

Distortion Control for Delay-Sensitive Sources

Azadeh Faridi, *Member, IEEE*, and Anthony Ephremides, *Fellow, IEEE*

Abstract—We investigate the problem of finding minimum-distortion policies for streaming delay-sensitive but distortion-tolerant data. We consider cross-layer approaches which exploit the coupling between presentation and transport layers. We make the natural assumption that the distortion function is convex and decreasing. We focus on a single source-destination pair and analytically find the optimum transmission policy when the transmission is done over an error-free channel. This optimum policy turns out to be independent of the exact form of the convex and decreasing distortion function. Then, for a packet-erasure channel, we analytically find the optimum open-loop transmission policy, which is also independent of the form of the convex distortion function. We then find computationally efficient closed-loop heuristic policies and show, through numerical evaluation, that they outperform the open-loop policy and have near optimal performance.

Index Terms—Delay-sensitive, distortion, scheduling, source-coding, streaming.

I. INTRODUCTION

IN communication networks, traditionally, the source encoding is done independently of the network conditions. In the classical network architecture, the source symbols are encoded in the presentation layer, while the transport layer takes care of providing error-free transmission by the use of channel coding or retransmissions. Packets traveling through the network may fail to be correctly received due to various reasons, such as buffer overflows or wireless fades. Such losses are often modeled by packet-erasure channels, where packets are randomly dropped. When immediate error-free feedback is available, the best one can do is to retransmit each lost 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 transmits previously stored encoded media on demand to a client for playback in real time.

The problem of rate-distortion optimized streaming of layered media 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 has been carried out in [1] and [2]. Podolsky *et al.* in [1] 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. In [2], Miao and Ortega propose a low-complexity heuristic algorithm for scheduling layers of different frames for transmission, which picks, at any transmission opportunity, the “most important” layer according to an estimate of the received distortion. However, they assume that the number of layers representing each frame is predetermined.

Other less closely related works include [3], [4], and [5]. In [3] and [4], a group of interdependent data units are to be transmitted to a destination before their deadlines. Each data unit has a given number of transmit opportunities. The goal of the scheduler is to decide at each transmit opportunity whether to transmit or not to transmit the data unit corresponding to that opportunity. A cost is incurred for every transmission and a policy is optimum if there is no other policy that results in a smaller distortion while incurring a smaller cost. In [5], a policy for dynamic allocation of bandwidth to each layer of symbol representation is found. Although the problem we are considering here is conceptually related to the ones considered in these works, as we will see, the actual problem statement and also its mathematical model are very different.

A brief survey of different approaches and results for related problems can be found in [3]. A more general survey of the contributions in the field of streaming video over the Internet can be found in [6].

In this paper, we study the distortion-delay tradeoff by considering a source-destination pair connected through a single-link as shown in Fig. 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

Manuscript received June 11, 2007; revised December 13, 2007. This work was supported in part by a subaward from the U.S. Army, prime award W911NF-05-1-0246. Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under The Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The material in this paper was presented in part at the Data Compression Conference, Snowbird, UT, March 2006 and the IEEE Workshop on Statistical Signal Processing, Madison, WI, August 2007.

A. Faridi was with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA. She is now with the Department of Information and Communication Technologies at Universitat Pompeu Fabra, 08003 Barcelona, Spain.

A. Ephremides is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA.

Communicated by W. Szpankowski, Associate Editor for Source Coding.

Color versions of Figures 4 and 5 in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIT.2008.926338

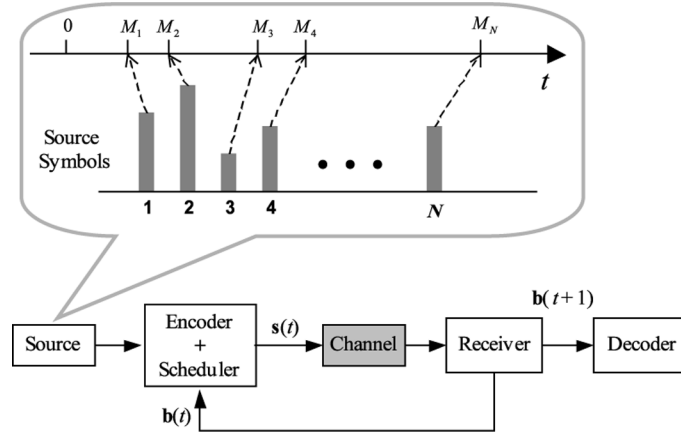


Fig. 1. System diagram.

function of the number of its bits received. It should be noted here that, 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 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 which bits corresponding to which source 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. We conclude the paper by discussing possible improvements and future directions.

The basis of the work presented here was first reported in [7] and [8].

II. 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*.

A. Problem Formulation and Notation

The Basic Problem is structured as follows.

- 1) N source symbols pre-encoded to packets of lengths $\gamma_1, \dots, \gamma_N$ 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 $i = 1, \dots, N-1$. 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 $\mathbf{y} = [y_1 \dots y_N]$, which solves

$$\boxed{\mathbb{P}_{\text{Basic}}} \quad \min_{\mathbf{y}} D(\mathbf{y}) = \sum_{i=1}^N d(y_i)$$

subject to

$$0 \leq y_i \leq \gamma_i, \quad i = 1 \dots N \quad (1)$$

$$\sum_{j=1}^i y_j \leq M_i, \quad i = 1 \dots N. \quad (2)$$

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

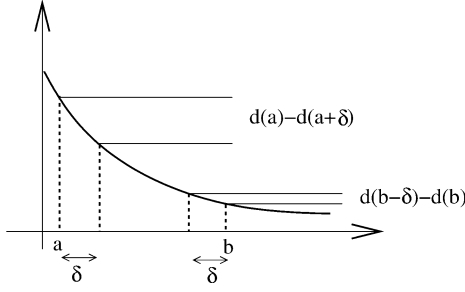


Fig. 2. Lemma 1 illustration.

set of constraints ensures that all transmitted bits corresponding to a source symbol are sent before that source symbol expires.

B. 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 and $a, b, \delta \in \mathbb{R}$ be such that $0 \leq \delta < b - a$. 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

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

□

Fig. 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: $\mathbb{P}_{\text{Basic}}$ 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 $\mathbb{P}_{\text{Basic}}$. Let $\mu_i = \frac{M_i}{i}$ for every i , and let \hat{j}_γ , \hat{j}_μ , and \hat{j}_y be such that $\gamma_{\hat{j}_\gamma} = \min\{\gamma_j\}_{j=1}^N$, $\mu_{\hat{j}_\mu} = \min\{\mu_j\}_{j=1}^N$, and $y_{\hat{j}_y}^* = \min\{y_j^*\}_{j=1}^N$. If $d(\cdot)$ is decreasing and strictly convex, then the value of $y_{\hat{j}_y}^*$ is uniquely given by

$$y_{\hat{j}_y}^* = \min\{\gamma_{\hat{j}_\gamma}, \mu_{\hat{j}_\mu}\}.$$

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

Case 1: $\gamma_{\hat{j}_\gamma} \leq \mu_{\hat{j}_\mu}$.

Suppose that $y_{\hat{j}_y}^* \neq \gamma_{\hat{j}_\gamma}$. Then $y_{\hat{j}_y}^* < \gamma_{\hat{j}_\gamma}$; otherwise, since $y_{\hat{j}_y}^*$ is the smallest of all y_i^* , we would have $y_{\hat{j}_\gamma}^* \geq y_{\hat{j}_y}^* > \gamma_{\hat{j}_\gamma}$ which violates inequality (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 $y_{\hat{j}_y}^* + \delta < \gamma_{\hat{j}_\gamma}$ and define the N -vector $\hat{\mathbf{y}}$ as follows:

$$\hat{y}_i = \begin{cases} y_{\hat{j}_y}^* + \delta, & i = \hat{j}_y \\ y_i^*, & i \neq \hat{j}_y. \end{cases}$$

Note that the elements of $\hat{\mathbf{y}}$ satisfy the inequalities (1). If $\hat{\mathbf{y}}$ meets the inequalities (2), since $d(\cdot)$ is decreasing, we have

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

and therefore, \mathbf{y}^* cannot be optimum. Otherwise, if $\hat{\mathbf{y}}$ violates some of the inequalities of (2), we let \hat{i} be the smallest index such that $\sum_{j=1}^{\hat{i}} \hat{y}_j > M_{\hat{i}}$ (i.e., inequality (2) is not met). Then since $\hat{y}_{\hat{j}_y} = y_{\hat{j}_y}^* + \delta < \gamma_{\hat{j}_\gamma} \leq \mu_{\hat{i}} = M_{\hat{i}}/\hat{i}$, there exists $k \in \{1, \dots, \hat{i}\}$ such that $y_k^* > y_{\hat{j}_y}^* + \delta$, otherwise $\sum_{i=1}^{\hat{i}} \hat{y}_i \leq \hat{i}(y_{\hat{j}_y}^* + \delta) < M_{\hat{i}}$. We set $\hat{y}_k = y_k^* - \delta > y_{\hat{j}_y}^*$. Since \hat{y}_k is present in all the inequalities in (2) with $i > \hat{i} \geq k$, adjusting \hat{y}_k is sufficient to ensure that all the remaining inequalities hold. Now we redefine $\hat{\mathbf{y}}$ as follows

$$\hat{y}_i = \begin{cases} y_k^* - \delta, & i = k \\ y_{\hat{j}_y}^* + \delta, & i = \hat{j}_y \\ y_i^*, & \text{otherwise.} \end{cases}$$

Since $y_{\hat{j}_y}^* < y_k^*$, and from the way we picked k we have $\delta < y_k^* - y_{\hat{j}_y}^*$, using Lemma 1 we get

$$d(\hat{y}_k) + d(\hat{y}_{\hat{j}_y}) < d(y_k^*) + d(y_{\hat{j}_y}^*).$$

Adding $\sum_{i \neq 1, \hat{j}_y} y_i^*$ 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 $y_{j_y}^* = \gamma_{j_\gamma}$.

Case 2: $\gamma_{j_\gamma} > \mu_{j_\mu}$.

If $y_{j_y}^* \neq \mu_{j_\mu}$, we have to have $y_{j_y}^* < \mu_{j_\mu}$; otherwise $\sum_{i=1}^{\hat{j}} y_i^* \geq \hat{j} y_{j_y}^* > \hat{j} \mu_{j_\mu} = M_{j_\mu}$ which violates inequality (2). Therefore we can pick δ such that $\hat{y}_{j_y} = y_{j_y}^* + \delta < \mu_{j_\mu}$ and the rest of the proof is similar to case 1. \square

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

Lemma 4: Let \mathbf{y}^* , \hat{j}_γ , and \hat{j}_μ be defined as in Lemma 3. Then we have

- 1) If $\gamma_{j_\gamma} < \mu_{j_\mu}$, then $y_{j_\gamma}^* = \gamma_{j_\gamma}$.
- 2) If $\gamma_{j_\gamma} > \mu_{j_\mu}$, then $y_1^* = \dots = y_{j_\mu}^* = \mu_{j_\mu}$.
- 3) If $\gamma_{j_\gamma} = \mu_{j_\mu}$, then $y_1^* = \dots = y_{j_\mu}^* = y_{j_\gamma}^* = \gamma_{j_\gamma}$.

Proof:

- 1) If $\gamma_{j_\gamma} < \mu_{j_\mu}$, Lemma 3 implies $\min\{y_i^*\} = \gamma_{j_\gamma}$. If $y_{j_\gamma}^* \neq \gamma_{j_\gamma}$, then necessarily $y_{j_\gamma}^* < \gamma_{j_\gamma}$ which implies $y_{j_\gamma}^* < \min\{y_i^*\}$ which is not possible and therefore we have to have $y_{j_\gamma}^* = \gamma_{j_\gamma}$.
- 2) If $\gamma_{j_\gamma} > \mu_{j_\mu}$, Lemma 3 implies $\min\{y_i^*\} = \mu_{j_\mu}$. If for some $j \in \{1, \dots, \hat{j}\}$, $y_j^* \neq \mu_{j_\mu}$, then either we have $y_j^* < \mu_{j_\mu} = \min\{y_i^*\}$ which is a contradiction or we have $y_j^* > \mu_{j_\mu}$ in which case there is at least one element $k \in \{1, \dots, \hat{j}\}$ such that $y_k^* < \mu_{j_\mu} = \min\{y_i^*\}$ because otherwise $\sum_{i=1}^{\hat{j}} y_i^* > M_{j_\mu}$ and again we reach a contradiction.
- 3) If $\gamma_{j_\gamma} = \mu_{j_\mu}$, both previous arguments hold. \square

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 $\mathcal{I}_j = \{1, \dots, j\}$, $\forall j \in \{i\}_{i=1}^N$.
- 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) $\forall i \in \{j \in \mathcal{I} \mid \gamma_j = z\}$, set $y_i^* = z$.
- 5) Let $\hat{j} = \max\{j \in \mathcal{I} \mid \mu_j = z\}$.
- 6) $\forall i \in \mathcal{I}_j \cap \mathcal{I}$, set $y_i^* = z$.
- 7) Set $\mathcal{I} = \mathcal{I} - \{j \mid y_j^* = z\}$.
- 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 $\mathbb{P}_{\text{Basic}}$.

Proof: The proof of the theorem is immediately followed from Lemma 4. \square

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 $\mu_{j_\mu} = \min\{\gamma_{j_\gamma}, \mu_{j_\mu}\}$ and μ_{j_μ} happens to lie on a linear segment of $d(\cdot)$, then there are infinite number of optimal values for y_i , $i = 1 \dots \hat{j}_\mu$ as long as they all sum up to M_{j_μ} and stay in the same linear segment of $d(\cdot)$. The optimality of \mathbf{y}^* for a merely convex $d(\cdot)$ follows from 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 $\mathcal{A} \subseteq \mathbb{R}^N$ 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 \mathbb{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 $d_0 \in \mathcal{D}_c$. 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}_{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 $d_\delta \in \mathcal{D}_{sc}$ for any $\delta > 0$, and therefore, \mathbf{y}^* must solve \mathbb{P}_{d_δ} for all $\delta > 0$. Let $D_\delta(\mathbf{y}) = \sum_{i=1}^N d_\delta(y_i)$, then

$$\begin{aligned} D_\delta(\mathbf{y}^*) - D_\delta(\mathbf{y}') &= D_0(\mathbf{y}^*) - D_0(\mathbf{y}') \\ &\quad + \delta \left(\sum_{i=1}^N g(y_i^*) - g(y_i') \right) \end{aligned}$$

since $g \in \mathcal{D}_{sc}$, 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_δ} , and we reach a contradiction. \square

Fig. 3 illustrates the algorithm for the case of $N = 5$. In this case the optimum solution is found in three steps. In the

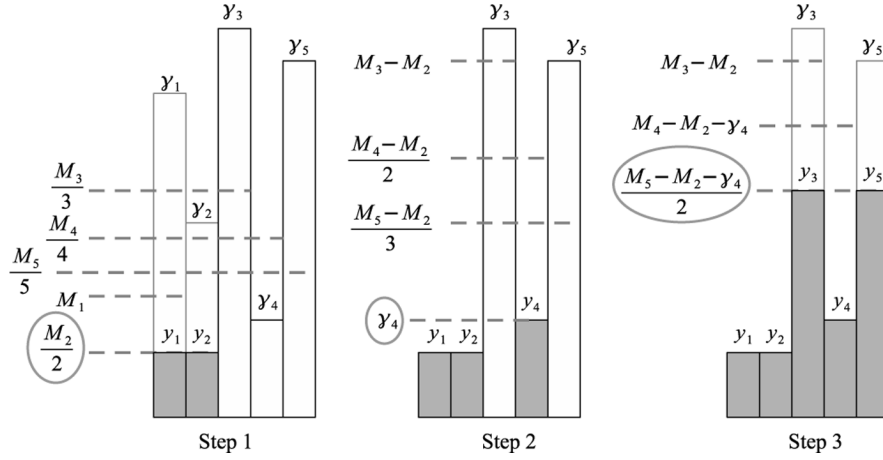


Fig. 3. Illustration of the base algorithm for $N = 5$.

first step, $\min\{\gamma_{j_\gamma}, \mu_{j_\mu}\} = \mu_2$ and therefore $y_1^* = y_2^* = \mu_2$. In the second step, the rest of M_i 's are adjusted and this time $\min\{\gamma_{j_\gamma}, \mu_{j_\mu}\} = \gamma_4$ and so $y_4^* = \gamma_4$. And finally in the last step, $\min\{\gamma_{j_\gamma}, \mu_{j_\mu}\} = \mu_5$ 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 $\tau_1 < \dots < \tau_L$ 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 left 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 $n_i = \lfloor \frac{M_i}{\theta} \rfloor$, $i = 1, \dots, N$, where θ is the period at which the transmit opportunities occur. Now the problem can be translated to solving the following constrained minimization problem:

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

subject to

$$\begin{aligned} 0 &\leq y_i \leq \gamma_i, \quad i = 1 \dots N \\ \sum_{j=1}^i y_j &\leq n_i B, \quad i = 1 \dots N. \end{aligned}$$

Since $M_i \leq M_{i+1}$, we have $n_i \leq n_{i+1}$ for $i = 1, \dots, N-1$, therefore, this problem is equivalent to the previous problem and can be solved using the base algorithm with $\mu_i = \frac{n_i B}{2}$. 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 $O(N^3)$.

III. PACKET-ERASURE CHANNEL

In this section, we consider a source–destination pair connected through a single-link, packet-erasure channel, for which an instantaneous error-free feedback is available, as shown in Fig. 1.

A. 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 $M_1 \leq M_2 \leq \dots \leq M_N$. 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 $B = 1$ and thus avoid unnecessarily carrying the variable B around in the mathematical expressions.

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

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

TABLE I
GENERAL NOTATION

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
$d(\cdot)$	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

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

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

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

where $d(\cdot)$ 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*. Note that here the variables γ_i are dropped for simplicity. In other words, it is assumed that each source symbol can be encoded into as many number of bits as needed. The extension to the case where there is an upper bound on the individual packet lengths is straightforward and can be done in a similar manner as in the Basic Problem.

Table I lists the notation used in this section.

B. 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.,

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

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

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

The problem statement therefore simplifies to the following:

$$\boxed{\mathbb{P}_{OL}} \quad \min_{\mathbf{z}} G(\mathbf{z}) = \sum_{i=1}^N g(z_i)$$

subject to

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

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

We refer to the above problem as \mathbb{P}_{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}_{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

$$g(z+2) - g(z+1) > g(z+1) - g(z), \quad \forall z \in \{0, 1, \dots\}. \quad (3)$$

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 parameter pairs (z_1, p) and (z_2, p) is a binomial($z_1 + z_2, p$) random variable, we can write

$$\begin{aligned} & E[d(b_{z+2})] - 2E[d(b_{z+1})] + E[d(b_z)] \\ &= E[d(b_z + b_1 + b'_1)] \\ & \quad - 2E[d(b_z + b_1)] + E[d(b_z)] \\ &= E[d(b_z + b_1 + b'_1) - 2d(b_z + b_1) + d(b_z)] \\ &= \sum_{i=0}^z \sum_{j=0}^1 \sum_{k=0}^1 P_{ijk} [d(i+j+k) - 2d(i+j) + d(i)] \end{aligned}$$

$$= \sum_{i=0}^z (P_{i00} \times 0 + (P_{i01} - P_{i10})[d(i+1) - d(i)] + P_{i11}[d(i+2) - 2d(i+1) + d(i)]) > 0$$

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 $P_{i01} = P_{i10}$. \square

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}_{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), then

$$\sum_{i=1}^{\hat{j}} z_i^* = M_{\hat{j}} \quad (4)$$

where \hat{j} is the index of the smallest element of the set $\{\lfloor \frac{M_j}{j} \rfloor\}_{j=1}^N$. When there are more than one such smallest elements, \hat{j} is taken to be the largest such index. In other words

$$\hat{j} = \max \left\{ \arg \min_j \left\{ \left\lfloor \frac{M_j}{j} \right\rfloor \right\}_{j=1}^N \right\}.$$

Proof: If (4) does not hold, in order to meet the constraints of \mathbb{P}_{OL} , we must have $\sum_{i=1}^{\hat{j}} z_i^* < M_{\hat{j}}$. Then there exists some $j \leq \hat{j}$ such that $z_j^* \leq \lfloor \frac{M_j}{j} \rfloor$. Let r be the largest such j , i.e.

$$r = \max \left\{ j \leq \hat{j} \mid z_j^* \leq \left\lfloor \frac{M_j}{j} \right\rfloor \right\}.$$

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

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

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

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

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

$$\sum_{i=1}^{\tilde{j}} z_i^* + 1 \geq M_{\tilde{j}} + 1$$

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

$$\sum_{i=1}^{\tilde{j}} z_i^* = M_{\tilde{j}}.$$

On the other hand

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

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

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

Then there must exist some $\tilde{i} \in \{1, \dots, \tilde{j}\}$ such that

$$\begin{aligned} z'_{\tilde{i}} &\geq \left\lfloor \frac{M_{\tilde{j}}}{\tilde{j}} \right\rfloor + 1 \\ &\geq \left\lfloor \frac{M_{\hat{j}}}{\hat{j}} \right\rfloor + 1 + 1 \\ &\geq z_r^* + 2. \end{aligned}$$

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

$$G(\mathbf{z}') - G(\mathbf{z}^*) = g(z_r^* - 1) + g(z_r^* + 1) - g(z_r^*) - g(z_r^*) < 0$$

hence a contradiction. \square

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

Open-Loop Algorithm

- 1) Let $\hat{j} = \max\{\arg \min_j \{\lfloor \frac{M_j}{j} \rfloor\}_{j=1}^N\}$, $k = M_{\hat{j}} - \hat{j} \lfloor \frac{M_{\hat{j}}}{\hat{j}} \rfloor$.
- 2) Set $z_j^* = \begin{cases} \lfloor \frac{M_j}{j} \rfloor, & j = 1, \dots, \hat{j} - k \\ \lfloor \frac{M_j}{j} \rfloor + 1, & j = \hat{j} - k + 1, \dots, \hat{j} \end{cases}$.
- 3) If $\hat{j} < N$, remove $\{M_j\}_{j=1}^{\hat{j}}$, set $j = j - \hat{j}$ for $j > \hat{j}$, 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}_{OL} . We prove its feasibility in Lemma 8. Then, in Lemma 9, we show that for a strictly convex $d(\cdot)$, the first \hat{j} elements of \mathbf{z}^* minimize $\sum_{j=1}^{\hat{j}} g(z_j^*)$ among all integer-valued \hat{j} -vectors \mathbf{z} which meet (4). Since according to Lemma 7, (4) is a necessary condition for any vector that solves \mathbb{P}_{OL} , this suffices to show the optimality of the first \hat{j} 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. \square

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 nonnegative. To show that they meet the deadline constraints, two possible cases need to be considered

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

$$\begin{aligned} \sum_{i=1}^j z_i^* &= \sum_{i=1}^j \left\lfloor \frac{M_j}{\hat{j}} \right\rfloor = j \left\lfloor \frac{M_j}{\hat{j}} \right\rfloor \\ &\leq j \frac{M_j}{\hat{j}} \leq j \frac{M_j}{j} = M_j. \end{aligned}$$

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

$$\begin{aligned} \sum_{i=1}^j z_i^* &= \sum_{i=1}^j \left\lfloor \frac{M_j}{\hat{j}} \right\rfloor \\ &\quad + \sum_{i=\hat{j}-k+1}^j 1 = j \left\lfloor \frac{M_j}{\hat{j}} \right\rfloor (j - \hat{j} + k) \\ &= j \left(\frac{M_j}{\hat{j}} - \frac{k}{\hat{j}} \right) + (k - \hat{j} + j) \\ &\leq M_j + j \left(\frac{k - \hat{j} + j}{\hat{j}} - \frac{k}{\hat{j}} \right) \\ &= M_j + \frac{(\hat{j} - j)(k - \hat{j})}{\hat{j}} \leq M_j. \end{aligned}$$

The last inequality is due to the fact that $k < \hat{j}$ and $\hat{j} \geq j$. \square

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{j} -vector \mathbf{z}^* found in the first round of the open-loop algorithm minimizes $G_{\hat{j}}(\mathbf{z}) = \sum_{j=1}^{\hat{j}} g(z_j)$ among all \hat{j} -vectors \mathbf{z} for which (4) holds.

Proof: Let \mathcal{Z} be the set of all (nonnegative) integer N -vectors for which (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,

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

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

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

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

$$G_{\hat{j}}(\mathbf{z}') - G_{\hat{j}}(\mathbf{z}'') = g(z'_i) - g(z'_i - 1) + g(z'_j) - g(z'_j + 1) > 0.$$

Therefore, $G_{\hat{j}}(\mathbf{z}'') < G_{\hat{j}}(\mathbf{z}')$ and no vector in $\mathcal{Z} - \mathcal{Z}_1$ can be optimum. Since we are minimizing $G_{\hat{j}}(\mathbf{z})$ 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 $G_{\hat{j}}(\mathbf{z}) = G_{\hat{j}}(\mathbf{z}^*)$ for all $\mathbf{z} \in \mathcal{Z}_1$, $G(\mathbf{z}^*)$ is the minimum and \mathbf{z}^* is a minimizer. \square

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 IV.

C. 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 [9].

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 $i = 1, \dots, N$, 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 $\mathbf{y}(t) = [y_1(t) \dots y_N(t)]$. So the algorithm will consist of two parts. In the first part, at every time t , we solve the following minimization problem:

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

subject to

$$\begin{aligned} y_i(t) &\geq 0, \quad i = l(t), \dots, N \\ \sum_{i=l(t)}^j y_i(t) &\leq m_j(t), \quad j = l(t), \dots, N \end{aligned}$$

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 an arbitrary value. We assume that the scheduling policy $\psi(\cdot)$ is such that at any given time t we have

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

Note that this assumption can be made without loss of optimality since for any two consecutive unexpired symbols, the first symbol expires no later than the second one. Therefore, the distortion cannot be reduced by sending more bits of the second symbol when there has been fewer prior successful 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 $s(t) = \psi(\mathbf{y}(t))$.

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 $x_i = b_i + p y_i$. We can rewrite the problem in terms of x_i as follows:

$$\boxed{\mathbb{P}_{\text{CEC}}} \quad \min_{x_1, \dots, x_N} \sum_{i=l}^N d(x_i) \quad (5)$$

subject to

$$\begin{aligned} x_j &\geq b_j, \quad j = l, \dots, N \\ \sum_{i=l}^j x_i &\leq C_j, \quad j = l, \dots, N \end{aligned}$$

where $b_l \geq b_{l+1} \geq \dots \geq b_N \geq 0$, and $C_j = \sum_{i=l}^j b_i + p m_j$, for $j \in \{l \dots N\}$. We refer to this problem as \mathbb{P}_{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 $i \in \{1 \dots N\}$. Applying the base algorithm to the above problem will result in nonnegative 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}_{CEC} we only have the equality constraint of $\sum_{i=1}^n x_i = C$. We next show that if $n = \hat{j} = \max\{\arg\min_j \{\frac{C_j}{j}\}_{j=1}^N\}$, the \hat{j} -vector \mathbf{x}^* above, will also meet the deadline constraints of \mathbb{P}_{CEC} . We then proceed by showing that a necessary condition for a vector \mathbf{x}^* to solve \mathbb{P}_{CEC} is to have $\sum_{i=1}^{\hat{j}} x_i = C_{\hat{j}}$. We finally use these results to find the unique optimum solution to \mathbb{P}_{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:

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

subject to

$$\sum_{i=1}^n x_i = C \quad (6)$$

$$x_i \geq b_i, \quad i = 1 \dots n \quad (7)$$

where $C \geq \sum_{i=1}^n b_i$ is a constant, $b_1 \geq b_2 \geq \dots \geq b_n$, 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 $\hat{k}_n^C = \min\{k | b_{k+1} < \mu_n^C(k)\}$.

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 (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 (7) trivially for $k \leq \hat{k}_n^C$, and for $k > \hat{k}_n^C$ since by definition of \hat{k}_n^C we have $x_k = \mu_n^C(\hat{k}_n^C) > b_{\hat{k}_n^C+1} \geq b_k$.

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 $\hat{i} < n$ for which $x'_i > x_i^* = \max\{b_i, \mu_n^C(\hat{k}_n^C)\}$. On the other hand, since \mathbf{x}' must meet (6) and (7), there must exist another element $\hat{i} > \hat{k}_n^C$ such that $x'_i < x_i^* = \mu_n^C(\hat{k}_n^C)$. Define a new vector \mathbf{x}'' as follows:

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

where $0 < \delta < x'_i - \max\{b_i, \mu_n^C(\hat{k}_n^C)\}$. 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. \square

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

Proof: We want to show

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

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

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

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

$$\begin{aligned} \sum_{i=1}^j x_i^* &= \sum_{i=1}^{\hat{k}_j^C} b_i + (j - \hat{k}_j^C) \mu_j^C(\hat{k}_j^C) \\ &= \sum_{i=1}^{\hat{k}_j^C} b_i + (j - \hat{k}_j^C) \frac{C_j - \sum_{i=1}^{\hat{k}_j^C} b_i}{\hat{j} - \hat{k}_j^C} \\ &\leq \sum_{i=1}^{\hat{k}_j^C} b_i + (j - \hat{k}_j^C) \frac{C_j - \sum_{i=1}^{\hat{k}_j^C} b_i}{\hat{j} - \hat{k}_j^C} \\ &\leq \sum_{i=1}^{\hat{k}_j^C} b_i + (j - \hat{k}_j^C) \frac{C_j - \sum_{i=1}^{\hat{k}_j^C} b_i}{\hat{j} - \hat{k}_j^C} = C_j \end{aligned}$$

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

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

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

where $\hat{j} = \max\{\arg\min_j \{\frac{C_j}{j}\}_{j=1}^N\}$, as in Lemma 11.

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

$$x_k^* = \begin{cases} b_k, & k \leq \hat{k}_j^C \\ \mu_j^C(\hat{k}_j^C), & \hat{k}_j^C < k \leq \hat{j}. \end{cases}$$

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

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

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 $\tilde{j} > \hat{j}$ be the smallest index for which $\sum_{i=1}^{\tilde{j}} x'_i > C_j$. Then there must be some \tilde{i} such that $\hat{j} < \tilde{i} \leq \tilde{j}$ and $x'_{\tilde{i}} > x_{\tilde{j}}^* + \delta$; otherwise we have $x'_i \leq x_{\tilde{j}}^* + \delta < \frac{C_j}{\hat{j}}$, for all i such that $\hat{j} < i \leq \tilde{j}$. Therefore

$$\begin{aligned} \sum_{i=1}^{\tilde{j}} x'_i &= \sum_{i=1}^{\hat{j}} x'_i + \sum_{i=\hat{j}+1}^{\tilde{j}} x'_i < C_j + (\hat{j} - \tilde{j}) \frac{C_j}{\hat{j}} \\ &= \hat{j} \frac{C_j}{\hat{j}} + (\hat{j} - \tilde{j}) \frac{C_j}{\hat{j}} = \tilde{j} \frac{C_j}{\hat{j}} < \tilde{j} \frac{C_j}{\tilde{j}} = C_j. \end{aligned}$$

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

$$x''_i = \begin{cases} x_{\hat{j}}^* + \delta, & i = \hat{j} \\ x_i^* - \delta, & i = \tilde{i} \\ x_i^*, & \text{otherwise.} \end{cases}$$

Note that $x_{\tilde{i}}^* - \delta > x_{\hat{j}}^* \geq b_{\hat{j}} \geq b_{\tilde{i}}$, and $\sum_{i=1}^{\tilde{j}} x''_i \leq C_j$. 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. \square

The following algorithm finds a solution to \mathbb{P}_{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}_{CEC}

- 1) $\mu_j = \frac{C_j}{j-l+1}, \forall j \in \{l, \dots, N\}$.
- 2) $\hat{j} = \max\{\arg\min_j \{\mu_j\}_{j=l}^N\}$.
- 3)

$$\mu_{\hat{j}}(k) = \begin{cases} \frac{C_j}{\hat{j}}, & k = l - 1 \\ \frac{C_j - \sum_{i=l}^k b_i}{\hat{j} - k}, & k \in \{l, \dots, N\} \end{cases}$$

$$\hat{k} = \min\{k \in \{l - 1, \dots, N\} | b_{k+1} < \mu_{\hat{j}}(k)\}.$$

- 4) Let $x_k^* = \begin{cases} b_k, & l \leq k \leq \hat{k} \\ \mu_{\hat{j}}(\hat{k}), & \hat{k} < k \leq \hat{j}. \end{cases}$
- 5) If $\hat{j} < N$, let $\hat{l} = \hat{j} + 1, C_i = C_i - C_j, \forall i \geq l$ 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}^*

$$\sum_{i=l}^{\hat{j}} x_i^* = C_j.$$

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

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

where \hat{k} and $\mu_{\hat{j}}(\hat{k})$ are as given in the CEC algorithm. 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}^* 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}_{CEC} when $d(\cdot) \in \mathcal{D}_{\text{sc}}$. Furthermore, using Lemma 5, we can conclude that \mathbf{x}^* also solves \mathbb{P}_{CEC} for a merely convex $d(\cdot)$, i.e., when $d(\cdot) \in \mathcal{D}_c$. Note that in this case the solution is not necessarily unique. \square

The CEC algorithm, finds the real-valued solution vector $\mathbf{y}^*(t)$. The optimum $s(t)$, the index of the symbol from which one bit must be sent at time t , can be directly calculated from the integer-valued solution of the problem, if available. But unfortunately, 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 rest of this section we propose different heuristics to calculate $s(t) = \psi(\mathbf{y}(t))$. The overall algorithm then is given by

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).

As it was explained earlier, $\psi(\cdot)$ should be such that for every $t, b_i(t) \geq b_{i+1}(t), i \in \{l(t), \dots, N\}$. Furthermore, the values of y_i 's given by the CEC algorithm are used as a guideline for the way $\psi(\cdot)$ picks $s(t)$. For example, if $y_i \approx 0$ for some i , it indicates that symbol i is probably not the most important symbol among those awaiting transmission, and therefore, a good policy should not set $s(t) = i$.

The numerical performance evaluation of the heuristic policies proposed in this section is presented in the next section

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:

$$b_i(t) \geq b_{i+1}(t), i \in \{l(t), \dots, N\} \quad (8)$$

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

$$s(t) = \min \left\{ \arg\max_i \{y_i(t)\}_{i=l(t)}^{j^*} \right\}$$

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

In other words, this policy first finds the first j^* elements whose sum reaches a unit. This means that we can afford sending one bit from symbols $\{l, \dots, j^*\}$ without sacrificing more significant bits of the rest of the symbols with later deadlines. Then, the policy sends a bit of the symbol corresponding to the largest y_i among all symbols $\{l, \dots, j^*\}$.

Lemma 13: If $\psi(\cdot) = \psi_1(\cdot)$ in the Closed-Loop algorithm, inequality (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, (8) holds. Next, if (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

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

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, $j^* \leq \hat{j}$, since

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

and therefore

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

thus, $\arg\max_i \{y_i(t)\}_{i=l(t)}^{j^*} = \arg\min_i \{b_i(t)\}_{i=\hat{k}+1}^{j^*}$, and hence

$$s(t) = \min \left\{ \arg\min_i \{b_i(t)\}_{i=\hat{k}+1}^{j^*} \right\}.$$

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

Policy CEC2: $\psi_2(\cdot)$

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

$$s(t) = \begin{cases} j^* & , j^* \leq \hat{j} \\ \hat{k} + 1 & , j^* > \hat{j} \end{cases}$$

where

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

This policy sends one bit corresponding to the first symbol among $\{y_l, \dots, y_j\}$ for which $y_i \geq 1$. If all elements $\{y_l, \dots, y_j\}$ are smaller than a unit, it sends one bit from the first nonzero element of $\{y_1, \dots, y_j\}$. Such element always exists since $\sum_{i=l}^{\hat{j}} y_i = m_{\hat{j}}(t) > 0$.

Lemma 14: If $\psi(\cdot) = \psi_2(\cdot)$ in the Closed-Loop algorithm, for any $t \leq T$, inequality (8) holds.

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

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

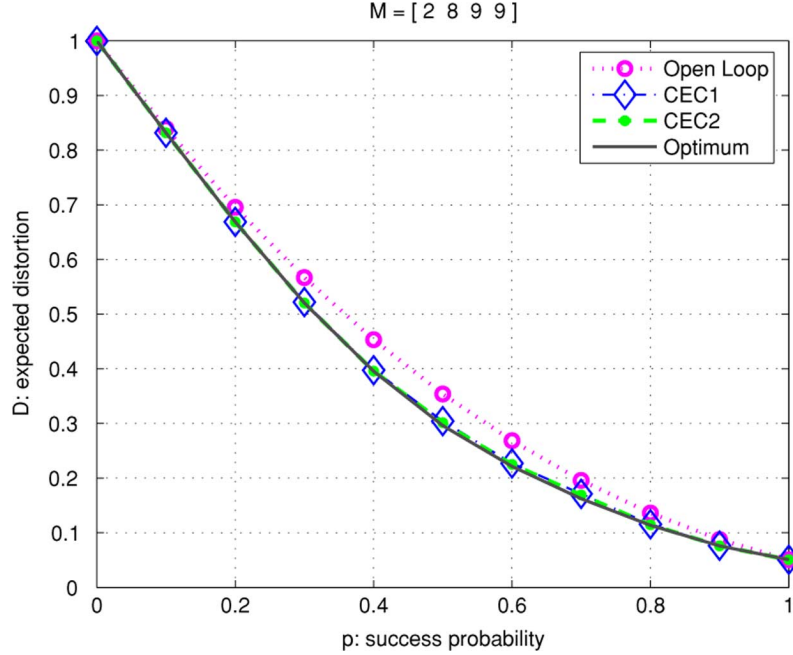


Fig. 4. Comparison of the optimum distortion with the open-loop policy and the CEC heuristic policies for $M = [2 \ 8 \ 9 \ 9]$.

therefore

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

so, $b_{s(t)}(t) < b_{s(t)-1}(t)$. \square

IV. 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 unit-variance 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 closed-loop solution, we use exhaustive search. The exhaustive search is carried out using a recursive algorithm which first generates the Markov Decision Process for the problem and then solves it backward in time by finding the best action for every possible state, given the success probability p . The expected distortion of the optimum policy is then found by taking an average over all possible sample paths.

Fig. 4 shows a comparison between the optimum solution, the open-loop algorithm, and the two 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.

To do a more thorough evaluation of these algorithms, in Fig. 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

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

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 0.003 of the optimum distortion. Another interesting observation, is the behavior of the heuristic policies at very high success probabilities. As we see here, the average performance of the heuristics is slightly worse than that of the open-loop policy. This may be due to the fact that the whole rationale and the methodology of the heuristic policies is based on having packet erasures. If the latter seldom happens then there is little reason to expect that the proposed method will do well. This is in contrast to the open-loop policy which performs best when there is the least

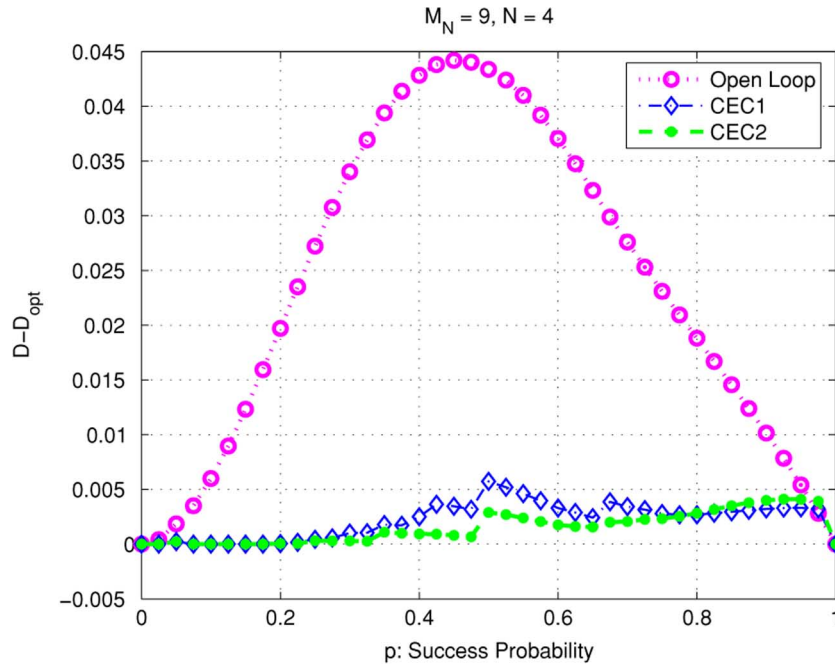


Fig. 5. Performance evaluation of the open-loop policy and the CEC policies for $N = 4$ and $M_N = 9$.

amount of uncertainty, i.e., very low and very high success probabilities.

V. CONCLUSION AND FUTURE WORK

We studied optimum streaming of delay-sensitive data over both error-free and packet-erasure channels. We found the optimum transmission policy for the case of error-free channel, and showed that this policy is independent of the exact 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 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, that not only outperforms the open-loop policy, but also it has near optimal performance.

Another advantage of the CEC algorithm is that it can be used for queuing systems with random arrivals, since it does not require for the information to be stored at the source at the beginning of transmission.

Furthermore, 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.

ACKNOWLEDGMENT

The authors would like to thank André L. Tits for his invaluable comments and feedback.

REFERENCES

- [1] M. G. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for layered streaming media," *J. VLSI Signal Process. Syst. Signal, Image and Video Technol., Special Issue on Multimedia Signal Processing*, vol. 27, no. 1–2, pp. 81–97, Feb. 2001.
- [2] Z. Miao and A. Ortega, "Optimal scheduling for the streaming of scalable media," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Nov. 2000, vol. 2, pp. 1357–1362.
- [3] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. MM-8, no. 2, pp. 390–404, Apr. 2006.
- [4] M. Röder, J. Cardinal, and R. Hamzaoui, "On the complexity of rate-distortion optimal streaming of packetized media," in *Proc. Data Compression Conf.*, Mar. 2004, pp. 192–201.
- [5] D. Saporilla and K. W. Ross, "Optimal streaming of layered video," in *Proc. INFOCOM*, Mar. 2000, vol. 2, pp. 737–746.
- [6] D. Wu, Y. Thomas, W. Zhu, Y. Q. Zhang, and J. M. Peha, "Streaming video over the internet: Approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. CSVT-11, no. 3, pp. 282–300, Mar. 2001.
- [7] A. Faridi and A. Ephremides, "Distortion control for queues with deadlines," in *Proc. Data Compression Conf.*, Mar. 2006, pp. 312–321.
- [8] A. Faridi and A. Ephremides, "Distortion control for packet-erasure channels," in *Proc. IEEE Workshop Stat. Signal Process.*, Aug. 2007, pp. 710–714.
- [9] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scientific, 2005, vol. I.