DISTORTION-BUFFER OPTIMIZED TCP VIDEO STREAMING

Anshul Sehgal — University of Illinois, USA
Olivier Verscheure — IBM T.J. Watson Research, USA
Pascal Frossard — EPFL, Switzerland

ABSTRACT

This paper presents a distortion optimized streaming algorithm for on-demand streaming of multimedia. Given the pre-encoded packets of a multimedia stream, we propose an algorithm for selecting an appropriate subset of these packets such that the overall client distortion is minimized. This minimization is performed within the rate constraints imposed by the communication channel. In the interest of computation it is desirable to limit the horizon (i.e. the look-ahead) over which the optimization is performed. Inevitably, shortening the horizon leads to sub-optimal results. We alleviate the impact due to this through the introduction of a buffering constraint that stipulates a minimum desired buffer occupancy at all time during the streaming session. We pose this problem as a Lagrangian minimization – the solution to which is obtained through an iterative descent algorithm. We demonstrate the efficacy of the proposed approach through empirical evaluation.

1. INTRODUCTION

The current day Internet, while being well-suited to delay-tolerant applications such as file-transfer and e-mail, poses numerous challenges to delay-sensitive applications such as multimedia streaming. Packets injected into the network are liable to be dropped, lost or delayed en-route to their destination. Multimedia delivery systems [2, 3, 4] attempt to address these issues within the stringent delay constraints associated with the packets of a multimedia stream, taking into account the dependencies between the packets of the presentation and the variations in the source coding rate.

However, if multimedia is streamed in conjunction with a reliable transport protocol, such as TCP, packets that are lost, dropped or delayed are re-transmitted within a few round-trip-times (RTT) of the link between the server and the client. Thus, for streaming applications that can tolerate a delay of a few hundred milliseconds the aforementioned issues can be efficiently circumvented. Reliable transport protocols induce another problem. The congestion control/avoidance mechanism used by these protocols to thwart (and encourage) the communication of packets indeed leads to fluctuations in the available streaming rate. Thus, unlike the rate-distortion optimized streaming strategies developed for streaming over loss-prone, delay-agnostic packet networks [2, 3, 4], Internet streaming over TCP requires a mechanism that is able to adapt to the fluctuations in the available bandwidth.

Perhaps the work of [2] comes closest to that proposed herein. We espouse their abstraction of the multimedia encoding process and the iterative descent approach to the minimization of a Lagrangian cost function. The key differences include the problem under consideration and its formulation. While Chou and Miao [2] address the minimization of the expected distortion under an aggregate rate constraint for multimedia streaming over *loss prone* networks, the proposed approach addresses the minimization of the expected distortion under an instantaneous rate-constraint for communication over a *variable bit-rate, reliable* channel. As we shall see, altering the constraints of the problem leads to a significantly different solution strategy.

The minimization of the expected distortion $E(D)$ is achieved through a *transmission policy* π that selects the packets to be transmitted. We wish to minimize $E(D)$ respecting the available channel bandwidth, which, as alluded to earlier fluctuates with time. Owing to the dependencies between the packets of the presentation, the selection of the packets requires a joint optimization over all the packets of the presentation. This, of course, can be computationally prohibitively expensive. On the other hand, shortening the horizon of the optimization, i.e. greedily optimizing over a small number of packets at a time, leads to sub-optimal results. We alleviate this problem through the introduction of a buffering penalty term, i.e. we augment the constraints of the above problem with a buffering constraint that ensures that the amount of data buffered at the client is always greater than a predetermined threshold. We pose this problem as a Lagrangian minimization and use an iterative descent technique to solve the minimization. We demonstrate through simulations that the augmented problem leads to minimal loss in performance.

2. INTERNET MEDIA STREAMING

This section defines our model of the encoding, packetization and streaming processes. We assume that the multimedia stream is encoded, packetized and subsequently, stored at the server, prior to communication.

During the streaming session, the server selects a subset of the pre-encoded packets to communicate to the client, taking into account the available bandwidth on the channel, and the amount of data buffered at the client. A cogent selection of the packets that should be communicated to the client is the topic of this paper.

Denote the total number of packets in a multimedia presentation as L. Depending on the algorithm used for encoding the multimedia presentation, packets have dependencies between them, which we represent by a directed acyclic graph (DAG). We represent the packets of the multimedia stream with nodes in the DAG, and the dependencies between the packets with directed links in the graph. If the decoding of any packet l is contingent on the successful decoding of some other packet l', we call l' an ancestor of l and represent this dependency by a link directed from l' towards l. We also denote this dependence by $l' \preceq l$. In general, successful decoding of a packet l may depend on the successful decoding of multiple packets, in which case, each of these packets is called an ancestor of packet l. Associated with each packet l of a multimedia stream is its size R_l, in bytes, and the decrement in distortion if it is successfully decoded at the client, Δd_l. Denote the distortion incurred if the client does not receive packet l by $d_{0,l}$. Thus, if packet l is successfully decoded, the incurred distortion is $d_{0,l} - \Delta d_l$, else it is $d_{0,l}$. Also associated with packet l is its decoding deadline, $t_{d,l}$, which is the time by which the packet must be available at the client for successful decoding. Often, it is desirable to allow for d seconds of delay prior to commencement of the decoding at the client, i.e. the client buffers the first d seconds of data. Thus, the server commences streaming at time $t = 0$, and the client starts decoding at $t = d$ seconds.

This work has been partly supported by the Swiss National Science Foundation.

To be published in IEEE ICIP, Singapore, October 2004. © IEEE.

3. STREAMING MODEL

In our streaming model, the server is connected to the channel through a buffering interface. We model the buffer as a FIFO queue. Thus, the channel drains the packets from the buffer in the same order in which the server places them in the buffer. We model the communication channel between the server and the client as a variable bandwidth, lossless link. The variable bandwidth nature of the channel implies that the rate at which the channel drains data placed in the server's buffer changes as a function of time. A mechanism to predict the future behavior of the channel, and ascertain the accuracy of the prediction will aid the analysis in the sequel.

Our goal in this paper is to devise streaming strategies for Internet streaming. Thus, prior to addressing the issue of predictability, it would be prudent to define a channel model that captures the characteristics of the Internet well. We assume that the throughput of the Internet can be adequately modeled as an auto-regressive, moving average stationary process with Gaussian innovation. The validity of this model for Internet traffic traces on time scales of a few seconds has been verified in [1, 5]. We model the channel as a discrete-time system, with a sampling interval of T_s seconds. In our model, the channel communicates $x_k T_s$ bytes of data in the time interval $[kT_s, (k+1)T_s]$, where x_k is the available channel bandwidth in the k^{th} time step.

We model the process $\{x_k\}$ as a Gaussian autoregressive process of the form $x_k = \mu + (1-\alpha)\sum_{i=0}^{\infty}\alpha^i n_{k-i},\ k \in \mathcal{Z}, n_j = 0, \forall j < 0$, where each n_k is an independent zero mean Gaussian random variable with variance σ^2, μ denotes the average available bandwidth and α is a modeling parameter.

We conclude our discussion on channel modeling by calculating the pdf of the cumulative amount of data transferred in the interval $[kT_s, (k+r)T_s], r > 0$. In the interval $[kT_s, (k+r)T_s]$, the channel communicates $\sum_{j=k}^{k+r-1} x_j T_s$ bytes of data. Note that $\sum_{j=k}^{k+r-1} x_j T_s$ is a Gaussian random variable, since $\{x_j\}_k^{k+r-1}$ are jointly Gaussian random variables. Conditioned on the knowledge of $\{x_j\}_{j<k}$, it can be shown that [1]

$$\sum_{j=k}^{k+r-1} x_k T_s \sim \mathcal{N}(\mu r T_s + (1-\alpha^{r+1})\sum_{i=1}^{\infty}\alpha^{i-1} n_{k-i}, \beta(r)\sigma^2 T_s^2), \tag{1}$$

where $\beta(r) = \sum_{j=1}^{r}(1-\alpha^j)^2$.

In the sequel, we will be concerned with the time τ it takes the channel to drain an arbitrary number, R_0, of bytes of data placed in the server's buffer at time kT_s. Since the channel throughput is stochastic in nature, the time τ will be a random variable. In particular, we will be interested in the probability $P(\tau \le rT_s)$, where r is a positive integer. Note that the probability that τ is less than rT_s is equal to the probability that the channel transmits R_0 or more bytes of data in rT_s seconds. Equivalently,

$$P(\tau \le rT_s) = \quad P(\textstyle\sum_{j=k}^{k+r-1} x_j T_s \ge R_0). \tag{2}$$

The RHS of Equation 2 can be computed using Equation 1.

Next we turn our attention to the client buffer. We denote the time of arrival of packet l by $t_{a,l}$. When a packet arrives at the client, the client buffers the packet in its input buffer until its decoding deadline, $t_{d,l}$. We note that owing to the stochastic nature of the channel, $t_{a,l}$ will also be a random variable.

In conclusion to this section we note that even though we propose the solution to our problem using the channel model presented above, the proposed approach is applicable to any channel/channel model that affords us the ability to statistically predict the future based on the past observations. In the next Section, we will devise a rate-distortion optimized streaming algorithm for the setup described in this Section.

4. DISTORTION OPTIMIZED STREAMING

This section describes the proposed Distortion-optimized streaming algorithm. At any time t_k [2], the server runs an optimization algorithm and selects a subset of packets to be transmitted from among all the packets in the presentation whose decoding deadlines have not elapsed. We wish to accomplish this selection in a manner such that the expected distortion $E(D)$ of the decoded stream is minimized, or equivalently, the client utility is maximized, taking into account the estimates of the future bandwidth on the channel and the decoder buffer occupancy. The server then runs the optimization algorithm again at a future time t_{k+1}, taking into account updated estimates of the channel parameters.

In order to ease the computational complexity of the algorithm, we assume that the transmission order of the packets is the same as their decoding order. The transmission of any packet l is achieved with a transmission policy π_l. We denote the decision that packet l should be transmitted with $\pi_l = 1$, and the decision that packet l should not be transmitted with $\pi_l = 0$.

In order to simplify the ensuing exposition, we label the packets in the presentation in ascending order of their decoding deadlines. Thus, in our nomenclature, packet l has its decoding deadline prior (or equal) to all subsequent packets $l+1, l+2, ..., L$. If two packets, l and l', have the same decoding deadline, i.e. $t_{d,l} = t_{d,l'}$, then packet l precedes packet l' if $l \preceq l'$, i.e. packet l is an ancestor of packet l'. If neither $l \preceq l'$, nor $l' \preceq l$, we label them arbitrarily.

Consider t_k as the times at which the encoder decides on the packets to be scheduled for transmission. Denote the subset of packets whose decoding deadline has not elapsed, and are consequently eligible for transmission, by $\mathcal{S}_k$. We denote $\pi = \{\pi_{l_k}, \pi_{l_k+1}, ..., \pi_L\}$ as the vector-transmission-policy, where $l_k = \min\{l : l \in \mathcal{S}_k\} = \min\{l : t_{d,l} \ge t_k\}$, and L denotes the total number of packets in the presentation. The l^{th} component of π is denoted as $\pi(l)$. Since we have labeled the packets in non-decreasing order of their decoding deadlines, $t_{d,\pi(l)} \le t_{d,\pi(l')}$ for all $l, l' \in S_k$ such that $l \le l'$. In particular, the L^{th} packet is the last packet to be decoded in the presentation, and consequently, appears as the last component of π.

Note that the expected distortion $E(D)$ is a function of the streaming policy π. At time t_k, we minimize $E(D(\pi))$ over the $L - l_k + 1$ binary variables in π and $jointly$ determine the transmission schedule for all of these packets. The optimization attempts to minimize the overall distortion of the entire presentation at the client, or equivalently maximizes the overall client utility. Next, we determine the expression for the overall distortion.

Denote the time of arrival of packet $l \in S_k$ by $t_{a,l}$. If the encoder decides against transmitting packet l to the decoder, we set $t_{a,l} = \infty$. Owing to the fact that packets are transmitted in their decoding order, if packet l is scheduled for transmission $t_{a,l}$ is the time it will take for the encoder to transmit $\sum_{l'=l_k}^{l} \pi_{l'} R_{l'}$ bytes to the decoder, where R_l is the size in bytes of packet l. Thus, $t_{a,l}$ is a stochastic variable, with

$$P(t_{a,l} \le rT_s) = P(\sum_{k=t_k/T_s}^{rT_s} x_k T_s \ge \sum_{l'=l_k}^{l} \pi_{l'} R_{l'}) \tag{3}$$

where the RHS can be computed using Equation 2. Next, we note that packet l is decodable only if the client receives it before its

[1] Owing to space constraints, we omit the details here.

[2] The quantities $t_{d,l}$, $t_{a,l}$ and t_k are all assumed to be multiples of the sampling interval T_s.

decoding deadline, $t_{d,l}$, i.e. $t_{a,l} \leq t_{d,l}$, and, in addition receives all the ancestors, $l' \preceq l$ of packet l prior to each of their respective decoding deadlines. This can be succinctly written as $I(t_{a,l} \leq t_{d,l}) \prod_{l' \preceq l} I(t_{a,l'} \leq t_{d,l'})$, where $I(\mathcal{B}) = 1$ if the Boolean expression $\mathcal{B}$ is true, and zero otherwise. Noting that the distortion due to packet l is given by $d_{0,l} - \Delta d_l$ if it is decoded at the client, and $d_{0,l}$ otherwise, the distortion $D(\pi)$ under policy π is given by,

$$D(\pi) = \sum_{l \in S_k} \{d_{0,l} - \Delta d_l I(t_{a,l} \leq t_{d,l}) \Pi_{l' \preceq l} I(t_{a,l'} \leq t_{d,l'})\} \tag{4}$$

Distortion $D(\pi)$ is a random variable, since the channel is stochastic. Noting that $E(I(\mathcal{B})) = P(\mathcal{B})$, the expected distortion induced by π is

$$E(D(\pi)) = \sum_{l \in S_k} d_{0,l} - \Delta d_l P(t_{a,l} \leq t_{d,l}) \Pi_{l' \preceq l} P(t_{a,l'} \leq t_{d,l'}) \tag{5}$$

where we have factored the expectation of the product of indicator functions as the product of expectations [2]. Equation 5 can be evaluated for any π by substituting Equation 3 (which itself is obtained from Equation 2) for each term in the product.

4.1. Buffering Penalty

Our goal is to obtain a policy vector π such that Equation 5 is minimized. Note that the search space for minimization of Equation 5 is exponential in the length of π. Thus, instead of running the optimization over all of the $L - l_k + 1$ packets whose decoding deadline has not elapsed as yet, it is desirable to run the optimization over a smaller number of packets, say m. However, shortening the horizon of the optimization will lead to greedy results unless we penalize the cost function. It is natural to penalize Equation 5 with the buffer occupancy at the decoding deadline of the last of the m packets over which the optimization is run. This relaxation allows us to run the optimization over a small number of packets with minimal loss in performance while simultaneously ensuring a large reduction in complexity. We pose the minimization of this augmented cost function as a Lagrangian minimization.

As a notational convenience, we denote $m_k = l_k + m - 1$ in the sequel. We impose the penalty in terms of the expected client buffer occupancy at time t_{d,m_k}, the decoding deadline for packet m_k.

The expected buffer occupancy at time t_{d,m_k} is given by the sum of the buffer occupancy at time t_k, denoted as $B(t_k)$, and the expected number of bytes the encoder would transmit to the client in the interval $[t_k, t_{d,m_k}]$, less the number of bytes consumed by the client in the interval $[t_k, t_{d,m_k}]$. We determine the expected value of $B(t_{d,m_k})$ in a manner akin to the derivation of Equation 5, yielding,

$$\begin{aligned} E(B(t_{d,m_k})) = B(t_k) + & \qquad (6) \\ \frac{(t_k - t_{d,m_k})}{T_s} \mu r T_s + (1 - \alpha^{r+1}) \sum_{i=1}^{\infty} \alpha^{i-1} n_{k-i} & \\ - \sum_{j \in \mathcal{S}_k} \pi_j R_j P(t_{a,l} \leq t_{d,m_k}) I(t_{d,l} \in (t_k, t_{d,m_k}]), & \end{aligned}$$

where each of the three terms noted above appear in the three lines of the equation. We note from Equations 6 and 5 that the policy vector π affects the expected buffer occupancy and the expected distortion. The task at hand is to select π such that the expected distortion is minimized under an expected buffer occupancy constraint.

4.2. Distortion-Buffer Optimization

We frame the minimization problem as

$$\min_{\pi} E(D(\pi)) - \lambda E(B(t_{d,m_k})), \tag{7}$$

where λ is the Lagrange multiplier. We also note that the first two terms in the RHS of Equation 6 do not depend on the policy π, thus can be ignored while computing Equation 7.

For a fixed λ, we minimize Equation 7 using an iterative coordinate descent algorithm, akin to the Iterative Sensitivity Adjustment (ISA) algorithm of [2]. In any one iteration, we minimize Equation 7 over the decision variable π_l by evaluating the expression for $\pi_l = 0$, and $\pi_l = 1$, keeping all $\pi_j, j \in S_k, j \neq l$ fixed, and choosing the value of π_l that minimizes Equation 7. In the subsequent iteration, we keep all $\pi_j, j \in S_k, j \neq (l+1)$ fixed and minimize Equation 7 over π_{l+1}, and so on and so forth. Thus, we iteratively minimize Equation 7 over the decision variables $\{\pi_l\}$ in a round-robin manner. Convergence is guaranteed since at each step of the iteration the cost function decreases, and is lower bounded [3].

The question that remains is the choice of the Lagrange multiplier λ. We wish to choose λ such that the expected buffer occupancy is always greater than a level B_0 bits. Any iterative method, such as bisection search or gradient-descent, can be used to determine the value of λ that achieves this objective. However, this would require recursive minimization of Equation 7 until a value of λ that ensures that the expected buffer occupancy is B_0 is chosen. This, of course would require excessive computation and would defeat the purpose of introducing the penalty term. Instead, we adaptively track the value of λ over time. Thus, depending on the buffer occupancy at the current time t_k, we alter the value of λ using the following equation,

$$\lambda_{k+1} = \lambda_k + \gamma(B_0 - B(t_k)) \tag{8}$$

where γ is a small constant value and $B(t_k)$ is the buffer occupancy at the current time. Lagrange multiplier λ_{k+1} is used in Equation 7 when the optimization is run at time t_{k+1}. Note that Equation 8 increases the value of λ if the current buffer occupancy is less than B_0, and vice-versa. Equation 8, has numerous interpretations. Communications Networks analysis interprets Equation 8 as the AQM controller and Equation 7 as the user-rate control equation. In the Economics literature, the coupled pair of Equations 7 and 8 have an interesting interpretation. Equation 7 represents the user's demand curve, where the supply curve is the deviation in the buffer occupancy from the desired value of B_0. Equation 8 represents the shadow pricing function, and λ_k represents the shadow price. When the supply is scarce, Equation 8 increases the price, and the user communicates lesser data, and vice-versa.

In the next section, we evaluate the performance of the proposed approach for on-demand streaming of video.

5. EXPERIMENTAL RESULTS

In this section, we report our experimental results for on-demand streaming of video. Simulations were run on 160 seconds of the $Jurrasic\ Park$ sequence at 30 fps [7] encoded at 256 Kbps using the MPEG-4 algorithm in the $IBBPBBP...$ format with a GOP size of 13 video frames. The mean available bandwidth μ was varied from 172 Kbps to 258 Kbps to obtain the rate-distortion characteristics of the proposed approach. The parameter T_s was set to $20ms$, α was set to 0.98, and the desired buffer level B_0 was set to 0.25μ (i.e. the desired buffer occupancy was set to 250ms seconds of video). Next, note that $p(x_k)$ is a Gaussian with standard deviation $\sqrt{(1-\alpha)/(1+\alpha)}\sigma$. The parameter σ was set such that the standard deviation of x_k was a fraction ρ of the mean available bandwidth μ. Thus, σ was set to $\sqrt{(1+\alpha)/(1-\alpha)}\rho\mu$. We refer

[3] A trivial lower bound independent of the policy π is $-B(t_k) - \frac{(t_k - t_{d,m_k})}{T_s} \mu r T_s - (1 - \alpha^{r+1}) \sum_{i=1}^{\infty} \alpha^{i-1} n_{k-i}$

to the quantity $\rho\mu$ as the *channel burstiness*. The results were averaged over multiple realizations of the channel to obtain statistically meaningful results.

As a reference, we compare the performance of the proposed approach with a distortion-agnostic ad-hoc scheme. Denote the ratio of the average size of an I,P and B packet as $g_1 : g_2 : g_3$. The ad-hoc scheme operates as follows. If at any time t during streaming, the client buffer occupancy $B(t)$ is greater than the threshold B_0, all of the I,P and B packets are transmitted. If on the other hand $B(t) < \frac{g_1+g_2}{g_1+g_2+g_3}B_0$, subsequent B-frames are skipped until $B(t) \geq B_0$. Similarly, if $B(t) < \frac{g_1}{g_1+g_2+g_3}B_0$ subsequent P-frames are skipped until $B(t) \geq \frac{g_1}{g_1+g_2+g_3}B_0$.

Figure 1 depicts the PSNR vs. rate plots as the policy-length is varied (with the channel-burstiness set to 10%). Note that in the interest of computation, it is desirable to operate with a short policy length. As can be seen from the figure, there is minimal loss in the performance of the proposed approach as the policy length is shortened from 26 frames (2 GOPs) to 13 frames (1 GOP). . Also plotted in the Figure is the performance of the ad-hoc scheme. As expected, the performance of an optimized transmission policy, taking the rate-distortion characteristics of the video packets and the channel state into account exceeds that of the simple ad-hoc scheme. Further, it was empirically observed that reducing the horizon of the optimization by a factor of two leads to a ten-fold improvement in computation efficiency.

Figure 2 plots the performance of the proposed approach and the ad-hoc approach with the channel-burstiness set to 10% and 25%. As can be seen from the figure, altering the burstiness of the channel has a lesser impact on the performance of the proposed approach as opposed to the simple ad-hoc approach.

Lastly, Figures 3(a) and 3(b) plot the client buffer occupancy and the Lagrange multiplier λ as a function of time for a typical realization of the channel. The solid horizontal line in Figure 3(a) depicts the desired buffer level B_0. As can be observed from the Figure, when the buffer level $B(t)$ drops below B_0, λ is increased to restore $B(t)$ to B_0 and vice-versa. The key point to note is that the coupled pair of equations, Equations 7 and 8, ensure that client buffer underflow never occurs.

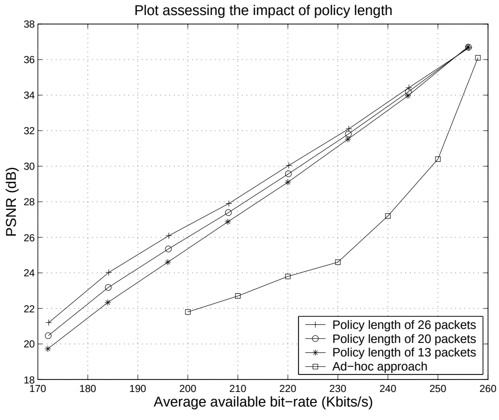

Fig. 1. PSNR vs. rate plot assessing the impact of policy length on the performance of the proposed system (over a channel with 10% burstiness).

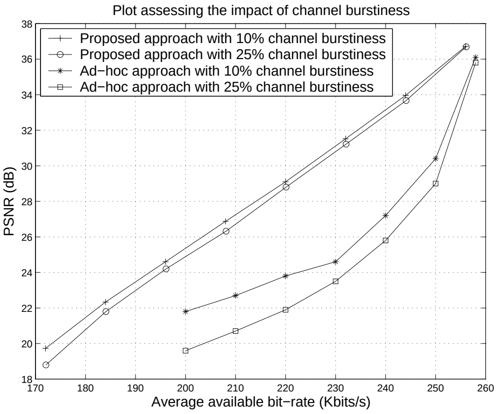

Fig. 2. PSNR vs. rate plot assessing the impact of channel burstiness on the performance of the proposed approach.

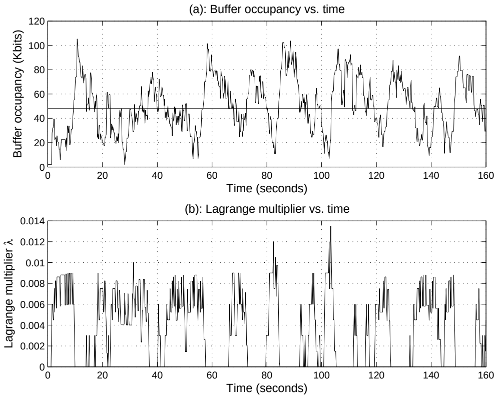

Fig. 3. (a): This figure plots the buffer occupancy at the client over the duration of a typical streaming session. (b): This figure plots the variation in the Lagrange multiplier λ as a function of time, corresponding to Figure 3(a).

6. CONCLUSIONS

We have proposed a distortion-buffer optimization strategy for selecting the transmission policy for multimedia streaming over a reliable but variable bit rate channel. The main contribution lies in its efficient formulation which affords us the ability to limit the horizon of the optimization, and thus the complexity, with acceptable loss in performance. A gain of several dBs in video quality (PSNR) is attained over ad-hoc strategies for streaming rates of interest. A generalization of the method to live streaming over reliable channels is the focus of our future work.

7. REFERENCES

[1] A.Sang, San-qi Li, “A Predictability Analysis of Network Traffic,” *IEEE INFOCOM 2000*, Tel-Aviv, Israel.

[2] P. A. Chou, Z. Miao, “ Rate-Distortion Optimized Streaming of Multimeida,” *IEEE Transactions on Multimeida*, under review.

[3] J. Chakareski, P. A. Chou, B. Girod, “Rate-Distortion Optimized Streaming from the Edge of the Network,”*Proc. IEEE Fifth Workshop on Multimedia Signal Processing*, St. Thomas, Virgin Islands, December 2002.

[4] J. Zhou, J. Li, “Scalable audio coding over the Internet with rate-distortion optimization,” *Proc. International Conference on Image Processing*, Thessaloniki, Greece, October 2001.

[5] Young Young Kim, S. Q. Li ,“Capturing Important Statistics of A Fading/Shadowing Channel for Network Performance Analysis,” IEEE Journal of selected area in communications, May 1999, vol.17, No.5.

[6] F.P. Kelly, P.B. Key and S. Zachar, “Distributed admission control,” *IEEE Journal on Selected Areas in Communications*, vol. 18, pages 2617-2628, 2000.

[7] http://peach.eas.asu.edu/index.html .

To be published in IEEE ICIP, Singapore, October 2004. © IEEE.