FAST DISTORTION-BUFFER OPTIMIZED STREAMING OF MULTIMEDIA

Anshul Sehgal [1]*, Ashish Jagmohan* [1]*, Olivier Verscheure* [1] *and Pascal Frossard* [2]
[1] IBM T. J. Watson Research, USA and [2] EPFL, Switzerland
{sehgal1, ashishja, ov1}@us.ibm.com, frossard@epfl.ch

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 a fast 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 particular, at each transmission opportunity, the proposed approach uses a linear-time algorithm to select the best packet to transmit through the minimization of the expected client distortion. The time complexity of the algorithm is reduced through a factorization of the streaming policy into simpler terms and performing a greedy optimization to select the packet. Inevitably, this in itself leads to sub-optimal results. To alleviate the adverse impact of the greedy optimization, the cost function is penalized with the expected buffer occupancy at the end of the epoch of the optimization. We pose this problem as a Lagrangian minimization. We demonstrate the efficacy of the proposed approach through empirical evaluation.

1. INTRODUCTION

The current Internet has originally been designed for handling delay-tolerant applications, such as file transfer, and packet loss is generally recovered by retransmission. Even if media streaming applications generally induce very strict delay constraints, there is no guarantee that the network packets arrive at destination on time. Multimedia delivery systems (e.g. [3, 4, 5]) attempt to alleviate the impact of these short-comings of the channel within the stringent delay constraints associated with the packets of a multimedia stream: They propose packet transmission policies that take into account the dependencies between the packets of the media stream, and the variations in the source coding rate.

An alternate way to address the network unreliability is to stream the multimedia data over a reliable transport protocol, such as TCP [1], where packets that are lost or delayed are automatically re-transmitted by the transport protocol. The use of such a reliable transport protocol, on the other hand, poses its own challenges on the bandwidth utilization. The congestion avoidance mechanism used by TCP to ensure that the link bandwidth is shared in a fair manner causes fluctuations in the available bandwidth to any one user. This paper presents a fast distortion optimized streaming algorithm that determines the subset of packets of a multimedia stream that should be selected for transmission, taking into account the varying throughput of the channel. While the streaming algorithm of [1] performs well in terms of its rate-distortion performance, its computational complexity is exponential in the horizon of the optimization. We suitably alter it, in order to significantly reduce the complexity of the streaming system. The novel approach proposed in this paper still performs competitively in terms of distortion, but the computational complexity now increases only linearly with the horizon of the optimization.

We espouse the abstraction of the multimedia encoding process proposed in [3], and similarly cast the streaming problem as a Lagrangian optimization. However, the very problem under consideration is quite different. While [3] addresses the minimization of the expected distortion under an aggregate rate constraint for multimedia streaming over *loss prone* networks, the approach proposed here addresses the minimization of the expected distortion, under an *instantaneous* rate-constraint for communication over a *variable bit-rate, reliable* channel. We demonstrate through simulations that the proposed greedy optimization achieves similar performance to the algorithm of [1], while using a fraction of the computational resources. In addition, it is observed that the performance of the proposed approach is far superior to that of a naïve distortion agnostic streaming algorithm.

2. MEDIA STREAMING MODEL

This section describes our model of the media streaming process and introduces the notations used throughout this work.

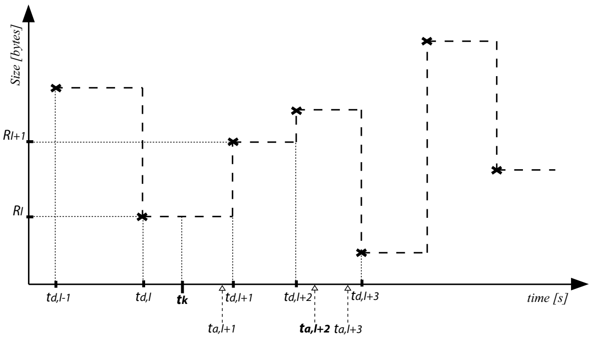

Fig. 1. Notations used throughout this work. The plot above depicts the size of the various packets in the encoded multimedia. The $x-$axis denotes the time line, typical realizations of the various time quantities used in this work are depicted on the $x-$axis.

We tackle the scheduling of pre-encoded media packets over a reliable but variable available rate network. We define the streaming schedule $\mathcal{T} = [t_0, t_1, \ldots, t_K]$ consisting of times t_k at which the server runs an optimization algorithm to determine the subset of encoded packets to be transmitted. We model the dependencies between encoded packets of a media stream by a directed acyclic graph (DAG) where the nodes represent the packets and the directed links denote the packets' dependencies. 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 denote it by $l' \preceq l$. Associated with each packet l of a media 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 have arrived at the client for successful decoding. We denote the time of arrival of packet l by $t_{a,l}$. Figure 1 illustrates the notations. Note that in Figure 1 at time t_k, packet l cannot be scheduled for transmission because its decoding deadline $t_{d,l}$ has passed (i.e. $t_{d,l} < t_k$). Also, packets $l+1$, $l+2$ and $l+3$ are selectable for transmission but packet $l+2$ is expected to not arrive before its decoding deadline (i.e. $t_{d,l+2} < t_{a,l+2}$).

3. DISTORTION OPTIMIZED STREAMING

In this section we briefly revisit the distortion optimized streaming algorithm proposed in [1], and cite the short-comings of the approach proposed therein.

The transmission of any packet l is achieved with a transmission policy π_l – the decision that packet l should not be transmitted is denoted as $\pi_l = 0$, and $\pi_l = 1$ denotes the decision that packet l should be transmitted. At scheduling opportunity t_k, $t_k \in \mathcal{T}$, the encoder selects the packets to be scheduled for transmission from a set $\mathcal{S}_k = \{l : t_{d,l} > t_k\}$ of packets whose decoding deadlines have not yet elapsed. The streaming policy vector $\pi = [\pi_{l_k}\ \pi_{l_k+1}\ \ldots\ \pi_L]$, of length $L - l_k + 1$, is a concatenation of the decision variables π_l for each packet $l \in \mathcal{S}_k$ in ascending order, where $l_k = \min\{l : l \in \mathcal{S}_k\}$ (the packet with the earliest decoding deadline greater than t_k) and $L = \max\{l : l \in \mathcal{S}_k\}$ (the last packet of the presentation).

Each of the 2^{L-l_k+1} values of π induces an expected distortion $E(D(\pi))$. In its simplest incarnation, the algorithm of [1] selects the policy vector π that minimizes $E(D(\pi))$, which can be written as:

$$E(D(\pi)) = \sum_{l \in S_k} \{d_{0,l} - \pi_l \Delta d_l P(l \text{ is decodable})\} \qquad (1)$$

Owing to the prohibitive computational complexity associated with the minimization of Equation (1) over the 2^{L-l_k+1} possible values of π, the approach of [1] limits the horizon of the optimization to m packets, i.e. $\pi = [\pi_{l_k}\ \ldots \pi_{m_k}]$, where $m_k = l_k + m - 1$ and the computational complexity reduces to 2^m. Further, in this case $\mathcal{S}_k = \{l : l_k \le l \le m_k\}$. Limiting the horizon, however, leads to a greedy optimization. To alleviate the impact of this, the cost function in Equation (1) is augmented with a buffering penalty term which stipulates that the expected decoder buffer occupancy $E(B(t_{d,m_k}))$[1] at time t_{d,m_k}, the decoding deadline of packet m_k, is greater than or equal to a predefined threshold B_0. This is achieved through a Lagrangian optimization, with the minimization of the augmented cost function expressed as

$$\min_{\pi} J(\pi) = \min_{\pi} E(D(\pi)) - \lambda_k E(B(t_{d,m_k})). \qquad (2)$$

The Lagrange multiplier λ_k is altered through the course of the streaming session as: $\lambda_k = \lambda_{k-1} + \gamma(B_0 - B(t_k))$, where γ is a small number.

The expected buffer occupancy at time t_{d,m_k} is expressed as follows. Note that at time t_{d,m_k} all the packets that are currently residing in the decoder buffer, and those that will be transmitted in the current optimization epoch would have been either consumed, or discarded if they arrived after their respective decoding deadlines, because $\forall l \le m_k, t_{d,l} \le t_{d,m_k}$. Thus if we denote by the random variable $X_{[t_k, t_{d,m_k}]}$ the cumulative channel throughput in the interval $[t_k, t_{d,m_k}]$, the expected decoder buffer occupancy at time t_{d,m_k} is the excess data over $\sum_{l \in \mathcal{S}_k} \pi_l R_l$ (the amount of data that will be communicated to the decoder under the current policy π and also consumed before t_{d,m_k}). Denote the event $\mathcal{L} = \text{true}$ if $X_{[t_k, t_{d,m_k}]} > \sum_{l \in \mathcal{S}_k} \pi_l R_l$ and $\mathcal{L} = \text{false}$ otherwise. Then $E(B(t_{d,m_k})) = E(X_{[t_k, t_{d,m_k}]} I(\mathcal{L}))$, where $I(\mathcal{L}) = 1$ if the Boolean expression $\mathcal{L}$ is true, and zero otherwise.

Finally, the computation of various quantities such as the probability that a packet will arrive before its decoding deadline or the number of bytes the network will deliver in a given time interval requires a mathematical model of the available channel throughput. We assume that this throughput is adequately modeled as an auto-regressive, moving average stationary process with Gaussian innovation. The validity of this model for real Internet traffic traces on time scales of a few seconds has been verified in [2, 6]. The channel is modeled 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. The cumulative throughput of the channel $X_{[kT_s,(k+r)T_s]}$ in the interval $[kT_s, (k+r)T_s], r > 0$ is written as:

$$X_{[kT_s,(k+r)T_s]} \sim \mathcal{N}((\mu + (1-\alpha^{r+1}) \sum_{i=1}^{\infty} \alpha^{i-1} n_{k-i}) r T_s, \beta(r) \sigma^2 T_s^2), \qquad (3)$$

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

4. SINGLE PACKET TRANSMISSION POLICY

We note that even for modest values of m (the horizon of the optimization) the search space for the solution (2^m policies) is quite large. In this section we make greedy approximations to the cost function in Equation (2) to reduce the search space for the transmission policy π. In particular, we are interested in determining a *single packet transmission policy* for transmission at time t_k. Once the packet to be transmitted is selected and communicated to the decoder, the optimization is run again at time t_{k+1} to select a packet for transmission, taking into account the updated estimates of the channel and the decoder buffer occupancy. Thus, the transmission schedule $\mathcal{T}_k$ in this case is itself a function of the channel rate.

A naïve solution to the single packet transmission policy problem is to restrict the $m \times 1$ policy vector π to the form $\pi(l) = 1$ for some $l \in \mathcal{S}_k$ and $\pi(l) = 0\ \forall\ l \in \mathcal{S}_k, l \neq l$. The number of policies over which the search need be performed in this case is m. This search can be accomplished directly by computing Equation (2) for each of the m policies, and selecting the minimizing argument. However, such an $ad-hoc$ approximation will inevitably lead to poor results, owing to the fact that an arbitrary subset of m policies of the 2^m policies is being sampled. In the sequel, we devise a single packet transmission policy based on an approximate factorization of the cost function in Equation (2) through a partitioning of the policy vector π. As we shall see, the solution to the factorization yields a policy π of length $l_\rho - l_k + 1$ which is of the form $\pi_l = 0\ \forall l_k \le l < l_\rho$, and $\pi_{l_\rho} = 1$, which differs from the naïve policy in only its length ($l_\rho - l_k + 1$ as opposed to m).

The approximate factorization of Equation (2) is achieved in

[1] The expressions for $E(B(t_{d,m_k}))$ proposed herein are slightly different from those presented in [1].

three steps: (1) First, without any loss of generality, we rewrite the cost $J(\pi)$ as the sum of two simpler cost functions $J_1(\pi^1)$ and $J_2(\pi^2)$, where $\pi = [\pi^1 \mid \pi^2]$ is the concatenation of the vectors π^1 and π^2, and π^1 has a specific form (detailed in the ensuing discussion), (2) Next, we make the greedy approximation that the minimization of $J(\pi)$ over π is approximately equal to the minimization of $J_1(\pi^1)$ over π^1 and $J_2(\pi^2)$ over π^2, and (3) We argue that since we are interested in a single packet transmission policy, the optimization over $J_2(\pi^2)$ is immaterial to the problem at hand. It is crucial to note that the only approximation we make is in step (2) above. Next, we provide the details of each of the steps mentioned above.

Step 1:

Consider the partitioning of the policy vector π as $\pi = [\pi_{l_k:l_\rho}|\pi_{l_\rho+1:m_k}]$, for some $l_\rho, l_k \leq l_\rho \leq m_k$. Policies $\pi_{l_k:l_\rho} (= \pi^1$ in the notation above) and $\pi_{l_\rho+1:m_k}$ $(= \pi^2$ in the notation above) are abbreviations for the policies $[\pi_{l_k} \ldots \pi_{l_\rho}]$ and $[\pi_{l_\rho+1} \ldots \pi_{m_k}]$, respectively. We restrict $\pi_{l_k:l_\rho}$ to a single packet transmission policy of the form $\pi(l) = 0 \, \forall \, l_k \leq l < l_\rho$ and $\pi(l_\rho) = 1$, i.e. $\pi_{l_k:l_\rho}$ is of the form $[0 \; 0 \; \ldots \; 1]$. We make no restrictions on the form of $\pi_{l_\rho+1:m_k}$. Thus, for any arbitrary policy π, there exists a l_ρ such that π can be represented as $\pi = [\pi_{l_k:l_\rho} \mid \pi_{l_\rho+1:m_k}]$. Thus, without any loss of generality, we can re-write Equation (2) as an optimization over $\pi_{l_k:l_\rho}$, $\pi_{l_\rho+1:m_k}$ and l_ρ. Note that given l_ρ, $\pi_{l_k:l_\rho}$ is fixed as defined above. Thus, Equation (2) can be written as

$$\min_{\pi} J(\pi) = \min_{l_\rho, \pi_{l_\rho+1:m_k}} \left\{ J_1(\pi_{l_k:l_\rho}) + J_2(\pi_{l_\rho+1:m_k}|\pi_{l_k:l_\rho}) \right\}, \tag{4}$$

where J_1 and J_2 are defined as follows. Note from Equation (1) that the expected distortion $E(D(\pi))$ is additive over the packets l, $\{l : l_k \leq l \leq m_k\}$. Thus, $E(D(\pi)) = E(D(\pi_{l_k:l_\rho})) + E(D(\pi_{l_\rho+1:m_k}|\pi_{l_k:l_\rho}))$. Further, the buffer penalty in Equation (2) can be written as $B(t_{d,m_k}) = B(t_{d,l_\rho}) + \Delta B(t_{d,m_k}|l_\rho)$. Thus we partition $J(\pi)$ as J_1 and J_2, where $J_1(\pi_{l_k:l_\rho}) = E(D(\pi_{l_k:l_\rho})) + B(t_{d,l_\rho})$ and $J_2(\pi_{l_\rho+1:m_k}|\pi_{l_k:l_\rho}) = E(D(\pi_{l_\rho+1:m_k}|\pi_{l_k:l_\rho})) + \Delta B(t_{d,m_k}|\rho)$.

Step 2:

To devise the greedy single packet optimization policy, we make the greedy approximation that the joint optimization over l_ρ and $\pi_{l_\rho+1:m_k}$ in Equation (4) is close to the optimization of J_1 over l_ρ and the optimization of J_2 over $\pi_{l_\rho+1:m_k}$, i.e.,

$$\min_{\pi} J(\pi) \approx \min_{l_\rho} J_1(\pi_{l_k:l_\rho}) + \min_{\pi_{l_\rho+1:m_k}} J_2(\pi_{l_\rho+1:m_k}|\pi_{l_k:l_\rho}), \tag{5}$$

where the optimization of the first term is over l_ρ of the single packet transmission policy $\pi_{l_k:l_\rho}$, and the optimization of the second term is over $\pi_{l_\rho+1:m_k}$ assuming knowledge of $\pi_{l_k:l_\rho}$. Equation (5) is the approximate factorization of Equation (4), and is the key step in the derivation of the single packet transmission policy.

Step 3:

Further note that once the first term of Equation (5) has been minimized over l_ρ to select the single packet transmission policy $\pi_{l_k:l_\rho}$, the optimization of J_2 need not be performed. Instead of performing the minimization of J_2 to determine $\pi_{l_\rho+1:m_k}$, running the optimization at time t_{k+1} after transmitting the selected packet from $\pi_{l_k:l_\rho}$ would have the added advantage of having a more reliable estimate of the available channel bandwidth.

Thus, substituting the expressions for the expected distortion and the expected buffer occupancy from Section 3, and simplifying the ensuing expressions, we get the single packet transmission policy as the minimization

$$\min_{l_\rho} J_1(\pi_{l_k:l_\rho}) = \min_{l_\rho} \sum_{l \in \mathcal{S}_{k,l_\rho}} d_{0,l} - \pi_{l_\rho} \Delta d_l$$
$$\times P(t_{a,l_\rho} \leq t_{d,l}) \Pi_{l' \preceq l_\rho} \pi_{l'} I(t_{a,l'} \leq t_{d,l'}) - \lambda E(X_{[t_k,t_{d,l_\rho}]} I(\mathcal{L})). \tag{6}$$

where $\mathcal{S}_{k,l_\rho} = \{l : l_k \leq l \leq l_\rho\}$. Since $l_k \leq l_\rho < m_k$, it can only take one of m values. The optimization is thus performed directly by searching for the l_ρ which minimizes Equation (6).

This concludes our discussion on the single packet transmission policy. We note that the proposed simplifications, while making a greedy approximation, reduce the search space of the optimization of Equation (2) from 2^m to m.

4.1. Probability Approximations

In this section we derive the various stochastic quantities inherent to our scheduling algorithm.

Consider the arrival time t_{a,l_ρ} of packet l_ρ scheduled for transmission at time t_k in Equation (6). Owing to the stochastic nature of the channel, the arrival time t_{a,l_ρ} will be a random variable. We are interested in $P(t_{a,l_\rho} \leq t_{d,l_\rho})$, the probability that packet l_ρ arrives at the decoder prior to its decoding deadline t_{d,l_ρ}. The probability $P(t_{a,l_\rho} \leq t_{d,l_\rho})$ is equivalent to the probability that $X_{[t_k,t_{d,l_\rho}]}$, the throughput of the channel in the interval $[t_k, t_{d,l_\rho}]$, is greater than R_{l_ρ}, i.e., $P(t_{a,l_\rho} \leq t_{d,l_\rho}) = P(X_{[t_k,t_{d,l_\rho}]} \geq R_{l_\rho})$. In practice, t_k and t_{d,l_ρ} are seldom integral multiples of T_s as is warranted by Equation (3). We appropriately amend the equations to take the effect of discretization into account.

Also, the expected buffer occupancy $E(X_{[t_k,t_{d,l_\rho}]}I(\mathcal{L}))$, $I(\mathcal{L}) = 1$ if $X_{[t_k,t_{d,l_\rho}]} \geq R_{l_\rho}$ and $I(\mathcal{L}) = 0$ otherwise, is given as – $E(X_{[t_k,t_{d,l_\rho}]}I(\mathcal{L})) = \int_{R_{l_\rho}}^{\infty} x p_X(x) dx$, where $p_X(x)$ is the pdf of $X_{[t_k,t_{d,l_\rho}]}$, given in Equation (3). With these quantities as given, the search for the minimizing l_ρ in Equation (6) is performed by substitution of the m values l_ρ can take.

Lastly, we make simplifying approximations to the above probabilities to aid computation. Succinctly, we approximate the pdf of Equation (3) as the *Dirac-delta* function $p_X(x) = \delta(x - x_0)$, where $x_0 = (\mu + (1 - \alpha^{r+1}) \sum_{i=1}^{\infty} \alpha^{i-1} n_{k-i}))(t_{d,l_\rho} t_k)$ is the mean throughput in the interval $[t_k, t_{d,l_\rho}]$ of the channel conditioned on time t_k (from Equation (3). As a consequence of this approximation,

$$P(t_{a,l} \leq t_{d,l}) = \begin{array}{l} 1 \; if \; R_{l_\rho} \leq x_0 \\ 0 \; if \; R_{l_\rho} > x_0 \end{array}. \tag{7}$$

Further, the expected buffer occupancy (from Equation (6)) is given as

$$E(X_{[t_k,t_{d,m_k}]}I(\mathcal{L})) = \begin{array}{l} x_0 - R_{l_\rho} \; if \; R_{l_\rho} \leq x_0, \\ 0 \; if \; R_{l_\rho} > x_0. \end{array} \tag{8}$$

With these approximations in place, the server only needs to maintain a time averaged estimate of the mean available bandwidth. While performing the optimization, it uses Equations (7) and (8) to make the comparisons with the related quantities, thereby further reducing the computational complexity of the proposed approach.

5. EXPERIMENTAL RESULTS

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 parameter T_s in the simulations was set to $20ms$, α was set to 0.98, and the desired buffer level B_0 was set to the average size of 1 GOF. The standard deviation σ in Equation (3) was set such that the standard deviation of the available bandwidth was 30 Kbps (the mean available bandwidth was varied from 172 Kbps to 258 Kbps). Thus, the σ interval, over which the bandwidth fluctuates most often is approximately 100 Kbps, or roughly between $1/2$ and $1/3$ the mean transmission rate. The results were averaged over multiple realizations of the channel to obtain statistically meaningful results.

We compare the performance of four systems: *System 1: 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 $B(t) \geq \frac{g_1}{g_1+g_2+g_3}B_0$. *System 2: Distortion-buffer optimized streaming [1]* This system computes the streaming policy π using the algorithm proposed in [1]. The horizon of the optimization was set to 26 packets, or 2 GOFs. Thus, the streaming policy was selected from among 2^{26} policies using the ISA algorithm proposed in [3]. *System 3: Single packet transmission policy* – This system used the proposed fast policy selection approach detailed in Section 4. The horizon of the optimization was set to $m = 26$ packets, or 2 GOFs. At each transmission opportunity, the best single packet transmission policy was selected through the minimization of Equation (6). *System 4: Single packet transmission policy using probability approximation* This system is identical to System 3, with the exception that the probability approximation of Section 4.1 was used instead of the actual probabilities. All the systems were implemented in $MATLAB$, it was noted that Systems 3 and 4 require approximately $1/60^{th}$ the computation of System 2, thereby providing a 60 fold improvement in computational complexity.

Figure 2 depicts the PSNR vs. rate plots of the four systems. As expected, the performance of the distortion optimized transmission policy [1], taking the rate-distortion characteristics of the video packets and the channel state into account is superior to that of the other systems. More importantly, as can be seen from the Figure, the performance of the proposed approach is within a fraction of a dB of the distortion-optimized policy. This in our opinion, combined with the fact that the proposed approach provides a multi-fold reduction in computational complexity, demonstrates the efficacy of the proposed work. Further, there is a miniscule loss in performance when using the proposed probability approximation as opposed to using the actual probabilities. This helps in a reduction of the quantities that a streaming server need calculate. Lastly, the performance of the proposed approaches (System 3 and System 4) is far superior to that of the distortion agnostic naïve streaming approach, especially at low bit-rates.

6. CONCLUSIONS

We have proposed a fast distortion-buffer optimization strategy for selecting the transmission policy for multimedia streaming over a reliable but variable bit rate channel. The main contribution of the current work is the approximate factorization of the cost function $J(\pi)$ through a partitioning of the policy π. The efficacy of the proposed single packet transmission policy was demonstrated through empirical evaluations. It was shown that the loss in performance as compared to the optimized streaming policy [1], whose computational complexity is far higher than that of the proposed approach, was minimal. We are currently compiling further simulation results that evaluate the precise impact of the greedy factorization. The final paper will contain further simulations addressing this issue.

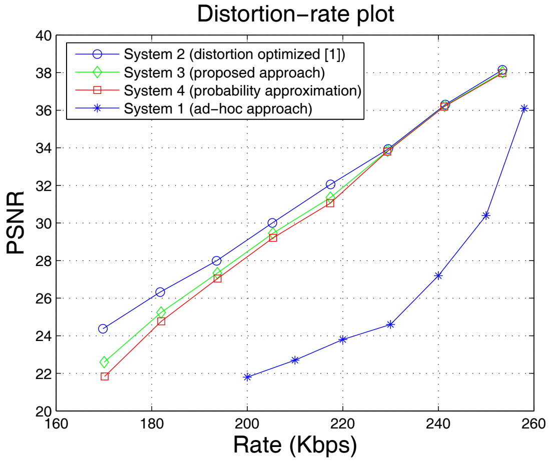

Fig. 2. PSNR vs. rate plot assessing the performance of the proposed approaches (System 3 and System 4) as compared to the Distortion-buffer optimized streaming policy approach of [1] and the distortion agnostic naïve approach.).

7. ACKNOWLEDGEMENTS

The authors wish to thank Dr. Jacob Chakareski, EPFL Switzerland, for his insightful comments on a preliminary version of the current work.

8. REFERENCES

[1] A. Sehgal, O. Verscheure, P. Frossard, "Buffer Optimized TCP Video Streaming," *IEEE ICIP 2004*, Singapore.

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

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

[4] 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.

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

[6] 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.

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