Ephremides
Department of Electrical and Computer Engineering

The existence of layers in the traditional network architecture facilitates the network design by modularizing it and thus enabling isolated design of the different layers. However, due to the inherent coupling and interactions between these layers, their isolated design often leads to suboptimal performance. On the other hand, the recent popularity of realtime multimedia applications has pushed the boundaries of layered designs. Cross-layer network design provides opportunities for significant performance improvement by selectively exploiting the interactions between layers, and therefore has attracted a lot of attention in recent years.

Realtime multimedia applications are characterized by their delay-sensitivity and distortion-tolerance. The focus of this thesis is on Source Coding for Delay-Sensitive Distortion-Tolerant data. In particular, we notice that even though using longer descriptions for source symbols results in smaller distortion for each particular symbol, it also increases the delay experienced in the network, which in turn causes information loss for a delay-sensitive source, and therefore, increases the overall distortion of the received message. In this thesis we investigate this trade-off across the layers by considering two different problems.

In the first problem, we focus on a single source-destination pair to exploit the interconnection between Source Coding, traditionally a presentation layer component, and Parallel Routing, a network layer issue. We use a Distortion Measure that combines signal reconstruction fidelity with network delay. We minimize this measure by jointly choosing the Encoder Parameters and the Routing Parameters. We look at both single-description and multiple-description codings and perform numerical optimizations that provide insight into design tradeoffs which can be exploited in more complex settings.

We then investigate the problem of finding minimum-distortion policies for streaming delay-sensitive distortion-tolerant data. We use a cross-layer design which exploits the coupling between the presentation layer and the transport and link layers. We find an optimum transmission policy for error-free channels, which is independent of the particular form of the distortion function when it is convex and decreasing. For a packet-erasure channel, we find computationally efficient heuristic policies which have near optimal performance.

CROSS-LAYER DISTORTION CONTROL FOR DELAY SENSITIVE SOURCES

by

Azadeh Faridi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Committee:
Professor Anthony Ephremides, Chair/Advisor
Professor Alexander Barg
Professor Michael Fu
Professor Prakash Narayan
Associate Professor Sennur Ulukus

© Copyright by
Azadeh Faridi
2007

DEDICATION

To my mother,
who artfully planted the seed of love for mathematics in me.

To the memory of my father,
who taught me about responsibility when I could not yet pronounce the word.

To my sister, Darya,
who taught me how to always keep an open mind.

ACKNOWLEDGEMENTS

First and foremost I would like to express my sincere gratitude to my advisor, Professor Anthony Ephremides, for his generous support and guidance throughout my PhD program.

I would also like to thank my dissertation committee members, Professors Alexander Barg, Sennur Ulukus, Prakash Narayan, and Michael Fu, for their time and feedback.

I am much indebted to Professor André L. Tits, for his continuous encouragement and feedback.

I would like to thank Francesc and my family, my mother and my sisters, for believing in me throughout my journey and for their continuous emotional support and encouragement.

Finally, I am also thankful to all my friends, especially Ali Pezashki, Anna Pantelidou, Tissaphern Mirfakhrai, Alireza Nojeh, Negin Nejati, and Rosta Farzan for their constant direct and indirect support.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

VoIP	Voice over IP
SDC	Single Description Coding
MDC	Multiple Description Coding
DDC	Double Description Coding
FCFS	First Come First Serve
MSE	Mean Square Error
i.i.d.	Independent Identically Distributed

OL	Open Loop
CEC	Certainty Equivalent Control

Chapter 1

Introduction

The recent increase of the popularity of rich multimedia realtime and peer-to-peer applications, as well as the ever more popular use of the wireless medium, have given rise to the need for rethinking the traditional layered designs of Communication Networks.

The traditional layered network architecture has facilitated the design of the networks by enabling isolated design of different layers. However, it is recognized that the inherent coupling and dependence among the layers of traditional architectures provides opportunities for significant performance improvement by exploiting these interactions selectively.

Cross-layer network design has attracted a lot of attention in recent years for different kinds of communication networks [1,2], and especially for wireless networks due to the unique challenges associated with the use of the wireless medium [3]. Significant improvements can be achieved by sharing information about the varying wireless channel conditions across the layers [4–6]. Energy concerns, specially in wireless ad hoc networks [7,8], have given rise to interesting cross-layer optimization problems [5,9–11].

The growing demand for real-time multimedia applications such as Voice over IP, multimedia teleconferencing, and gaming have introduced yet another challenge for the network designers which can be very effectively addressed through cross-layer optimization [6, 12].

This thesis is concerned with cross-layer optimization of Source-Coding for Delay-Sensitive applications. In particular, we focus on the distortion-delay tradeoff which arises when dealing with delay-sensitive distortion-tolerant data, such as real-time multimedia applications.

For such applications, any delay incurred by packets transiting the network could decrease the perceived quality at the receiver. The delay budget that each packet can afford is determined by the application. For example for Voice over IP (VoIP), a network delay of more than fifty milliseconds creates unacceptable quality or equivalently high distortion. In this thesis, we focus on applications with *hard delay-constraints*, for which, source symbol descriptions arriving after their corresponding deadlines are discarded and therefore result in maximum quality degradation when the signal is reconstructed.

A distortion function is a mathematical performance measure that indicates the amount of degradation in the quality of the decoded information. While encoding the source symbols into longer descriptions results in smaller distortion when the source symbols are reconstructed, it also causes longer delays due to capacity limitations of the channels utilized in the network. Due to the hard delay-constraint on each source symbol, long delays can result in loss of information, which in turn increases the overall distortion of the reconstructed message. The focus of this thesis is precisely on this tradeoff and the benefits that can be gained by exploiting the interconnection between the layers of traditional architecture when dealing with this tradeoff.

In particular, we note that when dealing with applications with hard delay constraints, the traditionally popular average delay performance metrics can often be misleading. Instead, performance metrics should reflect the dynamics of the delay experienced by each individual data unit traveling through the network. Therefore, we define a distortion measure that takes into account both the distortion incurred due to lossy source encoding, as well as the degradation caused when descriptions are received after their deadline.

Two different problems are considered in this thesis. In Chapter 2, we consider the problem of finding the best coding and routing strategies when there are two parallel routes available between a delay-sensitive source and its destination. This leads to a joint optimization problem which exploits the inherent coupling between the presentation layer and the network layer. Then in Chapter 3, we focus on the problem of finding the best transmission policy for sending a given number of delay sensitive source symbols to a destination. This is sometimes referred to as *streaming* and finds its use in applications such as video-on-demand, where a user views a content as it is being downloaded. In such applications, users often prefer a certain amount of degradation in the image quality to a large number of pauses in the stream of video they are viewing. Our goal here is to find a balance between the signal's quality and delay, by simultaneously optimizing the source coding parameters as well as the scheduling of the transmissions, while taking both distortion and delay into consideration. A more thorough introductory section to each of these problems is presented at the beginning of their corresponding chapter.

Chapter 2

Source Coding and Parallel Routing

2.1 Introduction

In this chapter we demonstrate the advantages of using cross-layer approaches by focusing on the interaction between Source Coding (traditionally a layer 6 issue with physical layer connections) and Routing (a layer 3 issue). In particular we observe that in networks, often, packets are duplicated and routed over separate paths to their common destination to increase the chance of timely delivery and to provide protection against packet loss and long delays. This practice is called parallel routing and obviously results in increasing the offered load to the network.

At the same time we know that compression techniques can reduce packet-length. So a natural question is how to choose the source encoding parameters in conjunction with the routing parameters so as to minimize a suitable distortion measure that incorporates both the quality of the signal reconstruction as well as its delay.

In particular if multiple description coding is used the possibility arises naturally that each description follows a different path to the destination, thereby combining the idea of protection through redundancy with the need to reduce the traffic load.

Finally choosing the packet-length itself (or, more accurately, the source encoding rate or “symbol”-length) along with designing the multiple description coding and choosing the routing parameters goes even further in exploiting the observed interrelationships.

In this chapter we study precisely this problem in the simplest of settings and identify and analyze the underlying trade-offs. More complicated and realistic models can be naturally studied along similar lines in the future.

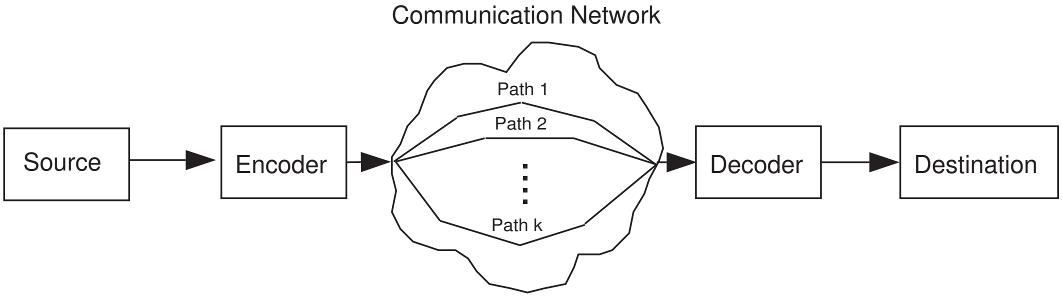


Figure 2.1: System Diagram

Consider the simple diagram shown in Figure 2.1 that consists of a source-destination pair, a source encoding-decoding module and a communication network that delivers packets from the source to the destination. The source is delay-sensitive, i.e., the source symbols that arrive after their corresponding deadlines will be useless at the receiver.

Normally, each source symbol is compressed and appropriately transformed to a single codeword by the encoder before entering the network. This coding scheme is referred to as Single Description Coding (SDC). The coded symbols (i.e., packets) usually follow a single path determined by the network to reach to the destination. A path usually consists of several segments or communication links that connect nodes in a network. Since each node could experience congestion, there is a chance that a particular packet will be excessively delayed or even be dropped from the transmission queue.

As mentioned before, parallel transmission reduces the chance of packet loss when congestion occurs in a network; however, the excess bit rate introduced by the extra copies creates additional traffic that in effect contributes to congestion and thus increases the probability of packet loss.

A Multiple Description Coder (MDC), [13] and [14], also transforms the sequence of source symbols into several parallel data streams; however, no excess bit rate over a single description coder is used. Therefore, the source traffic can potentially adjust itself to the state of the network without being a contributing factor to congestion [15].

Achievable rate-distortion region for a Double Description Coder (DDC) and a Gaussian source has been studied in [16], [17]. In [15], it is shown that for a simple network that consists of two parallel communication links, using an optimized DDC amounts to significant reduction in distortion compared to SDC. The distortion is minimized in [15] by optimization of some of the coding parameters. In [18] and [19] further gain is achieved, in SDC and DDC systems respectively, by considering the network parameters in the optimization process. However, in all these works, the expected packet-length is fixed. As a result, for high arrival rates the optimum solution is to use one of the links as a dump for the excess traffic and operate it fully congested in order to save the other link from being congested, and therefore, a fraction of the capacity of the network is wasted. In the work presented in this chapter, we add the average packet-length to our optimization parameters, which results in significant performance improvement. We also extend our analysis to the case where queue length is finite.

This chapter is organized as follows. In Section 2.2 general system model and assumptions are explained. The modeling and formulations specific to the SDC and DDC systems are explained in Section 2.3 in detail. The results are presented in Section 2.4. Some of these results were first reported in [20][1]and [21].

2.2 General Modeling and Assumptions

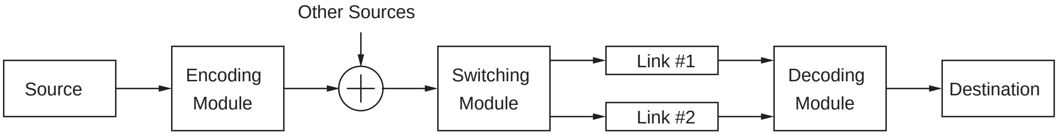


Figure 2.2: General System Model

To better focus on the interrelationships of the source coding and routing parameters, we consider the simple setting shown in Figure 2.2 along with the following simplifying assumptions. Note that these assumptions are similar to those taken in [18] and [19], and so the results are comparable.

1. The source generates i.i.d., zero mean, unit variance, Gaussian symbols.

2. The source is loss-tolerant and delay-sensitive (the source symbols can tolerate a delay of up to Δ seconds from generation to reception.)

3. Two classes of coding schemes, i.e., SDC (Single Description Coding) and DDC (Double Description Coding) are considered. In the case of SDC each

[1] The parts presented in [20] in which the optimization is done with fixed expected packet length were mostly carried out by the coauthors of the author of this thesis, and therefore, are not presented here except in some of the figures for comparison.

source symbol is encoded into a single description (packet) with an expected length of R bits. In the DDC case however, each symbol is encoded into two packets of average lengths R_1 and R_2 bits.

4. Two cases are studied for the packet-length distribution:

    (a) Exponential packet-length

    (b) Deterministic packet-length

5. The output of the encoding module is combined with the traffic coming from other similar (i.e., with the same packet-length distribution) independent sources. These traffic streams are coming from different paths in the network. According to the Palm-Khintchin theorem [22] this type of aggregate traffic converges to a Poisson point process. We therefore assume that the arrival process of the aggregate traffic is a Poisson process.

6. The model includes two disjoint, noise-free communication links with capacities C_i, $i = 1, 2$ bits/second.

7. The switching module routes each packet of the aggregate traffic to one of the two links.

8. Each communication link is modeled by a First Come First Serve (FCFS) single server queue. The time that every packet spends in service at the queues is assumed to be proportional to the length of that packet.

9. The decoding module drops the packets that have experienced a delay exceeding Δ seconds.

Details regarding the operation of the system with SDC and DDC are considered in the next section.

2.3 Problem Formulation

In this section we summarize the SDC and DDC systems and derive the formulations required for analyzing these systems.

2.3.1 System with Single Description Coding (SDC)

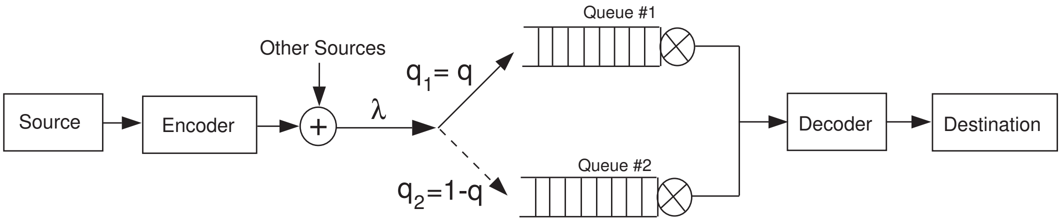


Figure 2.3: SDC System Model

In the SDC system each source symbol is encoded into a single packet with an average rate of R bits/symbol (in the deterministic case all symbols are encoded into packets of R bits). The packets generated by the encoder are combined with the traffic coming from other similar and independent sources in the network. The aggregate traffic forms a Poisson process with rate λ packets/sec. This traffic is routed to the first queue with probability $q_1 = q$ and to the second queue with probability $q_2 = 1 - q$. In other words λ_i, the arrival rate to the i^{th} queue, is

$$\lambda_i = q_i \lambda$$

We refer to q as the *switch parameter* hereafter. Figure 2.3 represents the SDC system model.

2.3.1.1 SDC Distortion

Let T be the random variable indicating the total delay that a packet experiences from generation until reaching its destination (hereafter referred to as end-to-end delay). If D represents the achievable mean square error (MSE) distortion for an i.i.d., zero mean, unit variance, Gaussian source, based on [23] and on our concept of delay-based distortion, we have [15]:

$$D = \begin{cases} 2^{-2R} \ , & T \leq \Delta \\ 1 \ , & T > \Delta \end{cases}$$

The end-to-end average distortion, therefore, can be written as

$$\overline{D} = 2^{-2R} \Pr[T \leq \Delta] + \Pr[T > \Delta] \tag{2.1}$$

The goal here is to minimize the average end-to-end distortion by choosing optimal values for R and q. To calculate the overall distortion, we need to know the delay distribution. The delay a packet experiences in the queue depends on the service time of the queue, which in turn depends on the packet-length. Denote the average service rate of the i^{th} queue by μ_i. Then we can write

$$\mu_i = \frac{C_i}{R} \text{ (packets/sec)}, \quad i = 1, 2$$

Three different cases are studied here

1. Infinite-buffer, exponential packet-length modelled by an M/M/1 queue
2. Infinite-buffer, deterministic packet-length modelled by an M/D/1 queue
3. Finite buffer, deterministic packet-length modelled by an M/D/1/k queue

It should be noted that since we are dealing with the rate distortion function, the most practical choice for the packet-length distributions would be one that complies with the codewords of the encoders that can achieve the rate-distortion function. To approach the rate-distortion limits, one needs to employ a vector quantizer with a sufficiently large block length (encode n symbols at a time). If n is large enough, the block lengths will approach a constant in the limit according to the Asymptotic Equipartition Property (AEP). This suggests that the more realistic choice for the packet-length distribution is the deterministic constant packet-length. However, for some of the problems we consider, the exponential packet-length distribution greatly simplifies the numerical analysis of the problems. Because of this, and in order for our results to be comparable to those in [18] and [19], where packet-lengths have exponential distribution, in most of the problems we consider, we assume the packet-lengths to have an exponential distribution.

Note that in the finite-buffer case, the deterministic packet-length assumption results in simpler calculations compared to the exponential case. Since the deterministic case is more realistic to deal with, and also, there are no previous exponential packet-length finite-buffer results to compare these results with, only the deterministic assumption was considered in that case.

It should finally be noted that large block lengths will result in long delays. Since delay is a crucial component of the distortion function we will choose, we need to implicitly assume that the transmission rates are sufficiently fast so that “long” packets in number of bits are not “long” with respect to time.

2.3.1.2 M/M/1 Delay Distribution

The distribution of the system delay for an $M/M/1$ queue is known [24] to be given by

$$F_{T_i}(t) = \Pr\left[T_i \leq t\right] = \left(1 - e^{-\mu_i(1-\rho_i)t}\right) u(t), \qquad i = 1, 2$$

where T_i, $i = 1, 2$ is the random variable indicating the total delay of a typical packet that is routed to queue i, $u(t)$ is the unit step function, and ρ_i is the loading factor of queue i given by

$$\rho_i = \min\left(\frac{q_i\lambda}{\mu_i}, 1\right), \qquad i = 1, 2$$

Therefore, the total delay of a typical packet is given by

$$\Pr[T > \Delta] = \sum_{i=1}^{2} q_i(1 - F_{T_i}(\Delta)) = \sum_{i=1}^{2} q_i e^{-\mu_i(1-\rho_i)\Delta}\ u(\Delta) \tag{2.2}$$

2.3.1.3 M/D/1/K Delay Distribution

In this case the length of the packets, R, is taken to be a deterministic constant. Let the parameter B represent the number of bits that fit in the system ,i.e., queue and service together (hereafter referred to as "buffer size"), and let C indicate the capacity of the link. Define the maximum buffer size, $\hat{B}$ as

$$\hat{B} = \Delta \times C$$

If after the arrival of a given packet, there are no more than $\hat{B}$ bits in the system, that packet will reach its destination before the deadline. Therefore, there is no point in having a buffer size that is greater than $\hat{B}$ bits[2], and so we choose $B \leq \hat{B}$ in our analysis.

The probability of blocking, i.e., the probability that a packet arrives when the buffer is full, for an M/G/1/K queue is known in the queueing literature (see [25] for example). Replacing the general service distribution in the M/G/1/K model of [25] with a deterministic constant service time, results in the following derivations for

[2] Note that when $\hat{B}$ is not an integer multiple of R, when there are $\lfloor \frac{\hat{B}}{R} \rfloor$ packets in the system, a new arrival could make it in time if the packet that is already in service has spent more than $\frac{1}{\mu} - (\Delta - \lfloor \frac{\hat{B}}{R} \rfloor \frac{1}{\mu})$ in service. But in order to be able to admit such packets to the queue, we need to increase the buffer size by R bits, which is not known a priori and can be anywhere between 1 bit and $\hat{B}$ bits. We will discuss this issue further in the infinite-buffer case.

the M/D/1/K queue of our problem.

Let the parameter K denote the number of packets that fit in the buffer, in other words

$$K = \left\lfloor \frac{B}{R} \right\rfloor$$

The service rate μ is a deterministic constant in this case, and $\mu = C/R$ packets/second. Let L_n indicate the number of packets left in the buffer right after the n^{th} departure. Define π_k as the steady state probability that after a departure, k packets remain in the system, i.e.,

$$\pi_k = \lim_{n \to \infty} \Pr[L_n = k], \qquad 0 \leq k \leq K-1$$

and define

$$\pi'_k = \pi_k / \pi_0$$

Then π'_k can be calculated recursively as follows

$$\begin{aligned} \pi'_0 &= 1 \\ \pi'_{k+1} &= \frac{1}{a_0}(\pi'_k - \sum_{j=1}^{k} \pi'_j a_{k-j+1} - a_k), \quad 0 \leq k \leq K-2 \end{aligned}$$

where

$$a_k = \frac{(\lambda/\mu)^k}{k!} e^{-\lambda/\mu}, \qquad 0 \leq k \leq K-2$$

Therefore

$$\pi_0 = (\sum_{k=0}^{K-1} \pi'_k)^{-1}$$

$$\pi_k = \pi_0 \pi'_k, \qquad 0 \leq k \leq K-1$$

Now define

$$P_k = \Pr[\ k \text{ messages in the system at an arbitrary time }]$$

$$\bar{\pi}_k = \Pr[\text{ an arrival finds } k \text{ messages in the system }]$$

Using the PASTA property (Poisson Arrivals See Time Average), $P_k = \bar{\pi}_k$. Since for every departure, there is an arrival and for every arrival there is a departure unless that arrival is blocked, we have

$$\bar{\pi}_k = \underbrace{(1 - P_{\text{block}})}_{c} \pi_k \quad \Rightarrow \quad P_k = c\pi_k, \qquad 0 \leq k \leq K-1$$

where $P_{\text{block}} = P_K$ is the blocking probability. Given that $\sum_{k=0}^{K} P_k = 1$ and that $P_0 = 1 - c\rho$, we get

$$c = \frac{1}{\pi_0 + \rho}$$

where $\rho = \lambda/\mu$. So we finally have

$$\begin{cases} P_k = \dfrac{\pi_k}{\pi_0 + \rho} \ , \qquad 0 \leq k \leq K-1 \\ P_K = 1 - \dfrac{1}{\pi_0 + \rho} \end{cases}$$

Since the only packets that will not make it to the destination on time are those that have been blocked, we have

$$\Pr[T > \Delta] = P_{\text{block}} = P_K$$

Note that the above derivations do not hold for the case where $K = 0$, in which case the solution is trivial and is $P_K = 1 = P_0 = P_{\text{block}}$.

2.3.1.4 M/D/1 Delay Distribution

The delay distribution for an M/D/1 queue is well known [26]. However, since we are going to use numerical methods to evaluate the delay probability, we choose to use a modification of the M/D/1/K formulas, because they have a better numerical behavior and moreover this enables us to easily compare the M/D/1/K and M/D/1 case.

It can be seen in the M/D/1/K derivations that π'_k , $k = 0, 1, \cdots, K-1$ are independent of the value of K. On the other hand $\pi_k = \pi_0 \pi'_k$, so the only component that is directly dependent on the value of K is π_0. Note that

$$P_0 = \frac{\pi_0}{\pi_0 + \rho}$$

Therefore

$$\pi_0 = \frac{P_0 \rho}{1 - P_0}$$

On the other hand we know that for an infinite K, we will have an M/D/1 queue in which case $P_0 = 1 - \rho$. Replacing P_0 in the above equation we have

$$\pi_0 = 1 - \rho$$

Therefore the M/D/1/K derivations can be used with the initial condition $\pi_0 = 1 - \rho$ to calculate the steady state probabilities of an M/D/1 queue. The delay probability can be calculated as follows

$$\Pr[T > \Delta] = 1 - (\sum_{k=0}^{\hat{K}-1} P_k + P_{\hat{K}} e^{-((\hat{K}+1)/\mu - \Delta)\lambda}) \tag{2.3}$$

where $\hat{K} = \left\lfloor \frac{\hat{B}}{R} \right\rfloor$. The last term takes care of the cases where Δ is not an integer multiple of $1/\mu$ (Figure 2.4). In this case, if an arrival occurs at the moment when $\hat{K}$ packets are in the queue but the packet currently being serviced has spent more than $\frac{\hat{K}+1}{\mu} - \Delta$ seconds in service, the new arrival will be able to make it to the destination before its deadline.

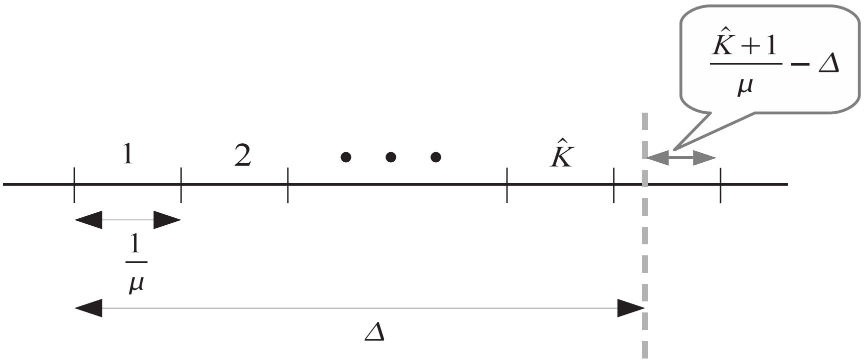


Figure 2.4: M/D/1 Delay: When Δ is not an integer multiple of the service time, it is possible that the $(\hat{K} + 1)$st packet accepted to the queue meets the deadline

2.3.2 Systems with Double Description Coding (DDC)

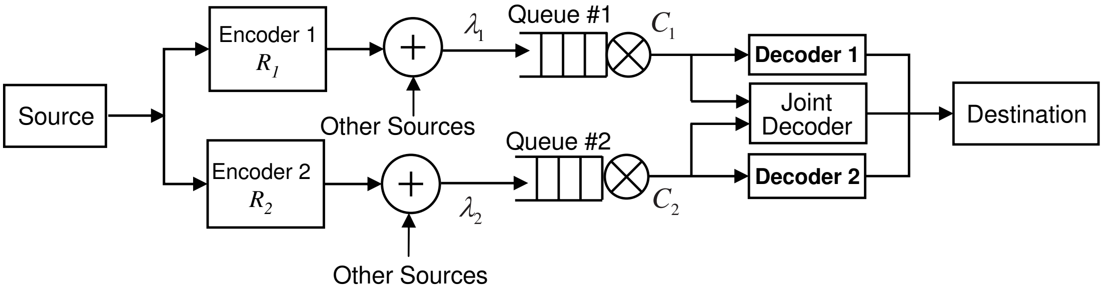


Figure 2.5: DDC System Model

In the DDC system, the source information is encoded by two side-encoders as shown in Figure 2.5. The codeword length generated by encoder i has an average length of R_i, $i = 1, 2$ bits per packet, with $R = R_1 + R_2$. The traffic from other similar (in packet-length distribution), independent sources is combined with the output traffic of each encoder. The arrival process of the aggregate traffic corresponding to encoder i is assumed to be a Poisson process with an average rate of $\lambda_i = \lambda$, $i = 1, 2$ packets/sec. In this system, the output of encoder i is routed to queue i. Note that the two side encoders generate the two descriptions of a symbol *simultaneously*. However, since two independent large traffic streams from "other sources" are combined with the outputs of the two encoders, we can assume that the arrival processes to the two queues are independent from one another. This means that the traffic streams entering the two queues are two independent Poisson processes.

2.3.2.1 DDC Distortion

Let T^i $(i = 1, 2)$ be the total delay experienced by the output packets of encoder i. Since the outputs of encoder i is routed to queue i, we have $T^i = T_i$, where T_i is the delay experienced in queue i. We use T_i hereafter in order to be consistent with the notation used in the SDC case. If D_{DDC} represents the achieved MSE distortion, then for an i.i.d., zero mean, unit variance, Gaussian source, based on [16] and on our concept of delay-based distortion, we have

$$D_{\text{DDC}} = \begin{cases} d_0 = 2^{-2(R_1+R_2)} \frac{1}{1-[max(0,(\sqrt{\Pi}-\sqrt{\Lambda}))]^2} & \text{if} \quad T_1 \leq \Delta \;\; \& \;\; T_2 \leq \Delta \\ d_1 = 2^{-2R_1(1-\delta_1)} & \text{if} \quad T_1 \leq \Delta \;\; \& \;\; T_2 > \Delta \\ d_2 = 2^{-2R_2(1-\delta_2)} & \text{if} \quad T_1 > \Delta \;\; \& \;\; T_2 \leq \Delta \\ 1 & \text{if} \quad T_1 > \Delta \;\; \& \;\; T_2 > \Delta \end{cases} \tag{2.4}$$

where

$$0 \leq \delta_1, \delta_2 < 1$$

$$\Pi = (1 - d_1)(1 - d_2) \;\; \& \;\; \Lambda = d_1 d_2 - 2^{-2(R_1+R_2)}$$

The end-to-end average distortion therefore can be written as

$$\overline{D}_{DDC} = d_0 \Pr[\, T_1 \leq \Delta, \; T_2 \leq \Delta \,] + d_1 \Pr[\, T_1 \leq \Delta, \; T_2 > \Delta \,] + \tag{2.5}$$

$$d_2 \Pr[\, T_1 > \Delta, \; T_2 \leq \Delta \,] + \Pr[\, T_1 > \Delta, \; T_2 > \Delta \,]$$

Our goal here is to minimize the average distortion over all parameters it depends on, i.e., R_1, R_2, δ_1, and δ_2.

Note that in (2.4), δ_i signifies the amount of redundancy in side-encoder i. Low values of δ_i (i.e., $\delta_i \approx 0$) indicate good individual descriptions that jointly contribute little extra information beyond one alone. On the other hand, high values of δ_i (i.e., $\delta_i \approx 1$) indicate independent descriptions that are not individually good; however, jointly they can achieve the same amount of distortion as in the case of an SDC encoder of rate $R = R_1 + R_2$. This is because according to (2.4), in order to get a perfect joint reconstruction (i.e. $d_0 = 2^{-2(R_1+R_2)}$), we need to have

$$\Pi = \Lambda$$

or equivalently

$$(1 - d_1)(1 - d_2) = d_1 d_2 - 2^{-2(R_1+R_2)}$$

so

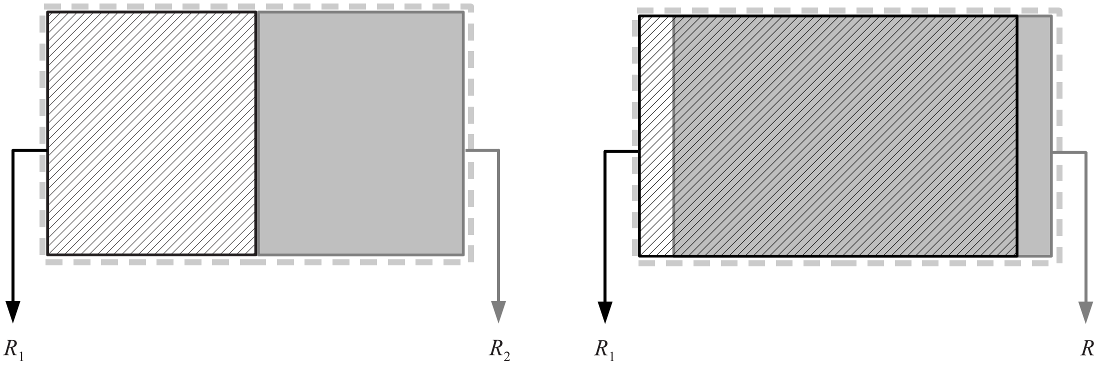
$$d_1 + d_2 = 1 + 2^{-2(R_1+R_2)} = \underbrace{(1 - 2^{-2R_1})(1 - 2^{-2R_2})}_{>\ 0} + 2^{-2R_1} + 2^{-2R_2}$$

therefore

$$d_1 + d_2 > 2^{-2R_1} + 2^{-2R_2}$$

This means that in order to minimize the value of d_0 we need to let the side encoders operate inside the rate-distortion region and therefore we need to have $\delta_i > 0$.

To facilitate the understanding of the parameter δ_i, in Figure 2.6 we illustrate an example in which an image, shown by the large dashed outer rectangle, is encoded into two descriptions of lengths R_1 (hatched rectangle) and R_2 (gray filled rectangle) for two different cases of large and small δ_i. We see in this example that when the joint description is good (Figure 2.6(a)), the individual descriptions are not very good since they contain information about half of the image. If the individual descriptions are good (Figure 2.6(b)), they carry "similar" information about the entire image, and therefore combining the two descriptions will not significantly improve the distortion.



(a) Large δ_i: $d_i \gg 2^{-2R_i}$, $d_0 \ll \min(d_1, d_2)$ (b) Small δ_i: $d_i \approx 2^{-2R_i}$, $d_0 \approx \min(d_1, d_2)$

Figure 2.6: Understanding δ_i, the redundancy of encoder i.

2.3.2.2 DDC Delay

Since the two queues are independent, we have

$$\Pr[\ T_1 > \Delta,\ T_2 > \Delta\ ] = \Pr[\ T_1 > \Delta\ ]\ \Pr[\ T_2 > \Delta\ ]$$

The delay probabilities therefore can be calculated using the derivations in the SDC case, by replacing T_i in SDC with T_i of DDC. Note that here μ_i and ρ_i are given by

$$\mu_i = \frac{C_i}{R_i}$$

$$\rho_i = \frac{\lambda_i}{\mu_i}$$

2.4 Results

As it was mentioned earlier, the goal here is to minimize the average end-to-end distortion over the parameters it is dependent upon. In [15], [18], and [19] the expected packet-length (encoding rate) R was fixed. In this work, we consider the encoding rate R as one of the optimization parameters. From the coding point of view, R determines the average codeword length; therefore, higher R results in smaller value of distortion at the decoder. On the other hand, R is the average packet-length which means that higher R results in longer packets or equivalently longer delays; and therefore, higher distortion for a delay sensitive source. This tradeoff points to the possibility of the existence of an optimal R that minimizes the average end-to-end distortion for such delay sensitive sources.

In this section the results for the SDC and DDC systems for the three different cases of M/M/1, M/D/1, and M/D/1/K queues are presented and compared to the previous works. At the end of this section, the SDC and DDC systems are compared to each other and it is shown how using multiple description coding can lead to a better use of the path diversity. Throughout the section we take $\Delta = 50$ msec.

It should be noted that, in all the cases, the optimization of the distortion was performed numerically by exhaustive search over the range of all parameter values.

2.4.1 SDC Results

The problem of SDC with parallel routing has been previously studied in [18] for a system with two parallel infinite buffer-length queues and an SDC encoder with an exponential packet-length distribution with a fixed average length of $R = 6$ bits/packet. The optimization is performed over the switch parameter, q, i.e.,

$$q^* = \underset{q}{\text{Argmin}} \ \overline{D}(q, R = 6)$$

Here we consider the exact same model, but we add the average rate R to our optimization parameters. In other words, we have

$$(q^*, R^*) = \underset{(q,R)}{\text{Argmin}} \ \overline{D}(q, R)$$

We consider both cases of deterministic and exponential packet-length distributions. In the case of exponential packet-length distribution, we compare our results to those obtained in [18].

2.4.1.1 Deterministic Packet Length

Here we consider the SDC system when the encoder generates packets of a deterministic constant length. We compare the case where the queue length (or buffer size) is infinite to several different cases where queue lengths are finite. Figures 2.7 and 2.8 show the minimum distortion and the optimum packet-length respectively in this case for different buffer sizes. The optimum switch parameter is $q^* = 0.5$ for all values of the arrival rate λ.

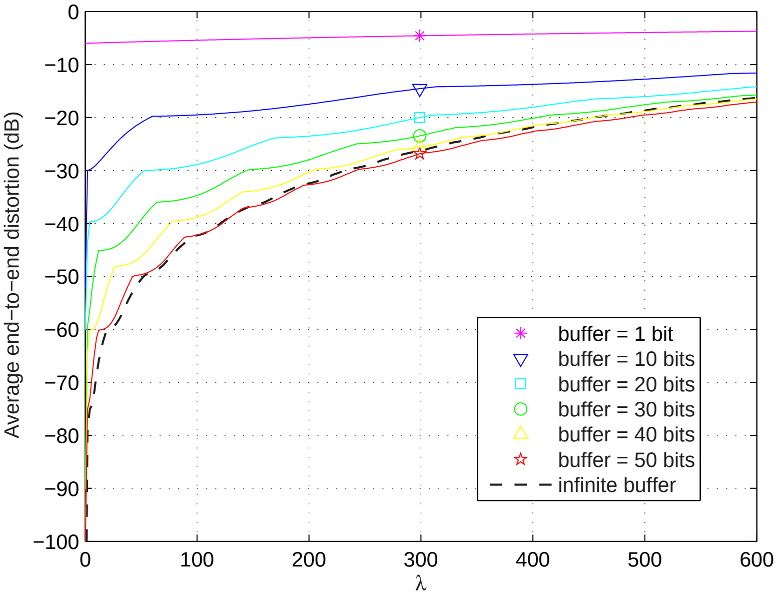


Figure 2.7: Average End-to-End Distortion for SDC with deterministic packet-length, $C_1 = C_2 = C = 1000$ bits/sec

As it can be seen in Figure 2.7, in the finite-buffer cases, as the buffer size increases, the distortion decreases, however the amount by which the distortion decreases becomes less significant as the buffer-length gets closer to $\hat{B} = \Delta \times C = 50$ bits.

Comparing the curve for $B = 50$ and $B = \infty$, it can be seen that for fast arrivals the finite buffer outperforms the infinite buffer. This is due to the fact that as we get closer to $\hat{B}$ the limited length of the buffer comes to our benefit by eliminating packets that will not make it on time to the destination. On the other hand, for slow arrivals, the infinite buffer results in smaller distortion than $B = \hat{B} = 50$. This can be explained by looking at the last term of equation (2.3). As was explained earlier, in the case where Δ is not an integer multiple of the service

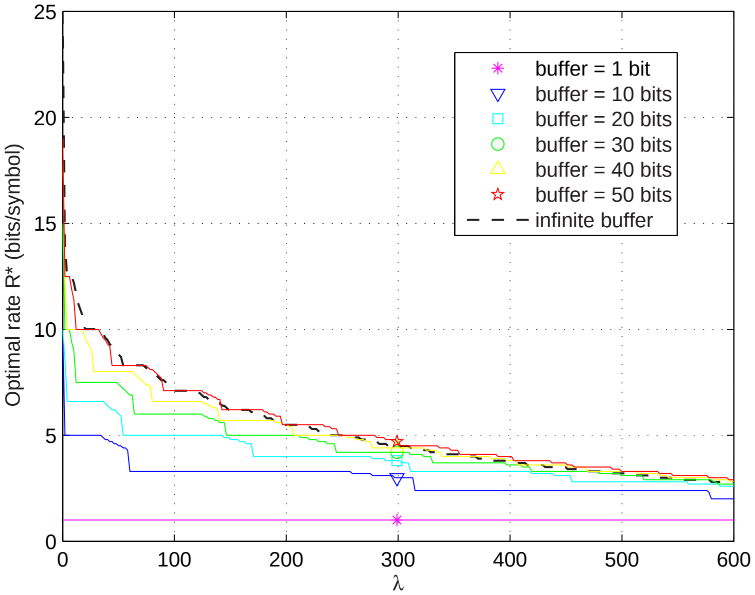


Figure 2.8: Optimum encoding rate for SDC with deterministic packet length, $C_1 = C_2 = C = 1000$ bits/sec

time, a packet can arrive at a moment when there are $\hat{K}$ packets in the system and still make it to the destination in time. That happens if the packet that is in service at the moment of the new arrival, has already spent a sufficient amount of time in service. In the finite-buffer case, such arrivals will not be admitted. On the other hand, as we can see in Figure 2.8, the smaller the arrival rate, the larger the optimal packet-length. So for small arrival rates, loosing a packet results in a greater loss of information than at high arrival rates, therefore the queue with $B = 50$ will have worse performance than the infinite-buffer queue. When arrivals are fast, however, packets are chosen to be smaller and the loss of a packet has an insignificant impact on the overall distortion and therefore the finite-buffer queue with $B = 50$ outperforms the infinite-buffer queue.

Figure 2.8 shows the variation of the optimum encoding rate versus the arrival

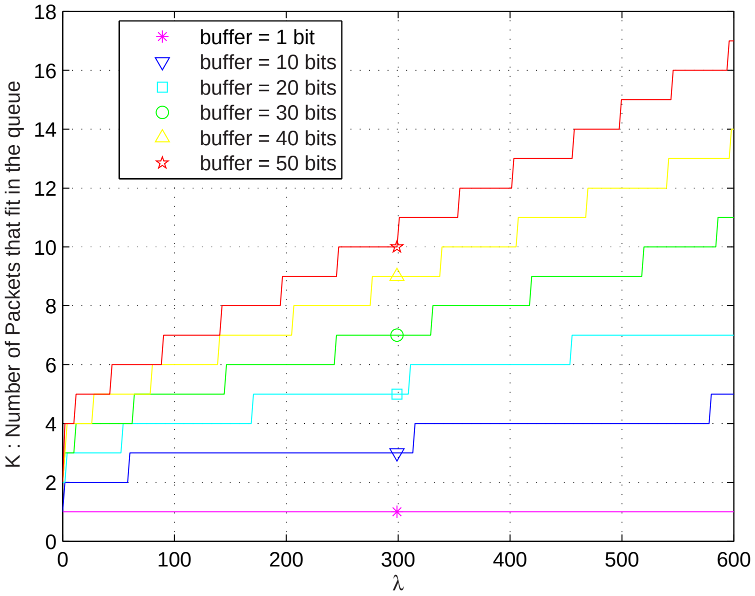


Figure 2.9: Number of packets that fit in the queue for optimized SDC with deterministic packet-length, $C_1 = C_2 = C = 1000$ bits/sec

rate (or packet-length) for different buffer sizes. As we can see here, the optimum packet-length decreases as the arrival rate increases. We can also see that the curves for the optimum rate are not smooth. This is due to the fact that the distribution of the delay a packet experiences depends on K, the number of packets that fit in the queue, which is an integer. As we can see in Figure 2.9 the discontinuities on the value of K occur exactly at the points where the optimum rate is not smooth.

2.4.1.2 Exponential Packet Length

In this case we find the minimum distortion for an SDC system with infinite buffer length and balanced capacities, i.e., $C_1 = C_2$. The expected distortion and delay probability are calculated as in equations (2.1) and (2.2) respectively. The average end-to-end distortion for the optimal-R case and the fixed-R case [18] are both depicted in Figure 2.10 for $C_1 = C_2 = 1000$ bits/sec. Note that in [18] for small arrival rates the optimal value for the switch parameter is $q^* = 0.5$, which means that the traffic is distributed equally between the two queues. However, as the arrival rate increases, even distribution of the traffic causes both queues to become congested, since the packet-length and thus the service time are fixed. So in that case the optimal solution is to provide one queue with as much traffic as it can handle and use the other queue as a dump for the leftover traffic. As it can be seen in Figure 2.10, optimizing R can improve the performance of this system significantly. This is because in this case as the arrival rate increases, instead of only using one of the queues efficiently, the packet-lengths can be decreased to match the capability of both queues. The value of R that minimizes the distortion is demonstrated in Figure 2.11. As it was expected, the optimal packet-length gets smaller as the arrivals become faster. Note that for $\lambda \approx 0$, the value of R^* happens to be 6.006 and that is the reason the two distortion curves seem to meet at $\lambda = 0$. The optimal value for q in the optimal-R case turns out to be $q^* = 0.5$ due to the symmetry in the system.

2.4.2 DDC Results

The DDC system has previously been studied in [19] for the case where the encoders generate packets of exponential length. However, the average length $R = R_1 + R_2$ is assumed to be fixed. In other words, the optimization problem in [19] is

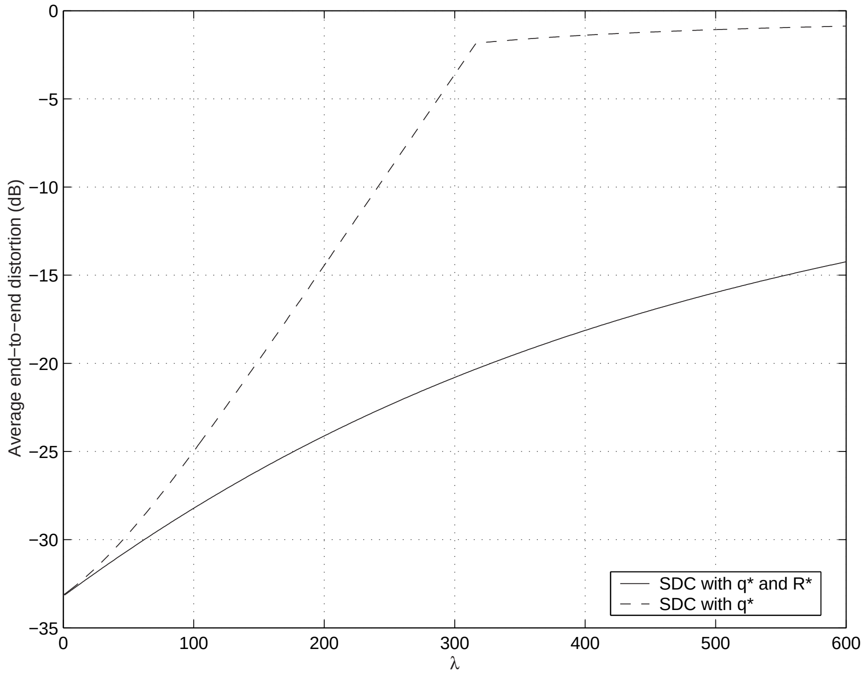


Figure 2.10: Improvement achieved by optimizing R in SDC system, $C_1 = C_2 = 1000$ bits/sec. The dashed curve is the SDC with optimized q and fixed rate of $R = 6$ (borrowed from [18]).

formulated as follows[3]

$$(\alpha^*, \delta_1^*, \delta_2^*) = \text{Argmin } \overline{D}(\alpha, \delta_1, \delta_2)$$

where $\alpha^* = R_1/(R_1 + R_2)$ is the rate ratio. In our work however, similarly to the SDC case, we add the expected packet length to the optimization parameters. The problem is therefore formulated as follows

[3] In [19] the system model is slightly different and the outputs of the two encoders pass through a switch with parameter q before entering the queues. Therefore, the actual formulation of the optimization problem is as follows.

$$(\alpha^*, \delta_1^*, \delta_2^*, q^*) = \text{Argmin } \overline{D}(\alpha, \delta_1, \delta_2, q)$$

however, it turns out that the optimum position of the switch for all values of λ is such that it routes the output of encoder i to queue i, as is the case in our work. To avoid confusion, we have removed the switch from our system model, and have stated the problem formulation of [19] without the switch parameter q.

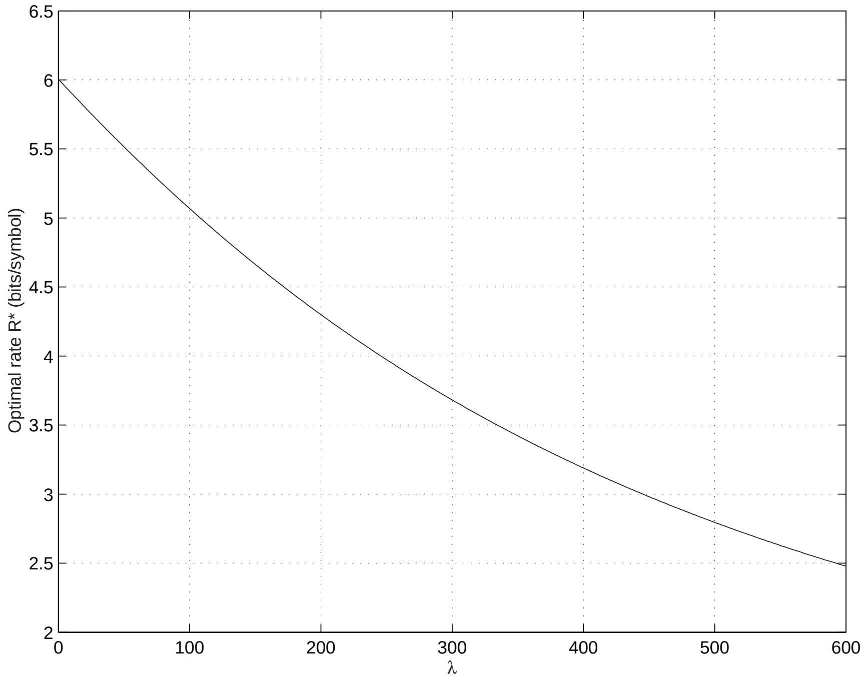


Figure 2.11: Optimal rate (R^*) in SDC with exponential packet-length, $C_1 = C_2 =$ 1000 bits/sec

$$(R^*, \alpha^*, \delta_1^*, \delta_2^*) = \text{Argmin } \overline{D}(R, \alpha, \delta_1, \delta_2) \tag{2.6}$$

We will consider two different packet-length distributions; deterministic and exponential. In the case of deterministic packet lengths, we investigate the effect of different buffer sizes on the minimum distortion when the two channels have the same capacity. In the case of exponential packet-lengths, we first consider a system with balanced capacities ($C_1 = C_2$) and find an optimum solution among all the balanced solutions, i.e., those with $R_1 = R_2$ and $\delta_1 = \delta_2$. The search for a balanced solution was motivated by the symmetry of the channels as well as the fact that the optimization in [19] is done under the constraint $\delta_1 = \delta_2$. However, we show that better performance can be achieved when we allow for asymmetric solutions. We finally look at a case where $C_1 \neq C_2$ and study the asymptotic behavior of α^* when the capacities of the two channels are not balanced.

2.4.2.1 Deterministic Packet Length

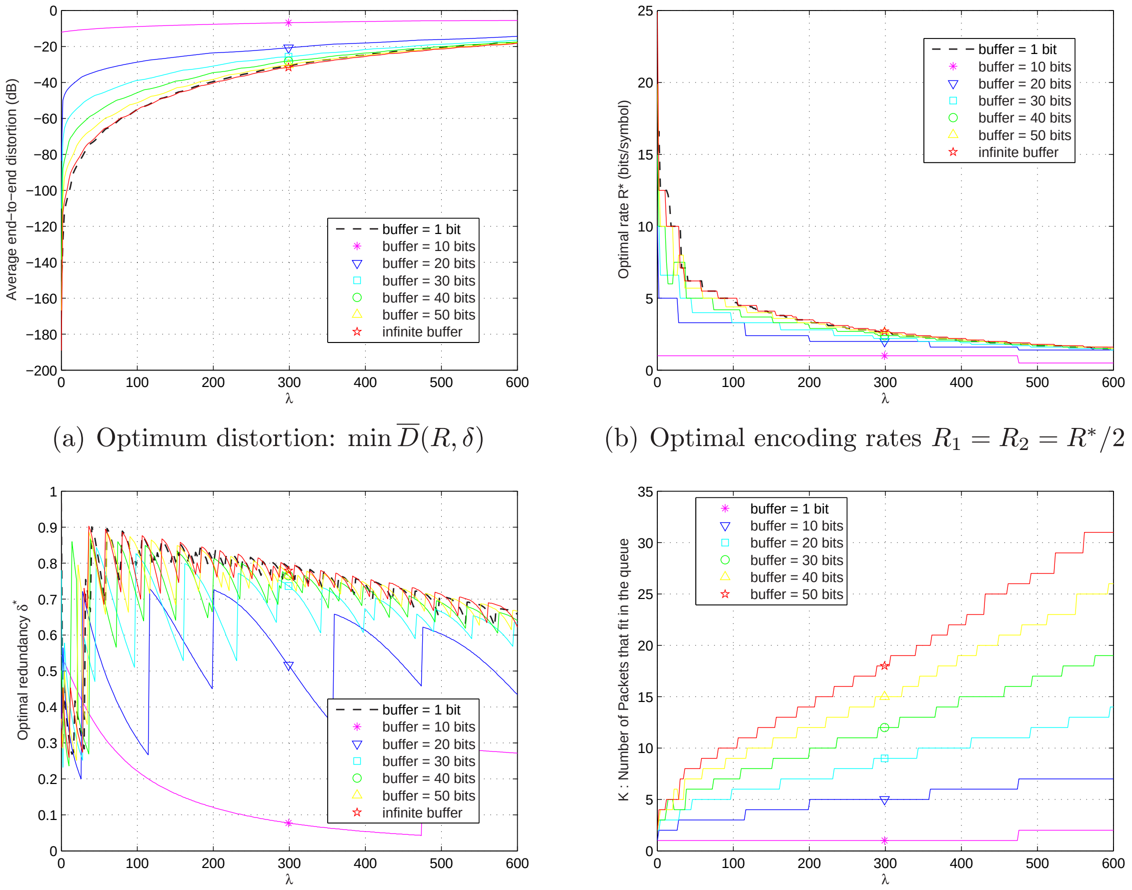
In this subsection we consider the DDC system with deterministic constant packet-lengths. Figure 2.12 shows the distortion and the optimum parameters for different buffer sizes and the infinite-buffer case. The optimization here is done under the constraints $R_1 = R_2$ and $\delta_1 = \delta_2$. It can be seen that there is a behavior very similar to what we observed in the SDC case. Namely, at large arrival rates, the distortion for the finite-buffer case with buffer size of $\hat{B} = 50$ bits is smaller than that of the infinite-buffer case. Also, similarly to the SDC case the discontinuities in the values of R and δ coincide with the discontinuities of K, the number of packets that fit in the queue.

2.4.2.2 Exponential Packet Length

Fixed R, Asymmetric Solution:

In this section our goal is to minimize the average end-to-end distortion, $\overline{D}_{DDC}$, for a fixed value of $R = R_1 + R_2$ when $C_1 = C_2 = 1000$ bits/sec. In [19] the optimization is simplified by taking $\delta_1 = \delta_2$. In other words, they find the solution to the following problem,

$$(\alpha^*, \delta^*) = \text{Argmin } \overline{D}(\alpha, \delta)$$



(a) Optimum distortion: $\min \overline{D}(R, \delta)$

(b) Optimal encoding rates $R_1 = R_2 = R^*/2$

(c) Optimal redundancies $\delta_1 = \delta_2 = \delta^*$

(d) Number of packets that fit in the queue

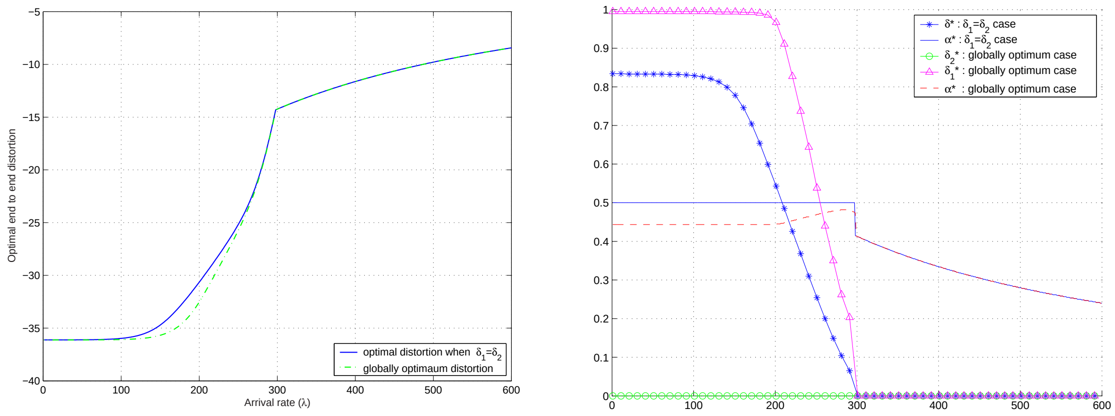
Figure 2.12: Optimum DDC with deterministic packet-lengths. $C_1 = C_2 = 1000$ bits/sec, $\delta = \delta_1 = \delta_2$, and $R = 2R_1 = 2R_2$.

where $\delta = \delta_1 = \delta_2$. Here we do the optimization for all values of δ_1 and δ_2. In other words,

$$(\alpha^*, \delta_1^*, \delta_2^*) = \text{Argmin } \overline{D}(\alpha, \delta_1, \delta_2)$$

We refer to this solution as the fixed-R *global optimum*.

Figure 2.13 shows how these two optimization problems compare. As it can be seen in Figure 2.13(a), allowing δ_1 and δ_2 to take different values in the optimization process results in a smaller distortion for arrival rates smaller than $\lambda \approx 300$



(a) Improvement achieved in the minimum distortion by optimizing δ_1 and δ_2 separately.

(b) Optimum coding parameters.

Figure 2.13: Optimum DDC with fixed-R ($R = R_1 + R_2 = 6$ bits/packet): Comparison of the two cases of $\delta_1 = \delta_2$ from [19], and the global optimum. $C_1 = C_2 = 1000$ bits/sec.

packets/sec. This is due to the fact that for $\lambda < 300$ the optimum values of δ_1 and δ_2 are different, as shown in Figure 2.13(b). This also affects the optimum value of the rate ratio α^*. As we see here, $\alpha^* < 0.5$, which means $R_1^* < R_2^*$. Therefore, packets of the second encoder have a better chance of getting to the destination and δ_i^*'s are picked so that $\delta_2^* < \delta_1^*$, so the second description has a better reconstruction quality. However for $\lambda > 300$, the globally optimum solution is to send to one queue as much traffic as it can handle and use the other as a dump for the leftover traffic. This is done here by choosing R_2^* small enough so the second queue remains stable, and letting $R_1^* = R - R_2^*$. The expected distortion for this region is given by

$$\overline{D} = d_2 \Pr[T_2 \leq \Delta] + \Pr[T_2 > \Delta]$$

therefore, we need to have $\delta_2^* = 0$ to minimize d_2, and the value of δ_1^* does not affect the distortion, and therefore, any value for δ_2^* is in fact optimum. As $\delta_1^* = \delta_2^* = 0$ is one of these globally optimum solutions, the two optimization problems are equivalent and therefore the minimum distortion is the same for the balanced and global solutions at $\lambda > 300$.

Optimum R, Symmetric Solution:

As we mentioned earlier, when the total encoding rate, R, is fixed, at hight arrival rates half of the capacity of the system is wasted, since the given encoding rate is too large with respect to channel capacities.

Adding R to our optimization parameters provides us with the possibility of decreasing the total rate as the arrivals become faster, and thus saving both queues simultaneously from getting congested. Here we first search for the symmetric solution, i.e., the optimization problem here is formulated as follows

$$(R^*, \delta^*) = \text{Argmin } \overline{D}(R, \delta)$$

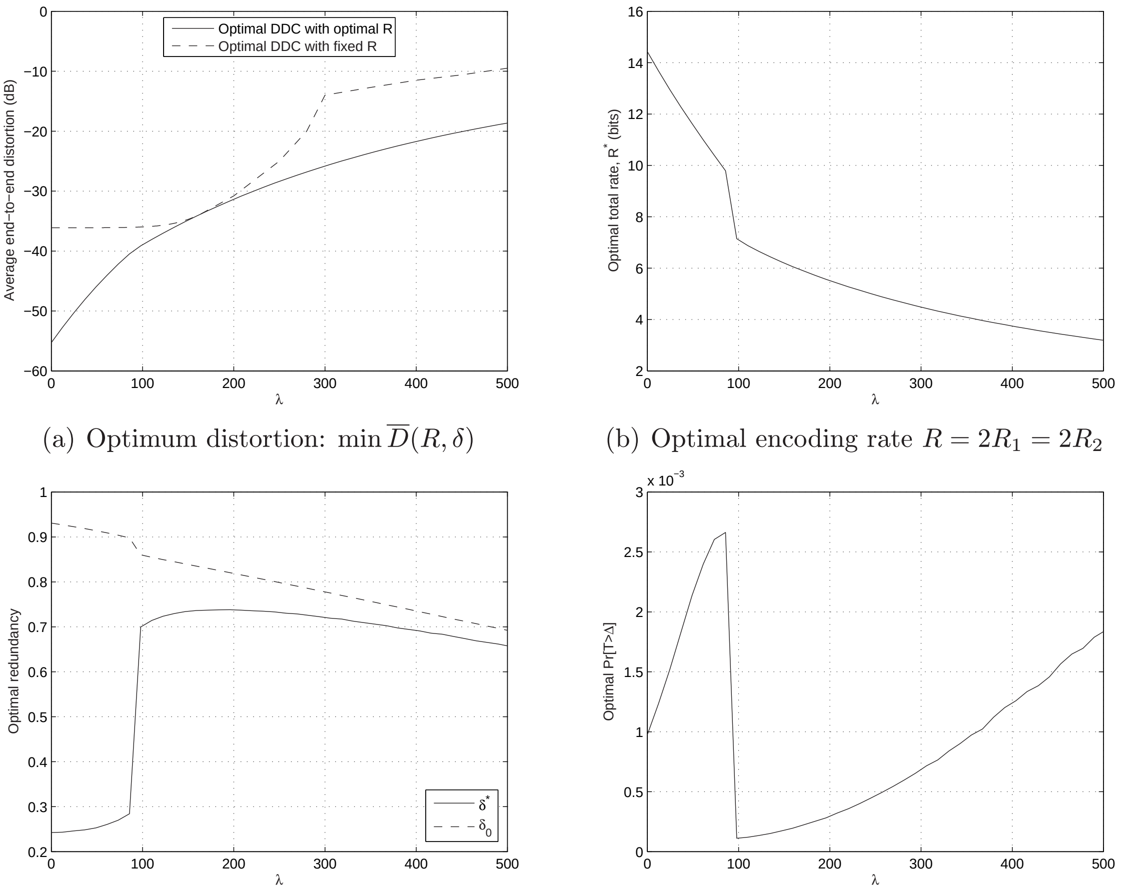
where

$$R_1 = R_2 = R/2$$

$$\delta_1 = \delta_2 = \delta$$

The optimal average end-to-end distortion is demonstrated in Figure 2.14(a) as it compares to the minimum distortion achieved in [19]. As shown in this figure, optimizing R contributes in lowering the overall distortion for all values of λ. The optimal encoding rate R^* is shown in Figure 2.14(b), where we can see that R^*

decreases with λ.



(a) Optimum distortion: $\min \overline{D}(R, \delta)$

(b) Optimal encoding rate $R = 2R_1 = 2R_2$

(c) Optimal redundancies $\delta_1 = \delta_2 = \delta^*$

(d) Delay probability

Figure 2.14: Improvement achieved by optimizing R in DDC system, $C_1 = C_2 =$ 1000 bits/sec

Note that there is a discontinuity in R^* as is depicted in Figure 2.14(b). To explain this behavior more clearly, let's define the parameter δ_0 to be the value of δ that minimizes d_0. In other words

$$\delta_0 = \text{Argmin } \ d_0(\delta)$$

Note that δ_0 is a function of R. Since d_1 and d_2 are monotonically increasing with

δ, and d_0 is minimized at δ_0, the optimal value of distortion happens at $\delta^* \leq \delta_0$. The value of δ_0 at the optimal rate R^* can be calculated as follows [15]

$$\delta_0 = \frac{1}{R^*} \log_2 \left(\frac{2^{R^*} + 2^{-R^*}}{2} \right)$$

Using equation (2.5), it is clear that if $\Pr[T_1 \leq \Delta, T_2 \leq \Delta] = 1$, then $\delta^* = \delta_0$. Figure 2.14(c) shows the value of δ^* as well as δ_0. As we can see in this figure, for small values of λ, where we expect to have smaller delay probabilities, the value of δ^* is significantly lower than δ_0, which seems counterintuitive. To explain this behavior, we take a look at the optimal delay probability ($\Pr[T > \Delta] = \Pr[T_1 > \Delta] = \Pr[T_2 > \Delta]$) as shown in Figure 2.14(d).

Note that the discontinuity occurs for R^*, δ^*, and $\Pr[T > \Delta]$ at $\lambda \approx 90$. For $\lambda < 90$ the optimization process chooses to increase the encoding rate as much as possible at the cost of getting a higher delay probability, which in turn decreases the chance of receiving both descriptions on time and thus relies mostly on receiving only one description and therefore chooses a small value for δ^* to get smaller values for d_1 and d_2. When $\lambda > 90$, the optimal solution would be one that decreases R^* to an extent that results in a small delay probability and thus increases the chance of both descriptions' on-time arrival at the destination. Therefore, as can be seen in Figure 2.14(c), the value of δ^* is very close to the value of δ_0 for $\lambda > 90$ packets/sec.

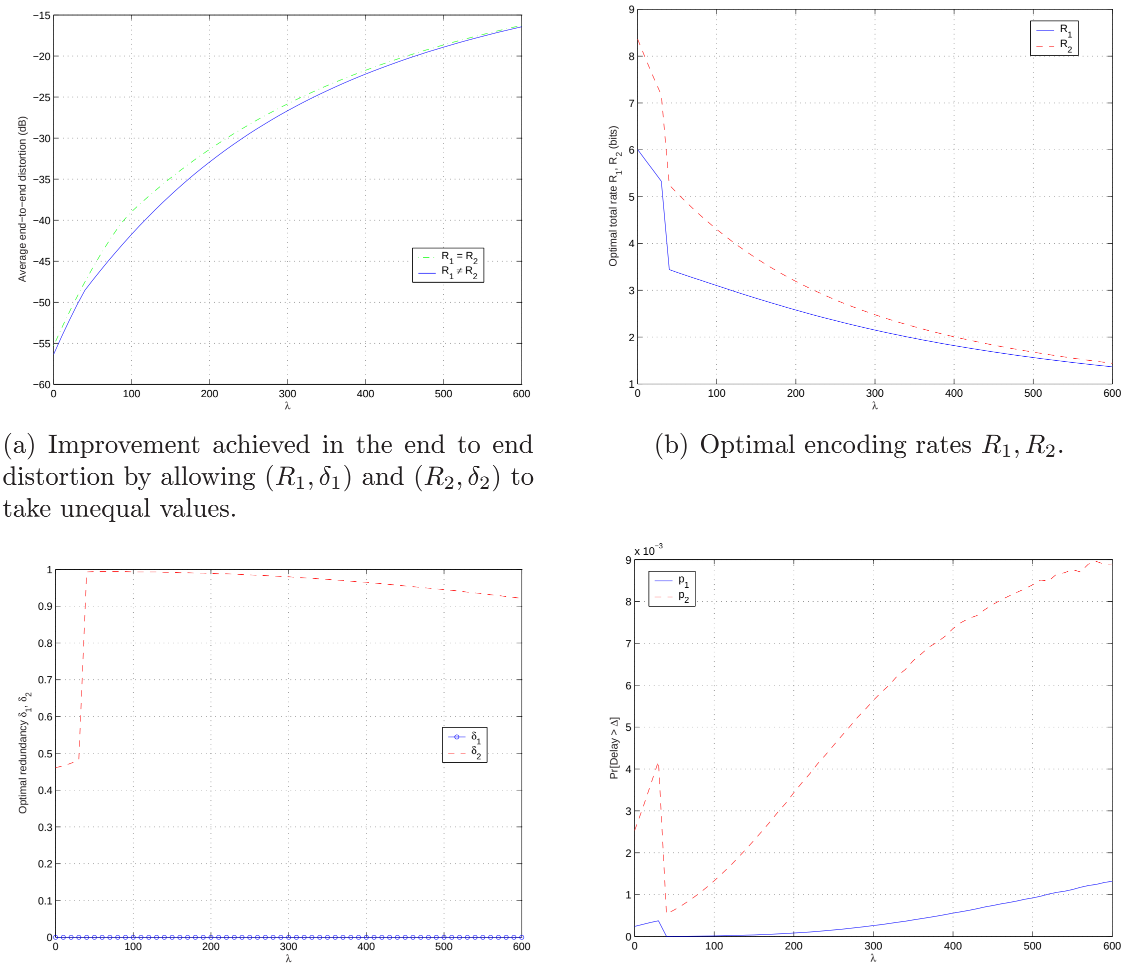
Optimum R, Asymmetric Solution:

In this section we further improve the performance of the DDC system by considering asymmetric solutions.We can formulate this problem as follows

$$(R_1^*, R_2^*, \delta_1^*, \delta_2^*) = \text{Argmin } \overline{D}(R_1, R_2, \delta_1, \delta_2)$$

which is equivalent to (2.6). Figure 2.15(a) shows the improvement achieved by allowing the pairs (R_1, δ_1) and (R_2, δ_2) to take different values. Figure 2.15(b) shows the optimum values of R_1 and R_2 which decrease as the arrival rate λ increases in order to keep the delays reasonably small. As it can be seen in this figure, $R_1^* < R_2^*$. On the other hand we can see in Figure 2.15(c) that the optimum value of δ_1 is always zero while δ_2 takes large values. This means that the first description has a minimal individual distortion ($d_1 = 2^{-2R_1}$) while the second description individually does not have a good quality ($d_2 \approx 1$) and is only used to improve the joint reconstruction quality.

To explain this behavior, let us look at the delay probabilities corresponding to the optimum value of the parameters as shown in Figure 2.15(d). Note that the probability of loosing the packets of the first encoder is significantly smaller than that of the second encoder since $R_1^* < R_2^*$. On the other hand, both of these delay probabilities are fairly small and therefore the most likely event is the event in which both descriptions make it to the destination. In other words, in equation (2.5) d_0 has the largest coefficient. This means that the optimization process will try to make d_0 as small as possible. For a given value of R_1 and R_2, this happens if we choose δ_i in a way that $\Pi = \Lambda$. The equation $\Pi = \Lambda$ has two degrees of freedom and



(a) Improvement achieved in the end to end distortion by allowing (R_1, δ_1) and (R_2, δ_2) to take unequal values.

(b) Optimal encoding rates R_1, R_2.

(c) Optimal redundancies δ_1, δ_2.

(d) Delay probabilities $p_1 = \Pr(T_1 > \Delta), p_2 = \Pr(T_2 > \Delta)$.

Figure 2.15: Optimum DDC : $(R_1^*, R_2^*, \delta_1^*, \delta_2^*) = \text{Argmin}\overline{D}_{DDC}(R_1, R_2, \delta_1, \delta_2)$, $C_1 = C_2 = 1000$ bits/sec

there are infinite values of δ_1 and δ_2 that satisfy this equation. On the other hand in equation (2.5), d_1 has the largest coefficient after d_0 and is minimized when $\delta_1 = 0$. By setting $\delta_1 = 0$ and choosing δ_2 such that $\Pi = \Lambda$, we can minimize the two most significant terms of equation (2.5). The value of δ_2 that minimizes d_0 (denoted by $\tilde{\delta}_2$) can be calculated by solving the equation $\Pi = \Lambda$ for δ_2 while replacing $\delta_1 = 0$.

$$\tilde{\delta}_2 = 1 + \log_2\left[1 + 2^{-2(R_1+R_2)} - 2^{-2R_1}\right]$$

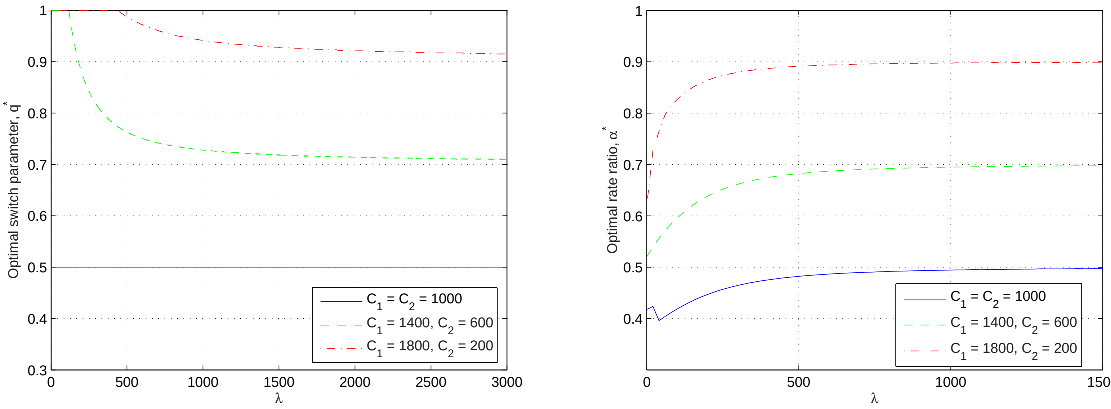
Note that setting $\delta_1 = 0$ and $\delta_2 = \tilde{\delta}_2$ minimizes the first two terms of equation (2.5) and the last term is independent of δ_1 and δ_2. However, the third term of this equation depends on δ_2 through d_2 which is an increasing function of δ_2. Therefore the actual optimum value of δ_2 is smaller than $\tilde{\delta}_2$. The amount of this difference depends on the derivatives of d_0 and d_2 at $\delta_2 = \tilde{\delta}_2$ which are functions of R_1 and R_2.

It should also be noted that there is a discontinuity in the optimal encoding rates at $\lambda \approx 33$. For $\lambda < 33$ the optimization process applies a different strategy by increasing the encoding rates as much as possible at the cost of getting higher delay probabilities. This in turn decreases the chance of receiving both descriptions on time and thus has to rely more on receiving only one description and therefore chooses a smaller value for δ_2 in order to get better values for d_2.

2.4.2.3 Unbalanced Capacities

So far, we assumed that the two channels had equal capacities ($C_1 = C_2 = 1000$ bits/sec). In this section we will study the effect of having unequal capacities and see how this affects the results.

We will consider two examples in this section for both the SDC and the DDC systems with exponential packet-lengths. In the first example we will have channels with capacities $C_1 = 1400$ and $C_2 = 600$ bits/sec, and in the second example we will have channels with capacities $C_1 = 1800$ and $C_2 = 200$ bits/sec . Note that the total capacity is kept at $C = 2000$ as in the previous cases.



(a) Optimal q^* in SDC system with $C_1 \neq C_2$.

(b) Optimal $\alpha^* = \frac{R_1^*}{R_1^*+R_2^*}$, in DDC system with $C_1 \neq C_2$.

Figure 2.16: Unbalanced Capacities: Asymptotic behavior of q^* (in SDC system) and α^* (in DDC system) for different capacity ratios. We see that both q^* and α^* converge the value of the capacity ratio as the arrival rate increases.

In the SDC system an interesting result is the behavior of the optimum switch parameter q^* as displayed in Figure 2.16(a). For small values of the arrival rate λ, we get $q^* = 1$; therefore, all packets are routed to the link with the higher capacity. This is due to the fact that for slow arrivals, at each arrival moment the queues are very likely to be empty and therefore there is no point in sending the packets to the slower link. When λ increases, the value of q^* starts to decrease to allow packets to utilize both links. Of course, at the same time R is decreasing to compensate for the queueing delay. As the arrival rate increases we see that the asymptotic behavior of q^* points to a system with balanced load. In other words, at high values of the arrival rate we obtain

$$q^* \approx \frac{C_1}{C_1 + C_2}$$

or equivalently

$$\frac{\lambda_1}{\mu_1} \approx \frac{\lambda_2}{\mu_2}$$

Figure 2.16(b) shows the value of the rate ratio α^* in the DDC case. We can see in this figure that as the arrival rate increases, the value of α^* converges to the capacity ratio of the two channels. In other words, in all the cases depicted in Figure 2.16(b) for large values of the arrival rate, λ, we have

$$\alpha^* \approx \frac{C_1}{C_1 + C_2}$$

Therefore similarly to the SDC case, the asymptotic behavior of α^* points to a balanced load distribution for high arrival rates.

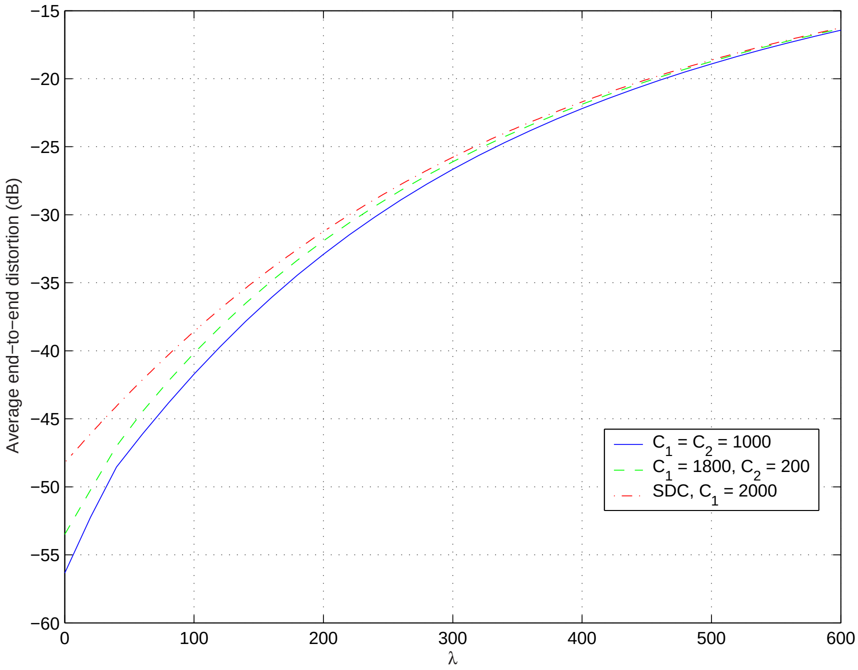


Figure 2.17: Comparison of the average distortion for different capacity ratios.

Figure 2.17 shows a comparison of the distortion achieved in three different cases for channel capacities. The total capacity in all cases is 2000 bits/sec. In the case where $C_1 = 2000$ we will basically have a single queue with capacity 2000 bits/sec and the DDC encoder simplifies to an SDC encoder, since only one description can make it to the destination. As it is shown in this figure, the case where $C_1 = C_2 = 1000$ outperforms the other two cases. On the other hand we can see that the system with a single channel has the worst performance among the three cases studied. This result is somewhat surprising since on one hand, we know from the queuing theory that splitting the capacity of a channel does not improve the overall delay of the system; on the other hand, we know that DDC encoding by itself cannot decrease the expected distortion compared to the SDC encoding. However, as we see here, using DDC encoding together with parallel routing renders the system more flexible and improves the overall performance.

This result implies that, with proper encoding, it is best to split a channel into two channels with half the capacity rather than using the entire channel at once. Naturally, the question arises that whether the distortion is a decreasing function of the number of description/channel pairs that are used for a fixed total channel capacity.

2.4.3 Comparison of the SDC and the DDC Systems

So far, we have considered all parameters that directly affect the end-to-end average distortion for a simple system consisting of two routes. Two general classes of coding schemes were considered and in each case a joint optimization problem to minimize the average end-to-end distortion was solved. In this section, we would like to quantify and compare the results obtained in the previous sections.

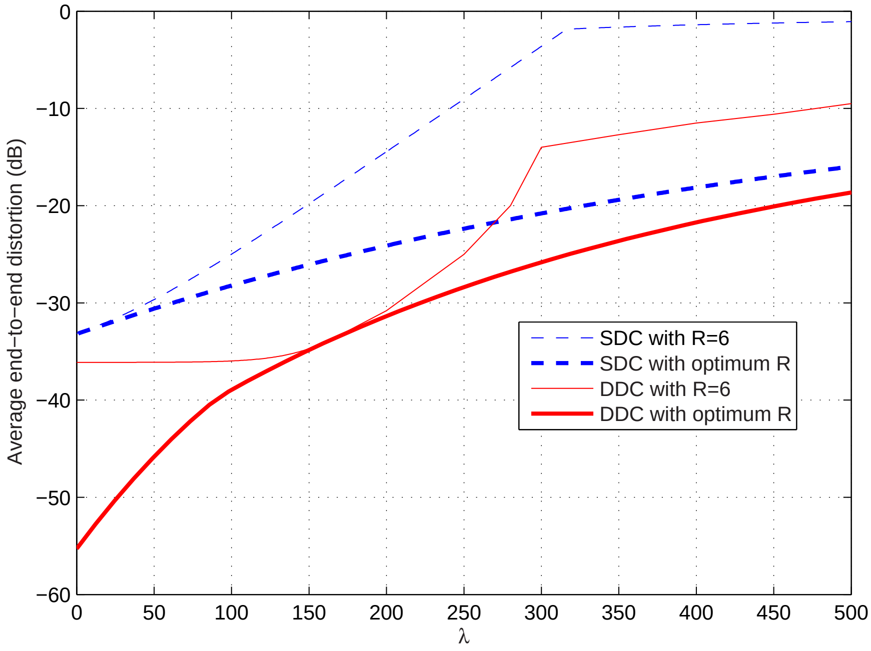


Figure 2.18: Comparison of SDC and DDC with symmetric solution in both cases of fixed R (from [19] and [18]) and optimum R. $C_1 = C_2 = 1000$ bits/sec

Figure 2.18 displays the performance of both systems (SDC and DDC) with and without optimal encoding rate for the case of symmetric solution. As observed, the DDC system with optimal rate outperforms all other systems. Similarly, the SDC system with a fixed rate has the highest distortion. Comparing the SDC system with optimal rate to the DDC system with fixed rate, we see that for arrival rates larger than $\lambda \approx 270$ the SDC system with optimal rate outperforms the DDC with fixed rate. This is because when the rate is kept fixed, unlike when it is optimized, at large arrival rates one of the queues is permanently congested and only one queue is used effectively. We also see in Figure 2.18 that in both the fixed-R and the optimum-R cases, the DDC system outperforms the SDC system.

Figure 2.19 shows the average end-to-end distortion for the SDC and DDC systems in both cases of deterministic and exponential packet-lengths when the

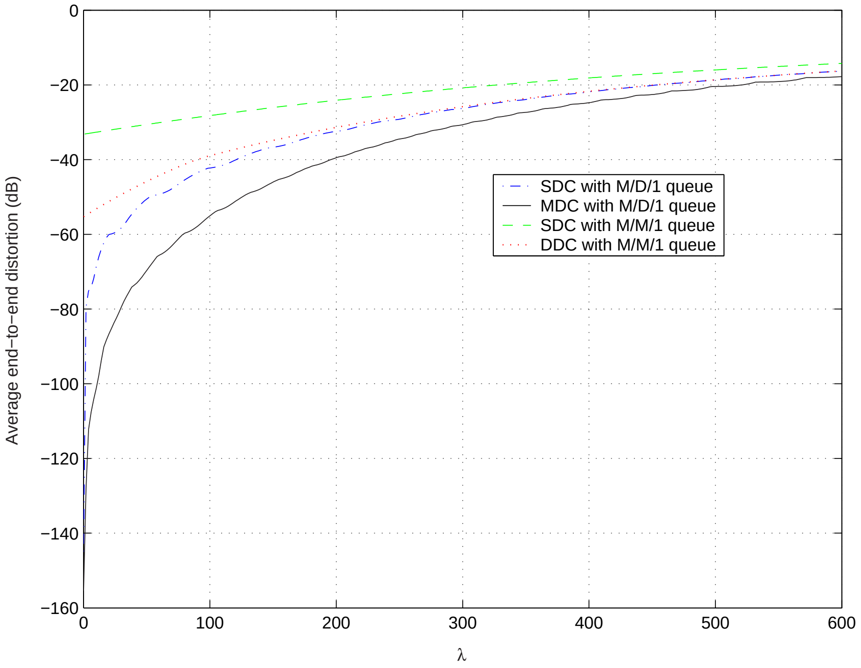


Figure 2.19: Comparison of SDC and DDC: deterministic vs. exponential packet-length distribution $C_1 = C_2 = 1000$ bits/sec

buffer has infinite capacity. As it can be seen in this figure, for a given packet-length distribution, the DDC system outperforms the SDC system. We also see that the system with deterministic packet-lengths outperforms the one with exponential packet-lengths. This can be explained using what we know from the queueing theory that among all service time distributions with the same expected value, the deterministic distribution minimizes the expected waiting time. In other words, for a given expected service time we have

$$E[W_{M/G/1}] \geq E[W_{M/D/1}]$$

since for an M/G/1 queue, the expected waiting time is given by the Pollaczek-

Khinchin (P-K) formula as follows

$$\overline{W} = \frac{\rho\overline{S}}{2(1-\rho)}\left(1 + \frac{\mathrm{var}[S]}{\overline{S}^2}\right)$$

where S is the service time random variable and therefore $\overline{S} = 1/\mu$. For a given expected service time, $\overline{S}$, the term $\frac{\rho\overline{S}}{2(1-\rho)}$ is fixed, and the term $\frac{\mathrm{var}[S]}{\overline{S}^2}$ is minimized when $\mathrm{var}[S] = 0$, which is the case for a constant service time.

Chapter 3

Distortion Control for Streaming of Delay-Sensitive Sources

3.1 Introduction

In the classical network architecture, the source symbols are encoded in the presentation layer, while the Data Link layer and/or the Transport layer take care of providing error-free transmission by the use of channel coding or retransmissions. In the case of packet erasure channels, packets traveling through the network are dropped randomly depending on the channel condition. When immediate errorfree feedback is available, the best one can do is to retransmit each dropped packet repeatedly until it reaches its destination. When dealing with delay-sensitive applications with a hard deadline for every source symbol, this approach can be modified to one which repeats the transmission of each lost packet until either the packet is expired or it has reached its destination. However, when dealing with distortion-tolerant data, this approach is no longer optimum. In this case, the overall distortion of the received message can significantly be improved by calculatedly sacrificing less significant bits corresponding to one symbol for more significant bits of another.

We consider the problem of transmitting a finite set of delay-sensitive source symbols. This is sometimes referred to as "streaming" and is used in applications such as video-on-demand where a server pre-stores encoded media and transmits it on demand to a client for playback in real time.

The problem of rate-distortion optimized streaming of layered video has been addressed under various scenarios in the literature. To the best of our knowledge, the works most closely related to the one we are presenting here have been carried out in [27] and [28]. Miao and Ortega [27] propose a low-complexity heuristic algorithm for scheduling of packet transmission. However, they assume that the number of layers representing each symbol is predetermined. Podolsky et al. [28] use a Markov chain analysis to find the optimal policy for transmitting layered media at a fixed rate over a lossy channel. However, since the state space grows exponentially with the size of the parameter space, the general solution is not presented in that paper. Other less closely related works include [29] in which a policy for dynamic allocation of bandwidth to each layer of symbol representation is found, and [30], where the complexity of rate-distortion optimized streaming is investigated. A brief survey of different approaches and results for this problem can be found in [31]. A more general survey of the contributions in the field of streaming video over the Internet can be found in [32].

In this chapter, we study the distortion-delay tradeoff by considering a source-destination pair connected through a single-link as shown in Figure 3.1. A number of source symbols are residing at the source and are to be encoded and transmitted to the destination before their corresponding deadlines. Each reconstructed symbol will result in a distortion which is a decreasing, convex function of the number of its bits received. If the bits in an encoded symbol are arranged in a decreasing order of utility, and furthermore, for decoding of a given bit, all the more significant bits are required, then the convexity of the distortion function follows. Therefore, the

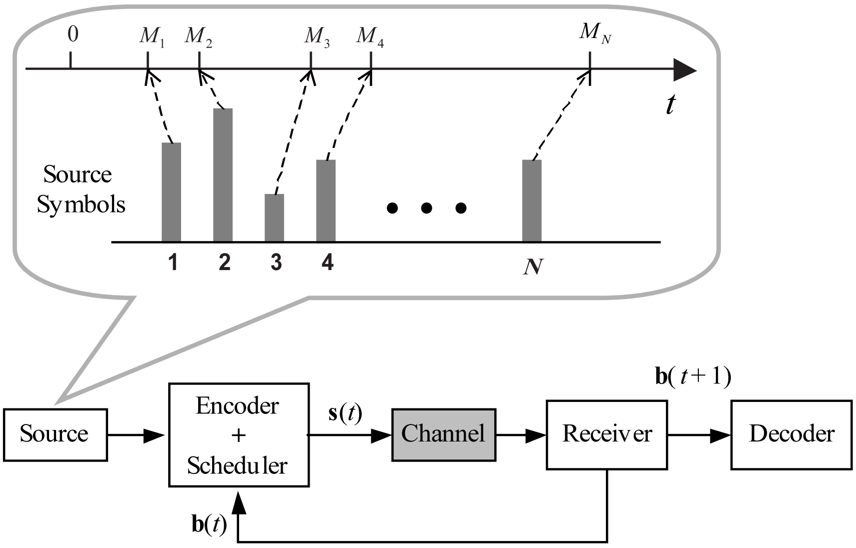


Figure 3.1: System diagram

convexity assumption on the distortion function is a reasonable one to make.

Our goal is to find a transmission policy which minimizes the total expected distortion. A policy determines what bits of what symbol to transmit at any time, based on the state of the system at that time. Finding the optimum policy depends on the values of the distortion function and, except for special trivial cases, can be computationally very costly.

We first consider a simple case where the packets are transmitted over an error-free channel. We find that when the distortion function is convex and decreasing, the optimum transmission policy is independent of the specific form of that function, and present a computationally inexpensive algorithm for solving this problem. We then proceed to solve the problem of minimum distortion streaming over packet-erasure channels by first showing that if we restrict ourself to the set of open-loop policies, the optimum policy is again independent of the form of the convex cost function. We next propose an algorithm to find a suboptimal closed-loop policy and provide numerical results to show how it improves the distortion compared to the optimal open-loop solution.

Most of the work presented here was first reported in [33–35].

3.2 The Basic Problem: Error-Free Transmission

In this section, we consider a simple scenario where a number of pre-encoded delay-sensitive source symbols, residing at the source, are to be transmitted to the destination through an error-free channel. We refer to this problem hereafter as the *Basic Problem.*

3.2.1 Problem Formulation and Notation

The Basic Problem is structured as follows.

1. N source symbols pre-encoded to packets of lengths $\gamma_1, \ldots, \gamma_N$ bits are residing at the source at time zero and must be transmitted to a receiver before they expire.

2. Each symbol i expires in M_i seconds, i.e., the bits corresponding to source symbol i transmitted after time M_i will be useless at the receiver.

3. Without loss of generality we assume that the source symbols are indexed in the order in which they expire, i.e., $M_i \leq M_{i+1}$ for $i = 1, \ldots, N-1$. We refer to M_N as the *end of the session.*

4. All encoded source symbols are available at the transmitter at the beginning

of the session and there are no arrivals to the system.

5. A total of y_i bits corresponding to source symbol i are transmitted by the end of the session

6. $d(y_i)$ is the distortion for source symbol i. The distortion function $d(\cdot)$ is convex and decreasing.

7. The channel can accommodate an error-free transmission of ω bits per second.

Note that to avoid integer constraints, we allow for fractions of bits to be transmitted, and assume that $d(\cdot)$ is defined on the set of real numbers. Given this assumption, without loss of generality, we can assume that $\omega = 1$.

Our goal is to find the number of bits corresponding to each source symbol to transmit in order to minimize the overall distortion at the end of the session, i.e., $D(\mathbf{y}) = \sum_{i=1}^{N} d(y_i)$, while meeting the deadline constraints. In other words, we wish to find the vector $\mathbf{y} = [y_1 \cdots y_N]$, which solves

$\boxed{\mathbb{P}_{\text{Basic}}}$:

$$\min_{\mathbf{y}} D(\mathbf{y}) = \sum_{i=1}^{N} d(y_i)$$

subject to

$$0 \le y_i \le \gamma_i, \quad i = 1 \cdots N \tag{3.1}$$

$$\sum_{j=1}^{i} y_j \le M_i, \quad i = 1 \cdots N \tag{3.2}$$

We denote this problem by $\mathbb{P}_{\text{Basic}}$ hereafter. The first set of constraints accounts for the fact that we cannot send more bits of a source symbol than what we have available, and the second set of constraints ensures that all transmitted bits corresponding to a source symbol are sent before that symbol expires.

3.2.2 Optimum Solution

In the following, we first prove that for a strictly convex function, $d(\cdot)$, a unique solution to $\mathbb{P}_{\text{Basic}}$ exists and is independent of the form of $d(\cdot)$. We provide a low complexity algorithm for finding the solution vector $\mathbf{y}^*$. We then show that $\mathbf{y}^*$ minimizes the distortion even if the convexity of $d(\cdot)$ is not strict; however, in this case $\mathbf{y}^*$ may no longer be the only solution to $\mathbb{P}_{\text{Basic}}$.

The following lemma, which proves a property of convex functions, is crucial to our proof.

Lemma 1 *Let $d(\cdot)$ be a strictly convex function. Let $0 \leq a < b$ and $\delta > 0$ such that $\delta < b - a$, then*

$$d(a+\delta) + d(b-\delta) < d(a) + d(b)$$

Proof For the strictly convex function $d(\cdot)$ and $\lambda \in (0,1)$ by definition we have

$$d((1-\lambda)a + \lambda b) < (1-\lambda)d(a) + \lambda d(b)$$

Similarly we can write

$$d(\lambda a + (1-\lambda)b) < \lambda d(a) + (1-\lambda)d(b)$$

Adding the corresponding sides of the above two inequalities we get

$$d(a + (b - a)\lambda) + d(\lambda(a - b) + b) < d(a) + d(b)$$

Setting $\lambda = \frac{\delta}{b-a}$ and substituting, we get

$$d(a + \delta) + d(b - \delta) < d(a) + d(b)$$

■

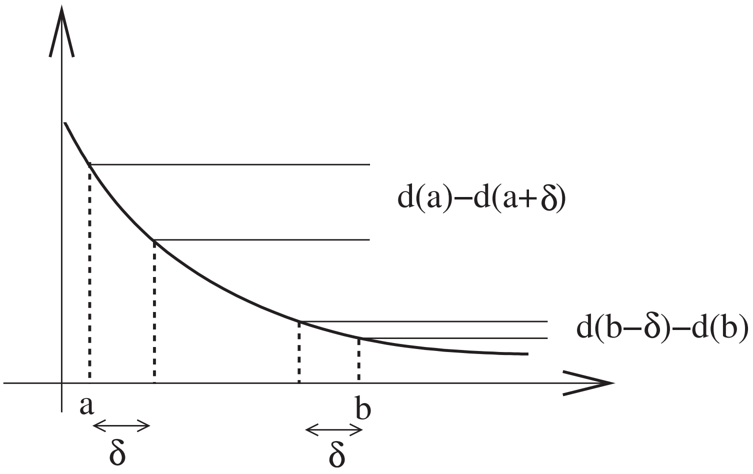


Figure 3.2: Lemma 1 illustration.

Figure 3.2 shows an example for the function $d(\cdot)$ as described in Lemma 1. As can be seen in this figure, $d(a)-d(a+\delta) < d(b-\delta)-d(b)$. Note that the function $d(\cdot)$ need not be differentiable for the lemma to hold.

Lemma 2 $\mathbb{P}_{Basic}$ *always has a solution.*

Proof Since $d(\cdot)$ is convex on the set of real numbers, it must be continuous, and therefore, $D(\cdot)$ is also continuous. On the other hand, the feasible set of $\mathbb{P}_{\text{Basic}}$

is compact, and since a continuous real-valued function attains its minimum on a compact set, a solution to $\mathbb{P}_{\text{Basic}}$ always exists. ■

Let $\mathbf{y}^*$ be a solution to $\mathbb{P}_{\text{Basic}}$. In the following lemma, we prove that if $d(\cdot)$ is strictly convex, the smallest component of $\mathbf{y}^*$ can be uniquely determined. Once the smallest component is found, we can remove this component and solve for the next smallest element of $\mathbf{y}^*$ by applying the same argument to the new $(N-1)$-dimensional problem. We can continue in this fashion until all the elements of the optimal solution $\mathbf{y}^*$ are found. Therefore, the entire vector $\mathbf{y}^*$ can be uniquely determined.

Lemma 3 *Let $\mathbf{y}^*$ be a solution to $\mathbb{P}_{Basic}$. Let $\mu_i = \frac{M_i}{i}$ for every i, and let $\hat{\jmath}_\gamma$, $\hat{\jmath}_\mu$, and $\hat{\jmath}_y$ be such that $\gamma_{\hat{\jmath}_\gamma} = \min\{\gamma_j\}_{j=1}^N$, $\mu_{\hat{\jmath}_\mu} = \min\{\mu_j\}_{j=1}^N$, and $y^*_{\hat{\jmath}_y} = \min\{y^*_j\}_{j=1}^N$. If $d(\cdot)$ is decreasing and strictly convex, then the value of $y^*_{\hat{\jmath}_y}$ is uniquely given by*

$$y^*_{\hat{\jmath}_y} = \min\{\gamma_{\hat{\jmath}_\gamma}, \mu_{\hat{\jmath}_\mu}\}$$

Proof We split the proof into two cases and prove the lemma by contradiction.

<u>**Case 1:**</u> $\gamma_{\hat{\jmath}_\gamma} \leq \mu_{\hat{\jmath}_\mu}$.

Suppose that $y^*_{\hat{\jmath}_y} \neq \gamma_{\hat{\jmath}_\gamma}$. Then $y^*_{\hat{\jmath}_y} < \gamma_{\hat{\jmath}_\gamma}$; otherwise, since $y^*_{\hat{\jmath}_y}$ is the smallest of all y^*_i, we would have $y^*_{\hat{\jmath}_\gamma} \geq y^*_{\hat{\jmath}_y} > \gamma_{\hat{\jmath}_\gamma}$ which violates inequality (3.1). We now construct a feasible vector $\hat{\mathbf{y}}$ such that $D(\hat{\mathbf{y}}) < D(\mathbf{y}^*)$, thus contradicting the optimality of $\mathbf{y}^*$. We pick $\delta > 0$ such that $y^*_{\hat{\jmath}_y} + \delta < \gamma_{\hat{\jmath}_\gamma}$ and define the N-vector $\hat{\mathbf{y}}$ as follows

$$\hat{y}_i = \begin{cases} y^*_{\hat{\jmath}_y} + \delta & , \quad i = \hat{\jmath}_y \\ y^*_i & , \quad i \neq \hat{\jmath}_y \end{cases}$$

Note that the elements of $\hat{\mathbf{y}}$ satisfy the inequalities (3.1). If $\hat{\mathbf{y}}$ meets the inequalities (3.2), since $d(\cdot)$ is decreasing, we have

$$D(\hat{\mathbf{y}}) - D(\mathbf{y}^*) = d(y^*_{\hat{\jmath}_y} + \delta) - d(y^*_{\hat{\jmath}_y}) < 0$$

and therefore, $\mathbf{y}^*$ cannot be optimum. Otherwise, if $\hat{\mathbf{y}}$ violates some of the inequalities of (3.2), we let $\hat{\imath}$ be the smallest index such that $\sum_{j=1}^{\hat{\imath}} \hat{y}_j > M_{\hat{\imath}}$ (i.e. inequality (3.2) is not met). Then since $\hat{y}_{\hat{\jmath}_y} = y^*_{\hat{\jmath}_y} + \delta < \gamma_{\hat{\jmath}_\gamma} \leq \mu_{\hat{\imath}} = M_{\hat{\imath}}/\hat{\imath}$, there exists $k \in \{1, \ldots, \hat{\imath}\}$ such that $y^*_k > y^*_{\hat{\jmath}_y} + \delta$, otherwise $\sum_{i=1}^{\hat{\imath}} \hat{y}_i \leq \hat{\imath}(y^*_{\hat{\jmath}_y} + \delta) < M_{\hat{\imath}}$. We set $\hat{y}_k = y^*_k - \delta > y^*_{\hat{\jmath}_y}$. Since $\hat{y}_k$ is present in all the inequalities in (3.2) with $i > \hat{\imath} \geq k$, adjusting $\hat{y}_k$ is sufficient to ensure that all the remaining inequalities hold. Now we redefine $\hat{\mathbf{y}}$ as follows

$$\hat{y}_i = \begin{cases} y^*_k - \delta & , \quad i = k \\ y^*_{\hat{\jmath}_y} + \delta & , \quad i = \hat{\jmath}_y \\ y^*_i & \text{otherwise} \end{cases}$$

Since $y^*_{\hat{\jmath}_y} < y^*_k$, and from the way we picked k we have $\delta < y^*_k - y^*_{\hat{\jmath}_y}$, using Lemma 1 we get

$$d(\hat{y}_k) + d(\hat{y}_{\hat{\jmath}_y}) < d(y^*_k) + d(y^*_{\hat{\jmath}_y})$$

Adding $\sum_{i \neq 1, \hat{\jmath}_y} y_i^*$ to both sides of the inequality we get

$$\sum_{i=1}^{N} d(\hat{y}_i) < \sum_{i=1}^{N} d(y_i^*)$$

Therefore,

$$D(\hat{\mathbf{y}}) < D(\mathbf{y}^*)$$

which implies that $\mathbf{y}^*$ cannot be the optimum solution unless $y_{\hat{\jmath}_y}^* = \gamma_{\hat{\jmath}_\gamma}$.

<u>**Case 2:**</u> $\gamma_{\hat{\jmath}_\gamma} > \mu_{\hat{\jmath}_\mu}$.

If $y_{\hat{\jmath}_y}^* \neq \mu_{\hat{\jmath}_\mu}$, we have to have $y_{\hat{\jmath}_y}^* < \mu_{\hat{\jmath}_\mu}$; otherwise $\sum_{i=1}^{q} y_i^* \geq q y_{\hat{\jmath}_y}^* > q\mu_{\hat{\jmath}_\mu} = M_{\hat{\jmath}_\mu}$ which violates inequality (3.2). Therefore we can pick δ such that $\hat{y}_{\hat{\jmath}_y} = y_{\hat{\jmath}_y}^* + \delta < \mu_{\hat{\jmath}_\mu}$ and the rest of the proof is similar to case 1. ■

In the following lemma we find the index of the smallest element(s) of an optimum solution.

Lemma 4 *Let* $\mathbf{y}^*, \hat{\jmath}_\gamma$, *and* $\hat{\jmath}_\mu$ *be defined as in Lemma 3. Then we have*

1. *If* $\gamma_{\hat{\jmath}_\gamma} < \mu_{\hat{\jmath}_\mu}$, *then* $y_{\hat{\jmath}_\gamma}^* = \gamma_{\hat{\jmath}_\gamma}$.

2. *If* $\gamma_{\hat{\jmath}_\gamma} > \mu_{\hat{\jmath}_\mu}$, *then* $y_1^* = \cdots = y_{\hat{\jmath}_\mu}^* = \mu_{\hat{\jmath}_\mu}$.

3. *If* $\gamma_{\hat{\jmath}_\gamma} = \mu_{\hat{\jmath}_\mu}$, *then* $y_1^* = \cdots = y_{\hat{\jmath}_\mu}^* = y_{\hat{\jmath}_\gamma}^* = \gamma_{\hat{\jmath}_\gamma}$.

Proof

1. If $\gamma_{\hat{\jmath}_\gamma} < \mu_{\hat{\jmath}_\mu}$, Lemma 3 implies $\min\{y_i^*\} = \gamma_{\hat{\jmath}_\gamma}$. If $y_{\hat{\jmath}_\gamma}^* \neq \gamma_{\hat{\jmath}_\gamma}$, then necessarily $y_{\hat{\jmath}_\gamma}^* < \gamma_{\hat{\jmath}_\gamma}$ which implies $y_{\hat{\jmath}_\gamma}^* < \min\{y_i^*\}$ which is not possible and therefore we have to have $y_{\hat{\jmath}_\gamma}^* = \gamma_{\hat{\jmath}_\gamma}$.

2. If $\gamma_{\hat{\jmath}_\gamma} > \mu_{\hat{\jmath}_\mu}$, Lemma 3 implies $\min\{y_i^*\} = \mu_{\hat{\jmath}_\mu}$. If for some $j \in \{1, \ldots, q\}$, $y_j^* \neq \mu_{\hat{\jmath}_\mu}$, then either we have $y_j^* < \mu_{\hat{\jmath}_\mu} = \min\{y_i^*\}$ which is a contradiction or we have $y_j^* > \mu_{\hat{\jmath}_\mu}$ in which case there is at least one element $k \in \{1, \ldots, q\}$ such that $y_k^* < \mu_{\hat{\jmath}_\mu} = \min\{y_i^*\}$ because otherwise $\sum_{i=1}^q y_i^* > M_{\hat{\jmath}_\mu}$ and again we reach a contradiction.

3. If $\gamma_{\hat{\jmath}_\gamma} = \mu_{\hat{\jmath}_\mu}$, both previous arguments hold.

■

Using Lemma 4 we can calculate the optimum value of the transmitted packet length y_i for some of the γ_i's. Now if we remove those γ_i's and the corresponding y_i's and M_i's from the optimization problem and adjust the remaining M_i's, the problem reduces to a similar optimization problem with fewer arguments for which the same lemma applies. Using this simple argument we can find the optimum algorithm for constructing $\mathbf{y}^*$. We call this algorithm the *base algorithm.*

Base Algorithm

1. Define $\mathcal{I}_j = \{1, \cdots, j\}$, $\forall j \in \{i\}_{i=1}^N$.

2. Let $\mathcal{I} = \mathcal{I}_N$, and $\mu_i = \frac{M_i}{i}, i \in \mathcal{I}$

3. Let $z = \min\{\{\mu_i\}_{i\in\mathcal{I}} \cup \{\gamma_i\}_{i\in\mathcal{I}}\}$

4. $\forall i \in \{j \in \mathcal{I} | \gamma_j = z\}$, set $y_i^* = z$

5. Let $\hat{\jmath} = \max\{j \in \mathcal{I} | \mu_j = z\}$

6. $\forall i \in \mathcal{I}_{\hat{\jmath}} \cap \mathcal{I}$, set $y_i^* = z$

7. Set $\mathcal{I} = \mathcal{I} - \{j | y_j^* = z\}$

8. $\forall i \in \mathcal{I}$, set $\mu_i = \frac{M_i - \sum_{j \in \mathcal{I}_i - \mathcal{I}} y_j^*}{|\mathcal{I}_i \cap \mathcal{I}|}$.

9. If $\mathcal{I} \neq \emptyset$, go back to step 3; otherwise, stop.

Note that once $\mathbf{y}^*$ is found, it suffices to send y_i^*'s in their order of expiration to ensure their timely delivery.

Theorem 1 (optimum algorithm) *For a strictly convex function $d(\cdot)$, the base algorithm finds the unique optimum solution to $\mathbb{P}_{Basic}$.*

Proof The proof of the theorem is immediately followed from Lemma 4. ■

It should be noted that if the function $d(\cdot)$ is convex but not strictly convex, the $\mathbf{y}^*$ found by the base algorithm is still optimal, although not necessarily unique. For example, if $\mu_{\hat{\jmath}_\mu} = \min\{\gamma_{\hat{\jmath}_\gamma}, \mu_{\hat{\jmath}_\mu}\}$ and $\mu_{\hat{\jmath}_\mu}$ happens to lie on a linear segment of $d(\cdot)$, then there are infinite number of optimal values for y_i , $i = 1 \cdots \hat{\jmath}_\mu$ as long as they all sum up to $M_{\hat{\jmath}_\mu}$ and stay in the same linear segment of $d(\cdot)$. The optimality of $\mathbf{y}^*$ for a merely convex $d(\cdot)$ follows form the next lemma.

Lemma 5 *Let $\mathbb{P}_d$ be a minimization problem defined as follows.*

$$\min_{\mathbf{y}} D(\mathbf{y}) = \sum_{i=1}^{N} d(y_i)$$

subject to

$$\mathbf{y} \in \mathcal{A}$$

where $\mathcal{A} \subseteq \mathrm{I\!R}^N$ for some $N \geq 1$. Let $\mathcal{D}_c$ and $\mathcal{D}_{sc}$ be the sets of all convex and all strictly convex functions defined on $\mathrm{I\!R}$, respectively ($\mathcal{D}_{sc} \subset \mathcal{D}_c$). If a given vector $\mathbf{y}^*$ solves $\mathbb{P}_d$ for all $d \in \mathcal{D}_{sc}$, then it solves $\mathbb{P}_{d_0}$ for all $d_0 \in \mathcal{D}_c$.

Proof We prove the lemma by contradiction. Suppose $\mathbf{y}^*$ does not solve $\mathbb{P}_{d_0}$ for some $d_0 \in \mathcal{D}_{\text{c}}$. Then there must be some vector $\mathbf{y}' \in \mathcal{A}$ such that

$$D_0(\mathbf{y}^*) - D_0(\mathbf{y}') > 0$$

where $D_0(\mathbf{y}) = \sum_{i=1}^N d_0(y_i)$. Let $g(\cdot)$ be a function in $\mathcal{D}_{\text{sc}}$. Define the function d_δ as follows

$$d_\delta(y) = d_0(y) + \delta g(y)$$

Since the sum of a strictly convex function with a convex function is strictly convex, we have $d_\delta \in \mathcal{D}_{\text{sc}}$ for any $\delta > 0$, and therefore, $\mathbf{y}^*$ must solve $\mathbb{P}_{d_\delta}$ for all $\delta > 0$. Let $D_\delta(\mathbf{y}) = \sum_{i=1}^N d_\delta(y_i)$, then

$$D_\delta(\mathbf{y}^*) - D_\delta(\mathbf{y}') = D_0(\mathbf{y}^*) - D_0(\mathbf{y}') + \delta(\sum_{i=1}^N g(y_i^*) - g(y_i'))$$

since $g \in \mathcal{D}_{\text{sc}}$, we have $\sum_{i=1}^N g(y_i^*) - g(y_i') < 0$, and therefore if we choose $\delta > 0$ such that

$$\delta < -\frac{D_0(\mathbf{y}^*) - D_0(\mathbf{y}')}{\sum_{i=1}^N g(y_i^*) - g(y_i')}$$

We get $D_\delta(\mathbf{y}^*) - D_\delta(\mathbf{y}') > 0$. In other words, we can always pick $\delta > 0$ in a way that $D_\delta(\mathbf{y}^*) > D_\delta(\mathbf{y}')$ which implies that $\mathbf{y}^*$ does not solve $\mathbb{P}_{d_\delta}$. ■

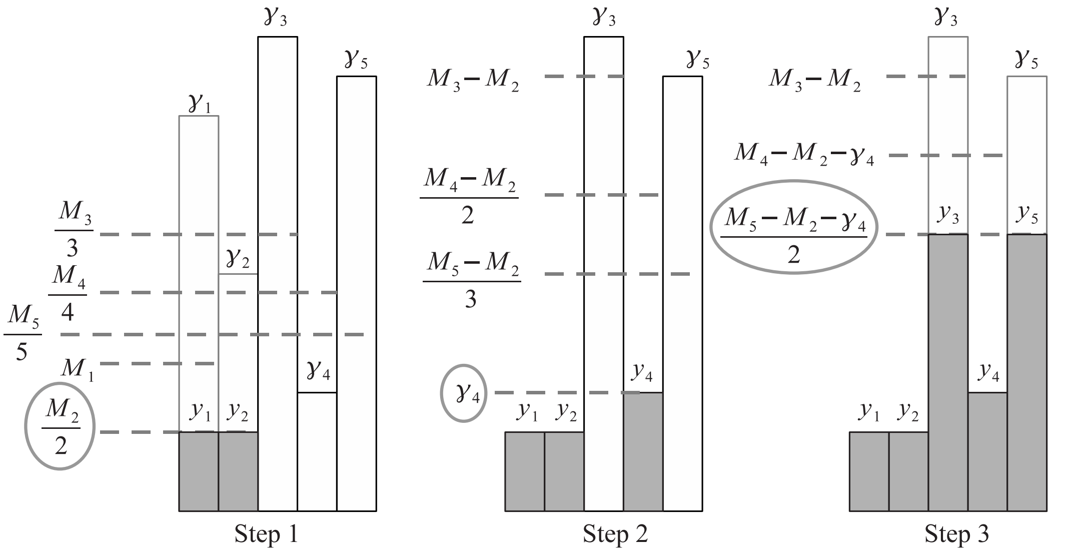


Figure 3.3: Illustration of the base algorithm for $N = 5$

Figure 3.3 illustrates the algorithm for the case of $N = 5$. In this case the optimum solution is found in three steps. In the first step, $\min\{\gamma_{\hat{\jmath}_\gamma}, \mu_{\hat{\jmath}_\mu}\} = \mu_2$ and therefore $y_1^* = y_2^* = \mu_2$. In the second step, the rest of M_i's are adjusted and this time $\min\{\gamma_{\hat{\jmath}_\gamma}, \mu_{\hat{\jmath}_\mu}\} = \gamma_4$ and so $y_4^* = \gamma_4$. And finally in the last step, $\min\{\gamma_{\hat{\jmath}_\gamma}, \mu_{\hat{\jmath}_\mu}\} = \mu_5$ and the remaining y_i^*'s are determined.

It should be noted that if instead of having a fixed rate continuous transmission we are only allowed to send data at scheduled times, we can still solve the problem using a modified version of this algorithm. To show this, let $\tau_1 < \cdots < \tau_L$ be the ordered sequence of transmit opportunities before the end of the session, i.e., $\tau_L \leq T = M_N$. Assume that at every transmit opportunity a maximum of B bits of information can be transmitted. Define n_i as the number of transmit opportunities available for the source symbol i before it expires, i.e.,

$$n_i = \max\{k | \tau_k \leq M_i\}$$

or in the case of periodic transmit opportunities $n_i = \left\lfloor \frac{M_i}{\theta} \right\rfloor$, $i = 1, \ldots, N$, where θ is the period at which the transmit opportunities occur. Now the problem can be translated to solving the following constrained minimization problem.

$$\min_{\mathbf{y}} D\left(\mathbf{y}\right)$$

subject to

$$0 \le y_i \le \gamma_i, \quad i = 1 \cdots N$$

$$\sum_{j=1}^{i} y_j \le n_i B, \quad i = 1 \cdots N$$

Since $M_i \le M_{i+1}$, we have $n_i \le n_{i+1}$ for $i = 1, \ldots, N-1$, therefore, this problem is equivalent to the previous problem and can be solved using the base algorithm with $\mu_i = \frac{n_i B}{i}$. After $\mathbf{y}^*$ is found, we send the y_i^*'s in their expiration order. For this, we might have to send some of the bits corresponding to a given source symbol in one transmit opportunity and the rest of them in the next opportunity. However, all the bits transmitted will still make it to the destination before their corresponding deadlines.

It should finally be noted that this algorithm achieves a worst case complexity of $O(N^2 \log N)$, since it involves a sorting of at most N variables in every iteration, which takes $N \log N$ operations, and a maximum of N iterations. On the other hand this is a convex minimization problem with linear constraints which can be solved by nonlinear programming. A general Linear Programming algorithm involves solving N-dimensional linear equations at each iteration which has a complexity of $O(N^3)$.

An extension of the basic algorithm presented in this section can be used for finding an optimum solution for the case where there are deterministic arrivals to the system as presented in Appendix A.1. Also, we assumed here that the channels was noise-free. In Appendix A.2 we show an example where the base algorithm can be extended to a system which uses a noisy channel.

3.3 Packet-Erasure Channel

In this section we consider a source-destination pair connected through a single-link, packet-erasure channel as shown in Figure 3.1.

3.3.1 Problem Formulation and Notation

N source symbols are residing at the source and are to be encoded and transmitted to the destination before their deadlines $M_1 \leq M_2 \leq \ldots \leq M_N$. We assume that the time is slotted and that at every time slot, B bits of information can be transmitted over the link. Each B-bit packet will either reach the destination in its entirety with probability p, or will be entirely lost otherwise. We make the simplifying assumption hereafter that the B bits transmitted at each time slot must correspond to a single symbol. In other words, we cannot send a combination of bits from different encoded source symbols in one transmission. Once we make this assumption, without loss of generality, we can assume that $B = 1$.

At each time slot t, let $\mathbf{b}(t)$ be an N-vector whose i^{th} element, $b_i(t)$, is the number of bits of the i^{th} symbol successfully received by the beginning of time slot t.

Therefore, $\mathbf{b}(t)$ indicates the state of the system at time t. We assume that $\mathbf{b}(t)$ is known to the transmitter at time t. Let $s(t)$ be the index of the symbol from which one bit is transmitted at time t. If the transmission at time slot t is successful, we get

$$\mathbf{b}(t+1) = \mathbf{b}(t) + \mathbf{e}_{s(t)}$$

where $\mathbf{e}_i$ is the unit N-vector with all but its i^{th} element set to zero. We denote the transmission policy by the function $\phi(\cdot,\cdot)$ such that

$$s(t) = \phi(\mathbf{b}(t), t)$$

We wish to find a policy ϕ which minimizes the total expected distortion while meeting the deadline constraints. In other words,

$$\min_{\phi} \sum_{i=1}^{N} E[d(b_i(T))]$$

subject to

$$z_j(T) \in \{0, 1, \ldots\} \quad , \quad j = 1, \ldots, N$$

$$\sum_{i=1}^{j} z_i(T) \leq M_j \quad , \quad j = 1, \ldots, N$$

where $d(.)$ is the distortion function, $T = M_N + 1$ is the time slot succeeding the expiration of the last packet, and $z_i(t)$ is the total number of transmission attempts on packet i before time t, i.e., $\mathbf{z}(t) = \sum_{\tau=1}^{t-1} \mathbf{e}_{s(\tau)}$. We refer to T hereafter as the *end of the session.*

Table 3.1 lists the notation used in this section.

Variable	Significance
N	number of source symbols to be transmitted
M_i	deadline of the i^{th} symbol (time-slots)
B	transmission rate (bits/time-slot)
p	probability of success
$d(.)$	distortion function (convex and decreasing)
$b_i(t)$	number of bits of i^{th} symbol successfully received by t
$s(t)$	index of the symbol from which one bit is sent in time slot t
$z_i(t)$	number of transmission attempts on i^{th} symbol by time t
T	end of the session ($= M_N + 1$)
$m_i(t)$	number of time-slots left at time t before symbol i expires

Table 3.1: General Notation

3.3.2 Optimal Open-Loop Policy

In this subsection, we search for the best policy among the subset of policies for which the decision as to which symbol is picked for transmission at each time slot does not depend on the outcome of the previous transmissions. In other words, we restrict ourself to the subset of policies which are only a function of time, i.e., for some function $\tilde{\phi}(\cdot)$,

$$\phi(\mathbf{b}(t), t) = \tilde{\phi}(t)$$

Therefore, we only need to decide on $z_i(T)$, the total number of bits corresponding to every source symbol to transmit by the end of the session, as long as we can schedule them in a way that they meet all the deadline constraints.

In this section we drop the time index from the mathematical expressions and simply use b_i and z_i in place of $b_i(T)$ and $z_i(T)$. Note that b_i is a binomial random variable with parameters z_i and p, i.e.,

$$\Pr[b_i = k] = \begin{pmatrix} z_i \\ \\ k \end{pmatrix} p^k (1-p)^{z_i - k}$$

therefore, $E[d(b_i)]$ is a function of z_i. Define the function $g : \{0, 1, ..\} \to \mathrm{I\!R}$ as follows

$$g(z_i) = E[d(b_i)]$$

The problem statement therefore simplifies to the following

$\boxed{\mathbb{P}_{\mathrm{OL}}}$:

$$\min_{\mathbf{z}} G(\mathbf{z}) = \sum_{i=1}^{N} g(z_i)$$

subject to

$$z_j \in \{0, 1, \ldots\} \quad , \quad j = 1, \ldots, N$$

$$\sum_{i=1}^{j} z_i \leq M_j \quad , \quad j = 1, \ldots, N$$

We refer to the above problem as $\mathbb{P}_{\mathrm{OL}}$ hereafter. In the following lemma, we prove that when $d(\cdot)$ is strictly convex, $g(\cdot)$ will have increasing forward differences. This property could be interpreted as an equivalent of strict convexity for discrete functions. We then use this property to find an optimum solution to $\mathbb{P}_{\mathrm{OL}}$.

Lemma 6 *Let b_z be a binomial(z, p) random variable and $d(\cdot)$ a strictly convex* *function. Then $g(z) = E[d(b_z)]$ has the following property*

$$g(z+2) - g(z+1) > g(z+1) - g(z) \ , \quad \forall z \in \{0, 1, \ldots\} \tag{3.3}$$

Proof We need to show that

$$E[d(b_{z+2})] - 2E[d(b_{z+1})] + E[d(b_z)] > 0$$

Let b_1 and b_1' be two independent binomial$(1, p)$ random variables, which are also independent of b_z. Then given the fact that the sum of two independent binomial random variables with parameters z_1, z_2 is a binomial random variable with parameter $z_1 + z_2$, we can write

$$\begin{aligned}
&E[d(b_{z+2})] - 2E[d(b_{z+1})] + E[d(b_z)] \\
&\qquad = E[d(b_z + b_1 + b_1')] - 2E[d(b_z + b_1)] + E[d(b_z)] \\
&\qquad = E[d(b_z + b_1 + b_1') - 2d(b_z + b_1) + d(b_z)] \\
&\qquad = \sum_{i=0}^{z} \sum_{j=0}^{1} \sum_{k=0}^{1} P_{ijk} \times [d(i + j + k) - 2d(i + j) + d(i)] \\
&\qquad = \sum_{i=0}^{z} \ (P_{i00} \times 0 + (P_{i01} - P_{i10})[d(i+1) - d(i)] \\
&\qquad\qquad + P_{i11}[d(i+2) - 2d(i+1) + d(i)]) > 0
\end{aligned}$$

where $P_{ijk} = \Pr(b_z = i, b_1 = j, b_1' = k)$ and the inequality of the last line is due to the strict convexity of $d(\cdot)$ and the fact that $P_{i01} = P_{i10}$. ■

It can similarly be shown that if $d(\cdot)$ is decreasing, $g(\cdot)$ is decreasing as well. In the following lemma, we prove a necessary condition for the optimality of a solution to $\mathbb{P}_{\mathrm{OL}}$ when $d(\cdot)$ is strictly convex and decreasing.

Lemma 7 *Let the N-vector $\mathbf{z}^*$ solve $\mathbb{P}_{OL}$. If $g(\cdot)$ is decreasing and meets inequality (3.3), then*

$$\sum_{i=1}^{\hat{\jmath}} z_i^* = M_{\hat{\jmath}} \tag{3.4}$$

where

$$\hat{\jmath} = \max\left\{\underset{j}{\operatorname{argmin}}\{\lfloor \frac{M_j}{j} \rfloor\}_{j=1}^{N}\right\}$$

Proof If (3.4) does not hold, in order to meet the constraints of $\mathbb{P}_{\mathrm{OL}}$, we must have $\sum_{i=1}^{\hat{\jmath}} z_i^* < M_{\hat{\jmath}}$. Then there exists some $j \leq \hat{\jmath}$ such that $z_j^* \leq \left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor$. Let r be the largest such j, i. e.,

$$r = \max\left\{ j \leq \hat{\jmath} \;\middle|\; z_j^* \leq \left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor \right\}$$

Define the N-vector $\mathbf{z}'$ as follows.

$$z'_j = \begin{cases} z_r^* + 1 & , \quad j = r \\ z_j^* & , \quad j \neq r \end{cases}$$

If $\mathbf{z}'$ meets all the constraints of $\mathbb{P}_{\mathrm{OL}}$, since $g(\cdot)$ is decreasing, we get $G(\mathbf{z}') < G(\mathbf{z}^*)$, and we reach a contradiction. Let $\tilde{\jmath}$ be the smallest index for which the constraints of $\mathbb{P}_{\mathrm{OL}}$ are not met. In other words, $\tilde{\jmath} = \min\{j \mid \sum_{i=1}^{j} z'_j > M_j\}$. Then we must have $\tilde{\jmath} > \hat{\jmath}$, since for $j < r$ we have

$$\sum_{i=1}^{j} z'_i = \sum_{i=1}^{j} z^*_i \leq M_j$$

thus, $\tilde{\jmath} \geq r$. If $r \leq \tilde{\jmath} \leq \hat{\jmath}$, we have $\sum_{i=1}^{\tilde{\jmath}} z'_i > M_{\tilde{\jmath}}$. Since we have integers on both sides of the inequality, we get

$$\sum_{i=1}^{\tilde{\jmath}} z^*_i \ + \ 1 \geq M_{\tilde{\jmath}} + 1$$

and since $\mathbf{z}^*$ must be feasible, we must have

$$\sum_{i=1}^{\tilde{\jmath}} z^*_i = M_{\tilde{\jmath}}$$

On the other hand

$$\begin{aligned}
\sum_{i=1}^{\hat{\jmath}} z^*_i &= \sum_{i=1}^{\tilde{\jmath}} z^*_i + \sum_{i=\tilde{\jmath}+1}^{\hat{\jmath}} z^*_i \\
&> M_j + (\hat{\jmath} - j)\left(\left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor + 1\right) && , \ \tilde{\jmath} > r \\
&= M_j - j\left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor + \hat{\jmath}\left(\left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor + 1\right) \\
&> M_j - j\frac{M_j}{j} + \hat{\jmath}\left(\left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor + 1\right) && , \ \frac{M_j}{j} > \left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor \\
&> M_{\hat{\jmath}}
\end{aligned}$$

which contradicts the feasibility of $\mathbf{z}^*$. Therefore, $\tilde{\jmath} > \hat{\jmath}$ and thus, by definition of $\hat{\jmath}$, we have

$$\left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor < \left\lfloor \frac{M_{\tilde{\jmath}}}{\tilde{\jmath}} \right\rfloor$$

Then there must exist some $\tilde{\imath} \in \{1, \ldots, \tilde{\jmath}\}$ such that

$$\begin{aligned} z'_{\tilde{\imath}} &\geq \left\lfloor \frac{M_{\tilde{\jmath}}}{\tilde{\jmath}} \right\rfloor + 1 \\ &\geq \left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor + 1 + 1 \\ &\geq z^*_r + 2 \end{aligned}$$

Set $z'_{\tilde{\imath}} = z^*_{\tilde{\imath}} - 1$. The new $\mathbf{z}'$ meets all the constraints of $\mathbb{P}_{\mathrm{OL}}$ and furthermore,

$$G(\mathbf{z}') - G(\mathbf{z}^*) = g(z^*_{\tilde{\imath}} - 1) + g(z^*_r + 1) - g(z^*_{\tilde{\imath}}) - g(z^*_r) < 0$$

hence a contradiction. ■

The following algorithm finds an optimum solution to $\mathbb{P}_{\mathrm{OL}}$.

Open-Loop Algorithm

1. Let $\hat{\jmath} = \max\left\{\operatorname{argmin}_j \{\lfloor \frac{M_j}{j} \rfloor\}_{j=1}^N\right\}$, $k = M_{\hat{\jmath}} - \hat{\jmath}\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \rfloor$
2. Set $z^*_j = \begin{cases} \lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \rfloor & , \ j = 1, \ldots, \hat{\jmath} - k \\ \lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \rfloor + 1 & , \ j = \hat{\jmath} - k + 1, \ldots, \hat{\jmath} \end{cases}$
3. If $\hat{\jmath} < N$, remove $\{M_j\}_{j=1}^{\hat{\jmath}}$, set $j = j - \hat{\jmath}$ for $j > \hat{\jmath}$, update the remaining M_j's, and go back to step 1. Stop otherwise.

Theorem 2 *The vector $\mathbf{z}^*$ found by the open-loop algorithm solves $\mathbb{P}_{OL}$ for any convex and decreasing function $d(\cdot)$.*

Proof We need to show that $\mathbf{z}^*$ minimizes $G(\mathbf{z})$ and meets the constraints of $\mathbb{P}_{\mathrm{OL}}$. We prove its feasibility in Lemma 8. Then, in Lemma 9, we show that for a strictly convex $d(\cdot)$, the first $\hat{\jmath}$ elements of $\mathbf{z}^*$ minimize $\sum_{j=1}^{\hat{\jmath}} g(z_j^*)$ among all integer-valued $\hat{\jmath}$-vectors $\mathbf{z}$ which meet (3.4). Since according to Lemma 7, (3.4) is a necessary condition for any vector that solves $\mathbb{P}_{\mathrm{OL}}$, this suffices to show the optimality of the first $\hat{\jmath}$ elements of $\mathbf{z}$. Furthermore, since the exact same procedure is followed for finding the remaining elements of $\mathbf{z}^*$, this completes the proof of optimality of $\mathbf{z}^*$ for strictly convex $d(\cdot)$'s. The optimality of $\mathbf{z}^*$ for merely convex functions directly follows by the use of Lemma 5. ■

The following lemma proves the feasibility of $\mathbf{z}^*$.

Lemma 8 *The N-vector $\mathbf{z}^*$ found by the open-loop algorithm meets the constraints of $\mathbb{P}_{OL}$.*

Proof $\mathbf{z}^*$'s components are, by construction, integer and non-negative. To show that they meet the deadline constraints, two possible cases need to be considered

Case 1: $j \le \hat{\jmath} - k$. In this case we have,

$$\sum_{i=1}^{j} z_i^* = \sum_{i=1}^{j} \left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor = j \left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor \le j \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \le j \frac{M_j}{j} = M_j$$

Case 2: $\hat{\jmath} - k < j \le \hat{\jmath}$. In this case we have

$$\sum_{i=1}^{j} z_i^* = \sum_{i=1}^{j} \left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor + \sum_{i=\hat{\jmath}-k+1}^{j} 1 \ = \ j \left\lfloor \frac{M_{\hat{\jmath}}}{\hat{\jmath}} \right\rfloor + (j - \hat{\jmath} + k)$$

$$\begin{aligned} &= j\left(\frac{M_{\hat{\jmath}}}{\hat{\jmath}} - \frac{k}{\hat{\jmath}}\right) + (k - \hat{\jmath} + j) \\ &\le M_j + j\left(\frac{k - \hat{\jmath} + j}{j} - \frac{k}{\hat{\jmath}}\right) \\ &= M_j + \frac{(\hat{\jmath} - j)(k - \hat{\jmath})}{\hat{\jmath}} \le M_j \end{aligned}$$

The last inequality is due to the fact that $k < \hat{\jmath}$ and $\hat{\jmath} \ge j$. ■

Note that this lemma proves the feasibility of the bit assignments in the first round of the algorithm. However, since at every round of the algorithm, the exact same procedure is followed, the same result applies for the next rounds, and therefore, the entire bit assignment is in fact feasible. In the following lemma, we prove the optimality of $\mathbf{z}^*$.

Lemma 9 *For a strictly convex function $d(\cdot)$, the $\hat{\jmath}$-vector $\mathbf{z}^*$ found in the first round of the open-loop algorithm minimizes $G_{\hat{\jmath}}(\mathbf{z}) = \sum_{j=1}^{\hat{\jmath}} g(z_j)$ among all $\hat{\jmath}$-vectors $\mathbf{z}$ for which (3.4) holds.*

Proof Let $\mathcal{Z}$ be the set of all (non-negative) integer N-vectors for which (3.4) holds. Let $\mathcal{Z}_1$ be a subset of $\mathcal{Z}$, for each member of which the difference between any two of its elements does not exceed a unit. In other words,

$$\mathcal{Z}_1 = \{\mathbf{z} \in \mathcal{Z} \mid \forall\, i, j \le \hat{\jmath}\ ,\ |z_i - z_j| \le 1\}$$

Then $\mathbf{z}^* \in \mathcal{Z}_1$, and furthermore for all $\mathbf{z} \in \mathcal{Z}_1$ we have,

$$G_{\hat{\jmath}}(\mathbf{z}) = kg(a+1) + (\hat{\jmath} - k)g(a) = G_{\hat{\jmath}}(\mathbf{z}^*)$$

where $a = \lfloor M_{\hat{\jmath}}/\hat{\jmath} \rfloor$. This is true since any vector in $\mathcal{Z}_1$ must have k elements with the value $a+1$, and $\hat{\jmath}-k$ elements with the value a.

Let $\mathbf{z}' \in \mathcal{Z} - \mathcal{Z}_1$. Then $\mathbf{z}'$ must have some elements, i and j, for which $z'_i - z'_j > 1$. Define a new vector $\mathbf{z}'' \in \mathcal{Z}$ which has the same elements as $\mathbf{z}'$ except that $z''_i = z'_i - 1$, and $z''_j = z'_j + 1$. Then, since $g(\cdot)$ is strictly convex and $z'_i - z'_j > 1$, using Lemma 1, we have

$$\begin{aligned} G_{\hat{\jmath}}(\mathbf{z}') - G_{\hat{\jmath}}(\mathbf{z}'') &= g(z'_i) - g(z'_i - 1) + g(z'_j) - g(z'_j + 1) \\ &> 0 \end{aligned}$$

Therefore, $G_{\hat{\jmath}}(\mathbf{z}'') < G_{\hat{\jmath}}(\mathbf{z}')$ and no vector in $\mathcal{Z} - \mathcal{Z}_1$ can be optimum. Since we are minimizing $G_{\hat{\jmath}}(z)$ over the set $\mathcal{Z}$ with a finite cardinality, at least one optimum solution must exist. This optimum cannot be in $\mathcal{Z} - \mathcal{Z}_1$, and therefore, it must be in its complement, $\mathcal{Z}_1$. Since $G_{\hat{\jmath}}(\mathbf{z}) = G_{\hat{\jmath}}(\mathbf{z}^*)$ for all $\mathbf{z} \in \mathcal{Z}_1$, $G(\mathbf{z}^*)$ is the minimum and $\mathbf{z}^*$ is a minimizer. ■

It should be noted here that the optimum solution found by the open-loop algorithm is independent of the form of the distortion function.

Numerical evaluation of the performance of the optimal open-loop policy is included in Section 3.4.

3.3.3 Suboptimal Closed-Loop Policy

In this section we present a computationally inexpensive closed-loop algorithm that improves the performance compared to the optimal open-loop policy. In order to do this, we employ the idea of Certainty Equivalent Controllers [36].

The certainty equivalent controller (CEC) is a suboptimal control scheme that applies, at each stage, the action that would be optimal if the random quantities were fixed at some "typical" value. The way we apply this to our problem is to find at each time slot t what would be the optimum total number of bits of each packet to be transmitted from t on, denoted by $y_i(t)$, if we fixed the random variable $b_i(T)$ to its conditional expected value, $E[b_i(T)|b_i(t)] = b_i(t) + py_i(t)$. Once we find the optimum values of $y_i(t)$, for $i = 1, \ldots, N$, we need to find some scheduling policy, $\psi(\cdot)$ that determines $s(t)$, the index of the symbol of which one bit will be transmitted at time t, based on $\mathbf{y}(t)$. In other words,

$$s(t) = \psi(\mathbf{y}(t))$$

where $\mathbf{y}(t) = [y_1(t) \cdots y_N(t)]$. So the algorithm will consist of two parts. In the first part, at every time t, we solve the following minimization problem

$$\min_{y_1(t),\ldots,y_N(t)} \sum_{i=l(t)}^{N} d(b_i(t) + py_i(t))$$

subject to

$$y_i(t) \geq 0 \quad , \quad i = l(t), \ldots, N$$

$$\sum_{i=l(t)}^{j} y_i(t) \leq m_j(t) \quad , \quad j = l(t), \ldots, N$$

where $l(t) = \min\{i|m_i(t) > 0\}$ is the smallest unexpired index. In the second part of the algorithm, we use the $\mathbf{y}(t)$ found in the first part to determine $s(t) = \psi(\mathbf{y}(t))$.

The value of $b_i(t)$ depends on $b_i(t-1)$ as well as $s(t)$, and therefore, the vector $\mathbf{b}(t)$ depends on the transmission policy and cannot take just any value. We assume that the scheduling policy $\psi(\cdot)$ is such that at any given time t we have

$$b_i(t) \geq b_{i+1}(t), \quad \text{for } \ i = l(t), \cdots, N-1$$

This assumption matches our intuition since for any two consecutive unexpired symbols, the first symbol expires no later than the second one, and therefore there is no reason to send more bits of the second one when there has been fewer successful prior transmissions of the first one. As we will see later, it is possible to find scheduling policies with the aforementioned property, and furthermore, these policies have near optimal performance. In the following, we will first find an optimum value for $\mathbf{y}(t)$. We will next propose some heuristic scheduling policies to find $s(t) = \psi(\mathbf{y}(t))$.

3.3.3.1 Part I : Finding $\mathbf{y}(t)$

For the time being we drop the index t from the above variables and simply refer to $y_i(t)$, $b_i(t)$, $m_i(t)$, and $l(t)$ as y_i, b_i, m_i, and l, respectively. Let $x_i = b_i + py_i$. We can rewrite the problem in terms of x_i as follows

$$\boxed{\mathbb{P}_{\text{CEC}}} : \qquad \min_{x_l,\dots,x_N} \sum_{i=l}^{N} d(x_i) \tag{3.5}$$

subject to

$$x_j \geq b_j \quad , \quad j = l, \ldots, N$$

$$\sum_{i=l}^{j} x_i \leq C_j \quad , \quad j = l, \ldots, N$$

Where $b_l \geq b_{l+1} \geq \cdots \geq b_N \geq 0$, and $C_j = \sum_{i=l}^{j} b_i + pm_j$, for $j \in \{l \cdots N\}$. We refer to this problem as $\mathbb{P}_{\text{CEC}}$ hereafter. Note that aside from the nonzero lower-bound constraints on x_i's, this is exactly the same problem as $\mathbb{P}_{\text{Basic}}$, with $\gamma_i = \infty$ for $i \in \{1 \cdots N\}$. Applying the base algorithm to the above problem will result in non-negative x_i's, but it does not guarantee that the lower bound constraints on x_i are met.

In what follows, we will first find the unique solution to the simple problem of finding the n-vector $\mathbf{x}^*$ which minimizes $\sum_{i=1}^{n} d(x_i)$ with a strictly convex $d(\cdot)$, if instead of the deadline constraints of $\mathbb{P}_{\text{CEC}}$ we only have the equality constraint of $\sum_{i=1}^{n} x_i = C$. We next show that if $n = \hat{\jmath} = \max\{\text{argmin}_j\{\frac{C_j}{j}\}_{j=1}^{N}\}$, the $\hat{\jmath}$-vector $\mathbf{x}^*$ will also meet the deadline constraints of $\mathbb{P}_{\text{CEC}}$. We then proceed by showing that a necessary condition for a vector $\mathbf{x}^*$ to solve $\mathbb{P}_{\text{CEC}}$ is to have $\sum_{i=1}^{\hat{\jmath}} x_i = C_{\hat{\jmath}}$. We finally use these results to find the unique optimum solution to $\mathbb{P}_{\text{CEC}}$ and then extend the results to the case where $d(\cdot)$ is merely convex.

For simplicity of presentation and without loss of generality, throughout the following proofs we set $l = 1$.

Lemma 10 *Consider the following minimization problem.*

$$\min_{x_1\cdots x_n}\sum_{i=1}^{n} d(x_i)$$

subject to

$$\sum_{i=1}^{n} x_i = C \tag{3.6}$$

$$x_i \geq b_i, \quad i = 1\cdots n \tag{3.7}$$

where $C \geq \sum_{i=1}^{n} b_i$ *is a constant,* $b_1 \geq b_2 \geq \cdots \geq b_n$*, and* $d(\cdot)$ *is a strictly convex function. Then* $\mathbf{x} = \mathbf{x}^*$ *defined below, uniquely solves this problem.*

$$x_k^* = \begin{cases} b_k & , \ k \leq \hat{k}_n^C \\ \mu_n^C(\hat{k}_n^C) & , \ \hat{k}_n^C < k \leq n \end{cases}$$

where

$$\mu_n^C(k) = \frac{C - \sum_{i=1}^{k} b_i}{n-k}$$

and $\hat{k}_n^C = \min\{k \mid b_{k+1} < \mu_n^C(k)\}$.

Proof To prove the optimality of $\mathbf{x}^*$, we need to show that it meets the constraints and minimizes the distortion. To show that it meets (3.6), we can write

$$\begin{aligned}\sum_{i=1}^{n} x_i^* &= \sum_{i=1}^{n} x_i^* = \sum_{i=1}^{\hat{k}_n^C} b_i + (n-\hat{k}_n^C)\mu_n^C(\hat{k}_n^C) \\ &= \sum_{i=1}^{\hat{k}_n^C} b_i + C - \sum_{i=1}^{\hat{k}_n^C} b_i = C\end{aligned}$$

and it meets (3.7) trivially for $k \leq \hat{k}_n^C$, and for $k > \hat{k}_n^C$ by definition of $\hat{k}_n^C$ we have

$$x_k = \mu_n^C(\hat{k}_n^C) > b_{\hat{k}_{n+1}^C} \geq b_k$$

To prove the optimality of $\mathbf{x}^*$, let the n-vector $\mathbf{x}' \neq \mathbf{x}^*$ solve the problem. Then there must be some element $\hat{\imath} < n$ for which $x'_{\hat{\imath}} > x^*_{\hat{\imath}} = \max\{b_{\hat{\imath}}, \mu_n^C(\hat{k}_n^C)\}$. On the other hand, since $\mathbf{x}'$ must meet (3.6) and (3.7), there must exist another element $\tilde{\imath} > \hat{k}_n^C$ such that $x'_{\tilde{\imath}} < x^*_{\tilde{\imath}} = \mu_n^C(\hat{k}_n^C)$. Define a new vector $\mathbf{x}''$ as follows

$$x''_i = \begin{cases} x'_{\hat{\imath}} - \delta \ , & i = \hat{\imath} \\ x'_{\tilde{\imath}} + \delta \ , & i = \tilde{\imath} \\ x'_i \ , & \text{otherwise} \end{cases}$$

where $0 < \delta < x'_{\hat{\imath}} - \max\{b_{\hat{\imath}}, \mu_n^C(\hat{k}_n^C)\}$. Then, due to the strict convexity of $d(\cdot)$, we have $\sum_{i=1}^n d(x''_i) < \sum_{i=1}^n d(x'_i)$, hence a contradiction. ■

Lemma 11 *Let $\hat{\jmath} = \max\{\operatorname{argmin}_j\{\frac{C_j}{j}\}_{j=1}^N\}$. Then the n-vector $\mathbf{x}^*$ defined in Lemma 10 meets the first n constraints of $\mathbb{P}_{CEC}$, when $n = \hat{\jmath}$ and $C \leq C_{\hat{\jmath}}$.*

Proof We want to show

$$\sum_{i=1}^{j} x_i^* \leq C_j, \quad \forall j < \hat{\jmath}$$

For $j \leq \hat{k}_{\hat{\jmath}}^C$, we have

$$\sum_{i=1}^{j} x_i^* = \sum_{i=1}^{j} b_i \leq C_j$$

For $j > \hat{k}_{\hat{\jmath}}^C$, by definition of $\mathbf{x}^*$ we have

$$\begin{aligned}
\sum_{i=1}^{j} x_i^* &= \sum_{i=1}^{\hat{k}_{\hat{\jmath}}^C} b_i + (j - \hat{k})\mu_{\hat{\jmath}}^C(\hat{k}_{\hat{\jmath}}^C) \\
&= \sum_{i=1}^{\hat{k}_{\hat{\jmath}}^C} b_i + (j - \hat{k}_{\hat{\jmath}}^C)\frac{C - \sum_{i=1}^{\hat{k}_{\hat{\jmath}}^C} b_i}{\hat{\jmath} - \hat{k}_{\hat{\jmath}}^C} \\
&\leq \sum_{i=1}^{\hat{k}_{\hat{\jmath}}^C} b_i + (j - \hat{k}_{\hat{\jmath}}^C)\frac{C_{\hat{\jmath}} - \sum_{i=1}^{\hat{k}_{\hat{\jmath}}^C} b_i}{\hat{\jmath} - \hat{k}_{\hat{\jmath}}^C} \\
&\leq \sum_{i=1}^{\hat{k}_{\hat{\jmath}}^C} b_i + (j - \hat{k}_{\hat{\jmath}}^C)\frac{C_j - \sum_{i=1}^{\hat{k}_{\hat{\jmath}}^C} b_i}{j - \hat{k}_{\hat{\jmath}}^C} \quad = \quad C_j
\end{aligned}$$

where the first inequality is because $C \leq C_{\hat{\jmath}}$, and the second and last inequality follows from $\frac{C_{\hat{\jmath}}}{\hat{\jmath}} \leq \frac{C_j}{j}$, which itself is true by definition of $\hat{\jmath}$. ■

Lemma 12 *If an N-vector $\mathbf{x}^*$ solves $\mathbb{P}_{CEC}$ for a strictly convex and decreasing $d(\cdot)$, we must have*

$$\sum_{i=1}^{\hat{\jmath}} x_i^* = C_{\hat{\jmath}}$$

where $\hat{\jmath} = \max\{\operatorname{argmin}_j\{\frac{C_j}{j}\}_{j=1}^N\}$, as in Lemma 11.

Proof Let $\sum_{i=1}^{\hat{\jmath}} x_i^* = C$. If $C \neq C_{\hat{\jmath}}$, then $C < C_{\hat{\jmath}}$ or $\mathbf{x}^*$ will not be feasible. Note that as long as the sum, C, of the first $\hat{\jmath}$ elements of $\mathbf{x}^*$ is fixed, the particular choice of each of those elements will not affect the feasibility of the rest of the elements, i.e., those with $j > \hat{\jmath}$. Therefore, given the sum C, we can determine the optimum value of each of the elements 1 through $\hat{\jmath}$ by choosing them such that $\sum_{j=1}^{\hat{\jmath}} d(x_j^*)$ is minimized, and the first $\hat{\jmath}$ inequalities are met. Using Lemmas 10 and 11, the first

$\hat{\jmath}$ elements of $\mathbf{x}^*$ are given by

$$x_k^* = \begin{cases} b_k & , \ k \leq \hat{k}_{\hat{\jmath}}^C \\ \mu_{\hat{\jmath}}^C(\hat{k}_{\hat{\jmath}}^C) & , \ \hat{k}_{\hat{\jmath}}^C < k \leq \hat{\jmath} \end{cases}$$

Let δ be such that $0 < \delta < \min\{C_{\hat{\jmath}} - C, \frac{C_{\hat{\jmath}}}{\hat{\jmath}} - x_{\hat{\jmath}}^*\}$. Define a new vector $\mathbf{x}'$ such that

$$x_i' = \begin{cases} x_{\hat{\jmath}}^* + \delta & , \ i = \hat{\jmath} \\ x_i^* & , \ i \neq \hat{\jmath} \end{cases}$$

If $\mathbf{x}'$ is feasible, since $d(\cdot)$ is decreasing, $\sum_{i=1}^N d(x_i') < \sum_{i=1}^N d(x_i^*)$ and we reach a contradiction. Otherwise, let $\tilde{\jmath} > \hat{\jmath}$ be the smallest index for which $\sum_{i=1}^{\tilde{\jmath}} x_i' > C_{\tilde{\jmath}}$. Then there must be some $\tilde{\imath}$ such that $\hat{\jmath} < \tilde{\imath} \leq \tilde{\jmath}$ and $x_{\tilde{\imath}}' > x_{\hat{\jmath}}^* + \delta$; otherwise we have $x_i' \leq x_{\hat{\jmath}}^* + \delta < \frac{C_{\hat{\jmath}}}{\hat{\jmath}}$, for all i such that $\hat{\jmath} < i \leq \tilde{\jmath}$. Therefore,

$$\begin{aligned} \sum_{i=1}^{\tilde{\jmath}} x_i' &= \sum_{i=1}^{\hat{\jmath}} x_i' + \sum_{i=\hat{\jmath}+1}^{\tilde{\jmath}} x_i' \ < \ C_{\hat{\jmath}} + (\hat{\jmath} - \tilde{\jmath})\frac{C_{\hat{\jmath}}}{\hat{\jmath}} \\ &= \hat{\jmath}\frac{C_{\hat{\jmath}}}{\hat{\jmath}} + (\hat{\jmath} - \tilde{\jmath})\frac{C_{\hat{\jmath}}}{\hat{\jmath}} \ = \ \tilde{\jmath}\frac{C_{\hat{\jmath}}}{\hat{\jmath}} \ < \ \tilde{\jmath}\frac{C_{\tilde{\jmath}}}{\tilde{\jmath}} \ = \ C_{\tilde{\jmath}} \end{aligned}$$

Define a new vector $\mathbf{x}''$ as follows

$$x_i'' = \begin{cases} x_{\hat{\jmath}}^* + \delta & , \ i = \hat{\jmath} \\ x_{\tilde{\imath}}^* - \delta & , \ i = \tilde{\imath} \\ x_i^* & , \ \text{otherwise} \end{cases}$$

Note that $x_{\tilde{\imath}}^* - \delta > x_{\hat{\jmath}}^* \geq b_{\hat{\jmath}} \geq b_{\tilde{\imath}}$, and $\sum_{i=1}^{j} x_i'' \ \leq \ C_j$. Therefore, $\mathbf{x}''$ is feasible by construction, and furthermore, due to strict convexity of $d(\cdot)$ we have $\sum_{i=1}^{N} d(x_i'') < \sum_{i=1}^{N} d(x_i^*)$, hence a contradiction. ■

The following algorithm finds a solution to $\mathbb{P}_{\text{CEC}}$ at a given time t, for $l(t) = l$. In Theorem 3 we will show that the solution found by this algorithm is optimum when $d(\cdot)$ is convex and decreasing.

CEC Algorithm: Solving $\mathbb{P}_{\textbf{CEC}}$

1. $\mu_j = \frac{C_j}{j-l+1}, \forall j \in \{l, \cdots, N\}$

2. $\hat{\jmath} = \max\{\text{argmin}_j\{\mu_j\}_{j=l}^{N}\}$

3. $\mu_{\hat{\jmath}}(k) = \begin{cases} \frac{C_{\hat{\jmath}}}{\hat{\jmath}}, & k = l-1 \\ \frac{C_{\hat{\jmath}} - \sum_{i=l}^{k} b_i}{\hat{\jmath}-k}, & k \in \{l, \cdots, N\} \end{cases}$

   $\hat{k} = \min\{k \in \{l-1, \cdots, N\} \mid b_{k+1} < \mu_{\hat{\jmath}}(k)\}$

4. Let $x_k^* = \begin{cases} b_k & , l \leq k \leq \hat{k} \\ \mu_{\hat{\jmath}}(\hat{k}) & , \hat{k} < k \leq \hat{\jmath} \end{cases}$

5. If $\hat{\jmath} < N$, let $l = \hat{\jmath} + 1,\ C_i = C_i - C_{\hat{\jmath}}, \forall i \geq l$ and go to step 1.

6. $\mathbf{y}^* = (\mathbf{x}^* - \mathbf{b})/p$

Theorem 3 *The vector $\mathbf{x}^*$ found by the CEC algorithm solves $\mathbb{P}_{CEC}$ for a convex and decreasing $d(\cdot)$.*

Proof For a strictly convex $d(\cdot)$, i.e., $d(\cdot) \in \mathcal{D}_{\text{sc}}$, by applying Lemma 12 for a given value of l, we get the following necessary condition for a solution $\mathbf{x}^*$

$$\sum_{i=l}^{\hat{\jmath}} x_i^* = C_{\hat{\jmath}}$$

Then using Lemmas 10 and 11, elements l through $\hat{\jmath}$ of $\mathbf{x}^*$ are uniquely given by

$$x_k^* = \begin{cases} b_k & , \; l \leq k \leq \hat{k} \\ \mu_{\hat{\jmath}}(\hat{k}) & , \; \hat{k} < k \leq \hat{\jmath} \end{cases}$$

where $\hat{k}$ and $\mu_{\hat{\jmath}}(\hat{k})$ are as given in the CEC algorithm. Equivalently, elements l through $\hat{\jmath}$ of $\mathbf{y}^*$ are given by

$$y_k^* = \begin{cases} 0 & , \; l \leq k \leq \hat{k} \\ \frac{\mu_{\hat{\jmath}}(\hat{k}) - b_k}{p} & , \; \hat{k} < k \leq \hat{\jmath} \end{cases}$$

Once these elements are determined, they can be removed from the problem, and using the same argument, the rest of the elements of $\mathbf{x}^*$ (and $\mathbf{y}^*$) can be derived in a similar manner, as is done in the CEC algorithm. Therefore, the CEC algorithm finds the unique solution to $\mathbb{P}_{\text{CEC}}$ when $d(\cdot) \in \mathcal{D}_{\text{sc}}$. Furthermore, using Lemma 5, we can conclude that $\mathbf{x}^*$ also solves $\mathbb{P}_{\text{CEC}}$ for a merely convex $d(\cdot)$, i.e., when $d(\cdot) \in \mathcal{D}_{\text{c}}$. Note that in this case the solution is not necessarily unique. ■

The CEC algorithm, finds the real-valued solution vector $\mathbf{y}^*(t)$. The overall algorithm is given by what we call the Closed-Loop Algorithm as follows.

Closed-Loop Algorithm

1. Let $t = 1$, and $b_i(t) = 0, m_i(t) = M_i,\ i \in \{1, \dots, N\}$
2. $l(t) = \min\{i | m_i(t) > 0\}$
3. Find $\mathbf{y}^*(t)$ using the CEC algorithm
4. $s(t) = \psi(\mathbf{y}^*(t))$
5. Set $b_{s(t+1)} = b_{s(t)} + 1$, and $m_i(t+1) = m_i(t) - 1$, for $i \in \{l(t), \dots, N\}$
6. Set $t = t + 1$. If $t < T = M_N + 1$, go to step 2.

The optimum value of $s(t)$ can be directly calculated from the integer-valued solution of the problem, if available. But the optimum integer-valued solution in fact depends on the form of the distortion function (and not just its convexity) and finding this solution can be computationally costly. Since the CEC algorithm is a heuristic algorithm, it does not make sense to go through the computation cost of finding the best integer solution, as it may not still help in getting a better final solution to the problem. Therefore, in the following subsection we propose different heuristics to calculate $s(t) = \psi(\mathbf{y}(t))$ and numerically evaluate their performance in Section 3.4. As it was explained earlier, $\psi(\cdot)$ should be such that for every t, $b_i(t) \geq b_{i+1}(t),\ i \in \{l(t), \dots, N\}$.

3.3.3.2 Part II : Finding $s(t) = \psi(\mathbf{y}^*(t))$

In the following, we will provide two different heuristics for the scheduling policy $\psi(\cdot)$. We will show that these heuristics have the following property

$$b_i(t) \geq b_{i+1}(t),\ i \in \{l(t), \ldots, N\} \tag{3.8}$$

given that the initial vector $\mathbf{b}(1)$ has the above property.

Policy CEC1: $\psi_1(\cdot)$

In this case, $s(t)$ is given by

$$s(t) = \min \left\{ \underset{i}{\operatorname{argmax}} \{y_i(t)\}_{i=l(t)}^{j^*} \right\}$$

where $j^* = \min\{j \mid \sum_{i=l(t)}^{j} y_i(t) \geq 1\}$.

Lemma 13 *If $\psi(\cdot) = \psi_1(\cdot)$ in the Closed-Loop algorithm, inequality (3.8) holds for all $t \leq T$.*

Proof We carry out the proof by induction. First, note that for $t = 1$, $b_i(t) = 0$ for all i, and therefore, (3.8) holds. Next, if (3.8) holds for a given t, if the transmission fails, we have $\mathbf{b}(t+1) = \mathbf{b}(t)$ and therefore the inequality is met at $t+1$. If, however, the transmission is successful, we have

$$b_i(t+1) = \begin{cases} b_i(t) + 1 & , \quad i = s(t) \\ b_i(t) & , \quad i \neq s(t) \end{cases}$$

therefore, at time $t+1$ we only need to show that $b_i(t+1) \geq b_{i+1}(t+1),$ for $i = s(t)-1$, or equivalently $b_{s(t)-1}(t) \geq b_{s(t)}(t)+1$. Since $\mathbf{b}(t)$ is an integer vector, this is equivalent to having

$$b_{s(t)-1}(t) > b_{s(t)}(t)$$

On the other hand, $j^* \leq \hat{\jmath}$, since

$$\begin{aligned} \sum_{i=l}^{\hat{\jmath}} y_i^*(t) &= \frac{\sum_{i=l}^{\hat{\jmath}} x_i^*(t) - \sum_{i=l}^{\hat{\jmath}} b_i(t)}{p} = \frac{C_{\hat{\jmath}} - \sum_{i=l}^{\hat{\jmath}} b_i(t)}{p} \\ &= \frac{pm_{\hat{\jmath}}}{p} = m_{\hat{\jmath}} \geq 1 \end{aligned}$$

and therefore,

$$y_k^*(t) = \begin{cases} 0 & , \ l(t) \leq k \leq \hat{k} \\ \frac{\mu_{\hat{\jmath}}(\hat{k}) - b_k(t)}{p} & , \ \hat{k} < k \leq j^* \end{cases}$$

thus, $\text{argmax}_i \{y_i(t)\}_{i=l(t)}^{j^*} = \text{argmin}_i \{b_i(t)\}_{\hat{k}+1}^{j^*}$, and hence

$$s(t) = \min \left\{ \underset{i}{\text{argmin}} \{b_i(t)\}_{\hat{k}+1}^{j^*} \right\}$$

This means that we either have $s(t) = \hat{k}+1$ if all the $b_i(t)$'s are equal for $i \in \{\hat{k}+1, \ldots, j^*\}$, in which case $b_{s(t)}(t) = b_{\hat{k}+1}(t) \leq \mu_{\hat{\jmath}}(\hat{k}) < b_{\hat{k}}(t) = b_{s(t)-1}(t)$; or else, if $b_i(t)$'s are not all equal, $s(t)$ must be such that $b_{s(t)}(t) < b_{s(t)-1}(t)$. ■

Policy CEC2: $\psi_2(\cdot)$

In this case, $s(t)$ is given by

$$s(t) = \begin{cases} j^* & , j^* \leq \hat{\jmath} \\ \min\{j|y_j(t) \geq 0\} & , j^* > \hat{\jmath} \end{cases}$$

where

$$j^* = \min\{i|y_i(t) \geq 1\}$$

Lemma 14 *If $\psi(\cdot) = \psi_2(\cdot)$ in the Closed-Loop algorithm, for any $t \leq T$, inequality (3.8) holds.*

Proof If $j^* > \hat{\jmath}$, $s(t) = \hat{k} + 1$ and we have $b_{s(t)}(t) = b_{\hat{k}+1}(t) \leq \mu_{\hat{\jmath}}(\hat{k}) < b_{\hat{k}}(t) = b_{s(t)-1}(t)$. If $j^* \leq \hat{\jmath}$, then $j^* > \hat{k}$ and

$$y_{s(t)}(t) = y_{j^*}(t) \geq 1 > y_{j^*-1}(t) = y_{s(t)-1}(t)$$

therefore,

$$\frac{\mu_{\hat{\jmath}}(\hat{k}) - b_{s(t)}(t)}{p} > \frac{\mu_{\hat{\jmath}}(\hat{k}) - b_{s(t)-1}(t)}{p}$$

so, $b_{s(t)}(t) < b_{s(t)-1}(t)$. ■

3.4 Numerical Results

In this section we compare the performance of the different algorithms discussed in the previous sections. The distortion function is assumed to be given by

$$d(R) = 2^{-2R}$$

which is the distortion-rate function when the source symbols are i.i.d. and are drawn according to a Gaussian distribution. Given this distortion function, in the open-loop case, the expected distortion is given by

$$E[d(b_i)] = \left(1 - \frac{3p}{4}\right)^{z_i}$$

which is, as expected, a convex function of z_i. For finding the actual optimum solution, we use exhaustive search.

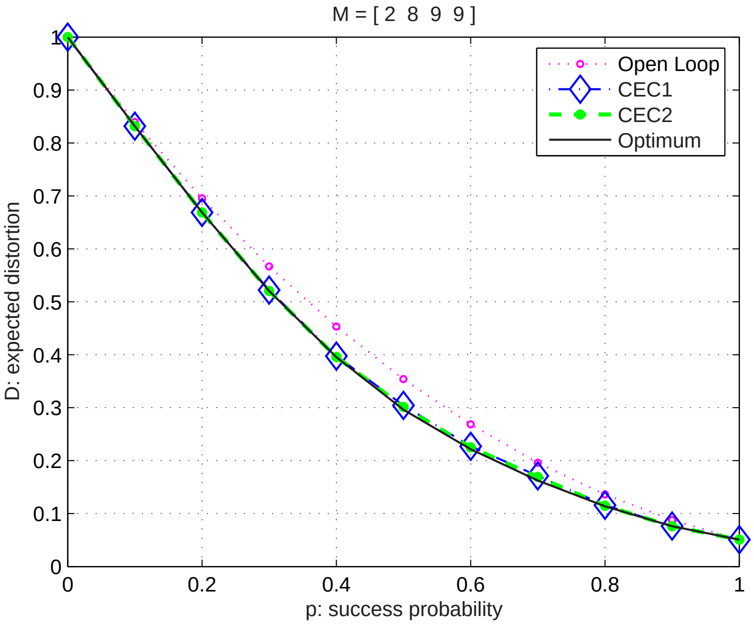


Figure 3.4: Comparison of the optimum distortion with the open-loop policy and the CEC policies for $M = [2\ 8\ 9\ 9]$

Figure 3.4 shows a comparison between the optimum solution, the open-loop algorithm, and the different heuristics for the CEC algorithm, for the case where $N =$

4 and $M = [2\ 8\ 9\ 9]$ is the vector of deadlines. As we see here, the performance of the CEC algorithm for the discussed heuristics is very close to the optimal solution.

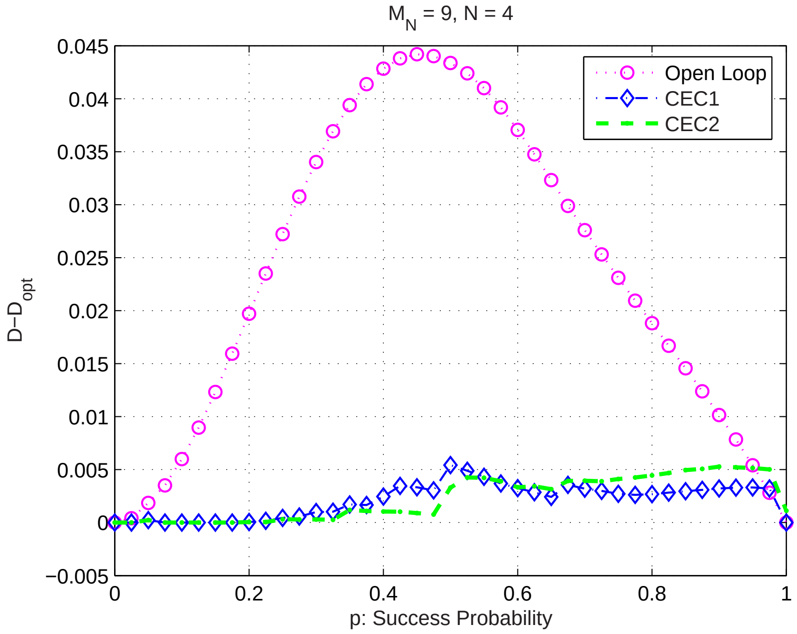


Figure 3.5: Performance evaluation of the open-loop policy and the CEC policies for $N = 4$ and $M_N = 9$

To do a more thorough evaluation of these algorithms, in Figure 3.5 we have considered all possible cases when $N = 4$ and $M_N = 9$, and have calculated the average expected distortion achieved by all the suboptimal algorithms discussed. In other words, we have solved the problem for all possible values of $M = [M_1\ M_2\ M_3\ 9]$ where $M_i \leq M_{i+1}$. So for every given policy π, we have calculated

$$\overline{D}_\pi - \overline{D}_{opt} = \frac{1}{n} \sum_{M_3=1}^{9} \sum_{M_2=1}^{M_3} \sum_{M_1=1}^{M_2} (E[D_\pi] - E[D_{opt}])$$

where $n = 165$ is the number of terms in the expression above, and π is the suboptimal policy, which can be either of the Open-Loop, CEC1, and CEC2 policies. As we see in this case, the CEC policies significantly outperform the Open-Loop policy. Furthermore, with very low computational cost, the union of these heuristics can be used to keep the distortion achieved by the CEC algorithm within about 0.004 of the optimum distortion, or in relative terms, within 5% of the optimum solution as is shown in Figure 3.6.

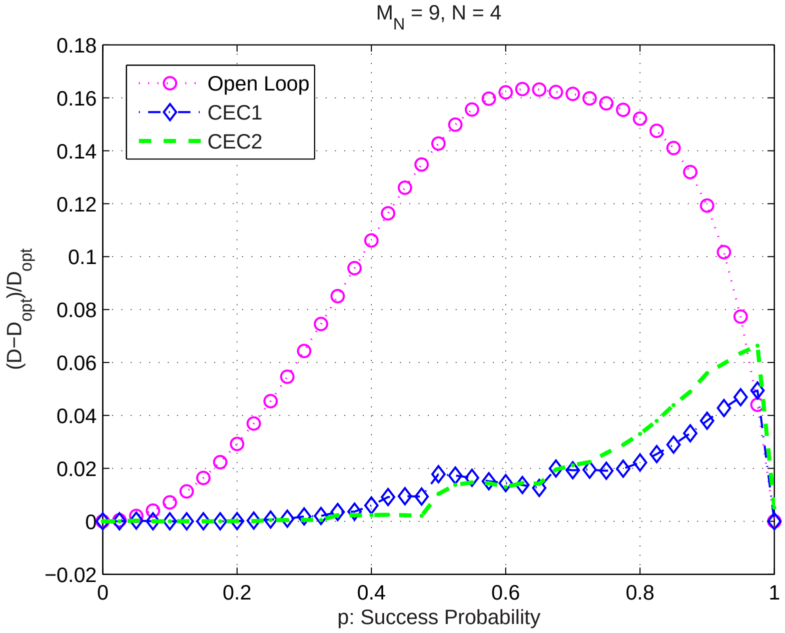


Figure 3.6: $\frac{\overline{D}_\pi - \overline{D}_{opt}}{\overline{D}_{opt}}$: Relative difference between the suboptimal policies and the optimum policy, for $N = 4$ and $M_N = 9$

Chapter 4

Conclusion and Future Work

We considered two different problems and studied the different cross-layer issues that arise due to the distortion-delay tradeoff in communication networks.

For both problems, we proposed models that are simple enough to give insight into the particular tradeoffs and cross-layer interactions considered, by eliminating other factors that might impact the network's performance. These models make the problems tractable while retaining a rich set of properties to study. In the second problem, Distortion-Control for Streaming Delay-Sensitive Applications, the simplicity of our models enabled us to derive some generalized analytical results, which distinguishes our work from other related works in the literature [27, 28]. Despite the simplifying modeling assumptions, in some cases we had to resort to simulations and heuristics for our analysis, which furthermore asserts the need for such assumptions. These models still offer many opportunities for future work, which can build upon the insights gained form our results. In the following sections, we summarize the contributions of each problem and discuss possible future directions.

4.1 Source-Coding and Parallel Routing

In Chapter 2, we presented a joint optimization problem that considers the effects of both coding and network parameters in minimizing the achieved distortion for delay-sensitive sources. In essence, we provided an illustration of a cross-layer interaction that contributes to the bridging between Networks and Information Theory. Our analysis shows that a smart encoding scheme that is done consciously of the routing can significantly contribute to lower the achieved distortion. Additional improvement can be expected if the switching module is intelligent enough to drop packets that have passed their deadline.

We outlined a trade-off between packet delay and average distortion. The average distortion is a decreasing function of the encoding rate; however, the encoding rate, which translates to packet length, in turn, determines the delay experienced by a packet. Higher encoding rates result in larger packet delays. We showed that there exists an optimal value for the encoding rate that significantly impacts the achieved distortion.

To obtain our results, we assumed the source to be memoryless and Gaussian; and we used the rate-distortion bounds obtained by Ozarow [16]. For a general memoryless source, explicit inner and outer bounds for the multiple description rate-distortion region have been found in [37]. These bounds maintain the form provided by Ozarow; therefore, we expect our analysis to be also applicable for any memoryless source.

Further studies need to be done to find the applicability of the results obtained in more realistic and complex networks (e.g., multi-hop, channel with noise, etc.). Moreover, we studied the case of double description coding. Our analysis can be further generalized for more than two encoders (i.e., Multiple Description Coding). The achievable rate-distortion bounds for such encoders have been found in [38].

4.2 Distortion Control for Streaming of Delay-Sensitive Sources

In Chapter 3, we studied optimum streaming of delay-sensitive data over both error-free and packet-erasure channels. We found an optimum transmission policy for the case of error-free channel, and showed that this policy is independent of the particular form of the distortion function when it is convex and decreasing. In the case of packet-erasure channel, we proposed an open-loop transmission policy and proved that when the rate-distortion function is convex, this policy is optimum among the set of all open-loop policies. While the general optimum policy for packet-erasure channels depends on the form of the rate-distortion function and finding it is usually computationally costly, our open-loop policy is independent of the form of the distortion function and is computationally inexpensive. We then proposed an efficient heuristic policy, which we called the CEC algorithm. We showed through numerical evaluations that the CEC policy not only outperforms the open-loop policy, but also has near optimal performance.

Further improvements to the performance of the CEC policy can be achieved by the use of what is called *policy improvement*. An example of policy improvement that is applicable to our problem is a one-step lookahead policy called the "rollout" policy [36], which at every step uses a heuristic policy (here the CEC algorithm) to calculate the cost-to-go from the next step to the end of the session for all the different possible actions that could be taken at the current state. It then picks the action with the smallest cost-to-go.

We considered streaming applications, for which the entire content is available at the transmitter at the beginning of the session. Since the CEC algorithm bases its decisions on the number of source symbols that are available for transmission and their deadlines, we expect the CEC results to be extendible, with some modification, to the case where there are arrivals to the network.

Finally, a natural extension of this line of research can be carried out into a network coding framework by considering the distortion as the performance criterion as opposed to the traditional throughput criterion.

Appendix A

A.1 Distortion Control for Streaming Delay-Sensitive Sources: Queue with Deterministic Arrivals

Consider a case where packets arrive at the queue according to a given deterministic arrival schedule. Assume that the transmit opportunities occur periodically every T seconds and B bits can be sent at every transmit opportunity. Assume furthermore that upon arrival, all packets can wait a maximum of m transmit opportunities before they expire and that they all have the same packet length γ. These assumptions are not crucial to the solution and are made to simplify the argument. We denote by a_i the number of packets that arrive during the i^{th} time slot. Since all packets that arrive in the same time slot have the same packet length and same deadline, the optimum number of bits transmitted for these packets should be equal. We denote by y_i the number of bits transmitted of every packet that has arrived in the i^{th} time slot. The following is a summary of the aforementioned variables:

We would like to minimize the total distortion for the packets arriving in the first N time slots, i.e.,

$$\min_{\mathbf{y}} \sum_{i=1}^{N} a_i d(y_i)$$

subject to

$$0 \leq y_i \leq \gamma, \quad i = 1 \cdots N$$

and

$$\sum_{i=k}^{j} a_i y_i \leq (m + j - k)B, \quad k = 1 \cdots N, \;\; j = k \cdots N$$

If we set $k = 1$ in the second set of constraints, it ensures that all packets arrive before their deadline. Setting $k > 1$ in these inequalities ensures that the solution would not require us to send packets before they arrive at the queue.

If the cost function is strictly convex, the optimum solution can be derived in a similar manner as in the case with no arrivals. Namely, if we define μ_{jk} as follows,

$$\mu_{jk} = \frac{(m + j - k)B}{\sum_{i=k}^{j} a_i}, \quad k = 1 \cdots N, \;\; j = k \cdots N$$

then the following algorithm finds the optimum values of y_i's for this problem.

Optimum Algorithm for Deterministic Arrivals Case

1. Let $\mathcal{I} = \{1, \ldots, N\}$, $\hat{\mathcal{I}} = \emptyset$, $\mathcal{J}_k = \{k, \ldots, N\}$ for $k \in \mathcal{I}$, and $\mathcal{M} = \{\mu_{jk}\}_{k \in \mathcal{I}, j \in \mathcal{J}_k}$

2. Let $z = \min\{\gamma, \min(\mathcal{M})\}$

3. If $\gamma = z$, then set $y_i = z$ for every $i \in \mathcal{I}$. STOP.

4. For every pair $(\hat{k}, \hat{\jmath})$ such that $\mu_{\hat{\jmath}\hat{k}} = z$, set $y_i^* = z$, $\forall i \in \{\hat{k}, \ldots, \hat{\jmath}\} \cap \mathcal{I}$,

5. Set $\mathcal{I} = \mathcal{I} - \{i | y_i^* = z\}$, and $\hat{\mathcal{I}} = \hat{\mathcal{I}} \cup \{i | y_i^* = z\}$

   Set $\mathcal{J}_k = \mathcal{J}_k \cap \mathcal{I}$, $\forall \; k \in \mathcal{I}$, and $\mathcal{J} = \mathcal{J} \cap \mathcal{I}$

Set $\mu_{jk} = \frac{(m+j-k)B-\sum_{i=k}^{j} y_i^* \cdot I(i \in \hat{\mathcal{I}})}{\sum_{i=k}^{j} a_i \cdot I(i \in \mathcal{I})}, \forall\ k \in \mathcal{I}, j \in \mathcal{J}_k$

Set $\mathcal{M} = \{\mu_{jk}\}_{k \in \mathcal{I}, j \in \mathcal{J}_k}$

6. Go back to step 2.

It should be noted that this solution can be easily extended to the cases where we have aperiodic transmit opportunities or different packet lengths.

A.2 Distortion Control for Streaming Delay-Sensitive Sources: Channel with Noise

Let us assume that for every y bits transmitted, only $Z(y)$ bits are received error-free according to a given distribution $f_{Z(y)}(z)$ where $z \in [0, y]$. Define $g(y) = E[d(Z(y))]$. We want to minimize

$$\sum_{i=1}^{N} E[d(Z(y_i))] = \sum_{i=1}^{N} g(y_i)$$

subject to

$$y_i \leq \gamma_i, \quad i = 1 \cdots N$$

$$\sum_{j=1}^{i} y_j \leq M_i, \quad i = 1 \cdots N$$

If for a given distribution we can show that $g(\cdot)$ is convex, then we can use the base algorithm to find the optimum transmitted packet lengths. For example, since the bits in y_i are arranged in decreasing utility order, it is reasonable to assume that the most significant bit that is affected by noise is the one that determines the distortion in the received codeword as shown in Figure A.1.

We can define the random variable $Z(y)$ as $Z(y) = y - \hat{n}$, where $\hat{n}$ is the most significant bit of y affected by noise. For a binary symmetric channel with crossover

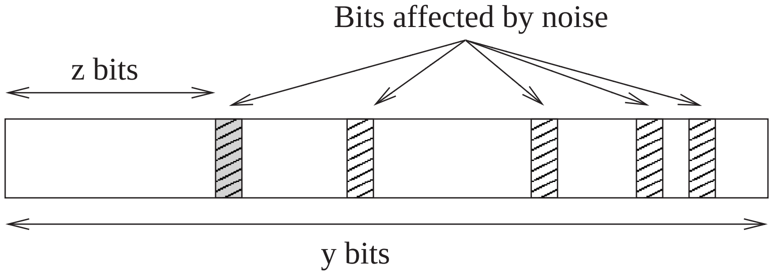


Figure A.1: Most significant bit affected by noise determines the distortion in the received packet. $d(y)$ is the distortion of the transmitted packet and $d(z)$ is the distortion of the received packet

probability p the distribution of $Z(y)$ can be written as follows

$$f_{Z(y)}(z) = \begin{cases} p(1-p)^z & , \quad z < y \\ (1-p)^z & , \quad z = y \\ 0 & , \quad \text{otherwise} \end{cases}$$

Then

$$g(y) = E[d(Z(y))] = d(y)(1-p)^y + \sum_{z=0}^{y-1} d(z)p(1-p)^z \tag{A.1}$$

Lemma 15 *If $d(\cdot)$ is decreasing and convex, $g(\cdot)$ given by (A.1) is also decreasing and convex.*

Proof If $d(\cdot)$ decreasing, we have

$$\begin{aligned} g(y) - g(y+1) &= d(y)(1-p)^y + p\sum_{z=0}^{y-1} d(z)(1-p)^z - d(y+1)(1-p)^{y+1} \\ &\quad -p\sum_{z=0}^{y} d(z)(1-p)^z \\ &= d(y)(1-p)^y - d(y+1)(1-p)^{y+1} - pd(y)(1-p)^y \\ &= d(y)(1-p)^{y+1} - d(y+1)(1-p)^{y+1} > 0 \end{aligned}$$

and therefore, $g(\cdot)$ is also decreasing.

To prove that $g(\cdot)$ is convex, we must show $g(y) - g(y+1) \leq g(y-1) - g(y)$. We have

$$g(y) - g(y+1) - g(y-1) + g(y) = (1-p)^{y+1}[d(y) - d(y+1)] - (1-p)^{y}[d(y-1) - d(y)]$$

If $d(\cdot)$ is convex, we have $d(y) - d(y+1) \leq d(y-1) - d(y)$. Since $(1-p)^{y+1} \leq (1-p)^{y}$ for $0 \leq p \leq 1$, therefore

$$g(y) - g(y+1) - g(y-1) + g(y) < 0$$

and the proof is complete. ■

Therefore, for the example above, we can use the base algorithm of Section 3.2 to find the optimum packet lengths. Note that this will determine the optimum packet lengths off-line and does not use a feedback to determine which bits are affected by noise. Therefore the solution found is only optimum among the open-loop solutions and can be improved in presence of feedback.

Bibliography

[1] Toufik Ahmed, Ahmed Mehaoua, Raouf Boutaba, and Youssef Iraqi. Adaptive packet video streaming over IP networks: A cross-layer approach. *IEEE Journal on Selected Areas in Communications*, 23(2):385–401, February 2005.

[2] Jiantao Wang, Lun Li, Steven H. Low, and John C. Doyle. Cross-layer optimization in TCP/IP networks. *IEEE/ACM Transactions on Networking*, 13(3):582–595, June 2005.

[3] Junshan Zhang, Anthony Ephremides, Lang Tong, Andrea Goldsmith, and P. R. Kumar. ICC panel on defining cross-layer design in wireless networking, 2003. http://www.eas.asu.edu/∼junshan/ICC03panel.html.

[4] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson. Cross-layer design for wireless networks. *IEEE Communications Magazine*, 41(10):74–80, 2003.

[5] U.C. Kozat, I. Koutsopoulos, and L. Tassiulas. A framework for cross-layer design of energy-efficient communication with QoS provisioning in multi-hop wireless networks. In *IEEE INFOCOM*, volume 2, pages 1446–1456, March 2004.

[6] T. Yoo, E. Setton, X. Zhu, A. Goldsmith, and B. Girod. Cross-layer design for video streaming over wireless ad hoc networks. *IEEE Wireless Communications*, 12(4):59–65, August 2005.

[7] A. Ephremides. Energy concerns in wireless networks. *IEEE Wireless Communications*, 9(4):48–59, August 2002.

[8] A. Goldsmith and B. Girod. Design challenges for energy-constrained ad hoc wireless networks,. *IEEE Wireless Communications*, 9(4):8–27, August 2002.

[9] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 3(1):74–85, January 2004.

[10] S. Toumpis and A. Goldsmith. Performance, optimization, and cross-layer design of media access protocols for wireless ad hoc networks. In *IEEE ICC*, volume 3, pages 2234–2240, May 2003.

[11] Mung Chiang. To layer or not to layer: balancing transport and physical layers in wireless multihop networks. In *IEEE INFOCOM*, volume 4, pages 2525–2536, March 2004.

[12] Lai-U Choi, Wolfgang Kellerer, and Eckehard Steinbach. On cross-layer design for streaming video delivery in multiuser wireless environments. *EURASIP Journal on Wireless Communications and Networking*, 2006, 2006.

[13] H.S. Witsenhausen. On source networks with minimal breakdown degradation. *Bell System Technical Journal*, 59(6):1083–1087, July–August 1980.

[14] J. Wolf, A. Wyner, and J. Ziv. Source coding for multiple descriptions. *Bell System Technical Journal*, 59(8):1417–1426, October 1980.

[15] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin. Multiple description coding in networks with congestion problem. *IEEE Transactions on Information Theory*, 47(3):891–903, March 2001.

[16] L. Ozarow. On a source coding problem with two channels and three receivers. *Bell Syst. Tech. J.*, 59(10):84–91, December 1980.

[17] A. A. El Gamal and T. Cover. Achievable rates for multiple descriptions. *IEEE Transactions on Information Theory*, 28(6):851–857, November 1982.

[18] K. Sayrafian-Pour, M. Alasti, A. Ephremides, and N. Farvardin. The effects of multiple routing on the average end-to-end distortion. In *Proceedings of IEEE ISIT*, Washington, DC, June 2001.

[19] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin. Multiple description coding, a network perspective. In *Proceedings of IEEE ISIT*, Lausanne, Switzerland, June 30–July 5 2002.

[20] A. Faridi, K. Sayrafian-Pour, M. Alasti, and A. Ephremides. *"Source Coding and Parallel Routing", Advances in Network Information Theory*, volume 66 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, chapter I, pages 3–24. American Mathematical Society, Providence, RI, 2004.

[21] A. Faridi and A. Ephremides. Source coding and parallel routing: Exploring the asymmetries. In *Conference on Information Sciences and Systems*, The Johns Hopkins University, Baltimore, MD, March 2005.

[22] S. Resnick. *Adventures in Stochastic Processes.* Cambridge, MA: Birkhaüser, 1992.

[23] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., New York, NY, 1991.

[24] R. W. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice Hall, 1989.

[25] H. Takagi. *Queueing Analysis, Volume 2: Finite Systems.* North-Holland, 1993.

[26] A. Leon-Garcia. *Probability and Random Processs for Electrical Engineering.* Reading, MA: Addison Wesley, 1994.

[27] Z. Miao and A. Ortega. Optimal scheduling for the streaming of scalable media. *Proc. of Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA*, Nov. 2000.

[28] M.G. Podolsky, S. McCanne, and M. Vetterli. Soft ARQ for layered streaming media. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, Special Issue on Multimedia Signal Processing*, 27(1–2):81–97, 2001.

[29] D. Saparilla and K.W. Ross. Optimal streaming of layered video. In *INFOCOM (2)*, pages 737–746, 2000.

[30] M. Roder, J. Cardinal, and R. Hamzaoui. On the complexity of rate-distortion optimal streaming of packetized media. In *Proc. Data Compression Conference*, pages 192–201, March 2004.

[31] P.A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. *IEEE Transactions on Multimedia*, 8(2):390–404, April 2006.

[32] D. Wu, Y. Thomas, W. Zhu, Y.Q. Zhang, and J.M. Peha. Streaming video over the internet: approaches and directions. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):282–300, March 2001.

[33] A. Faridi and A. Ephremides. Distortion control for queues with deadlines. In *Data Compression Conference*, pages 312–321, Snowbird, UT, March 2006.

[34] A. Faridi and A. Ephremides. Distortion control for packet-erasure channels. In *IEEE Statistical Signal Processing Workshop*, Madison, WI, August 2007.

[35] A. Faridi and A. Ephremides. Distortion control for delay sensitive sources. *IEEE Transactions on Information Theory*, June 2007. submitted.

[36] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, Belmont, MA, 3rd edition, 2005.

[37] R. Zamir. Gaussian codes and shannon bounds for multiple descriptions. *IEEE Transactions on Information Theory*, 45(7):2629–2636, November 1999.

[38] R. Venkataramani, G. Kramer, , and V.K. Goyal. Multiple description coding with many channels. *IEEE Transactions on Information Theory*, 49(9):2106–2114, September 2003.