Packetized Media Streaming over Multiple Wireless Channels

Dihong Tian, Yen-Chi Lee, Ghassan AlRegib, and Yucel Altunbasak
School of Electrical and Computer Engineering
Georgia Institute of Technology

Abstract— **This paper addresses the problem of streaming packetized media through a proxy server to a mobile client over multiple wireless channels. We follow a rate-distortion optimized streaming approach. A real-time sender-driven streaming framework is employed at the proxy server to maximize the playback quality at the client. The experimental results show that the proposed algorithm improves the video quality by 1.3-3dB as compared to the conventional system which simply transmits the video data in the order they are displayed.**

I. INTRODUCTION

The rapid growth in the number of both Internet and mobile clients has resulted in developing advanced wireless communication systems. For example, wireless LAN and the third generation (3G) wireless systems provide high data rates that will facilitate a wide variety of applications including real-time and multimedia applications. Next generation wireless systems are envisioned to provide even higher capacities and support heterogeneity in physical environment, network architecture and application. In such heterogeneous environment with a variety of wireless access technologies, it is possible that a mobile host will have multiple wireless interfaces (e.g., in a wireless *overlay* architecture [1]) and is capable of receiving higher quality of media service by aggregating the capacities from multiple physical channels.

In this paper, we consider the scenario when a mobile host has multiple active wireless interfaces to a *proxy* server. We address the problem of streaming packetized media through the proxy server to the mobile host over multiple wireless channels. The scenario under consideration is illustrated in Figure 1. The proxy server is located at the junction between the backbone and the wireless networks [2]. Packetized media data, originated from the media server, will be cached at the proxy and then forwarded to the mobile host. The challenge in this scenario is *how the proxy server can optimally transmit the media data over these multiple channels to achieve the best service quality at the mobile client.*

Motivated by this challenging problem, we develop a transmission method, which is implemented at the proxy server, based on a rate-distortion optimization approach. In the proposed method, the forward links (from the proxy server to the mobile host) are used to carry media data, and the acknowledgements (ACK/NAK's) are sent over the backward links. We model the forward links and the backward links separately with two independent two-state Markovian models. By jointly considering the channel statistics and the characteristics of the

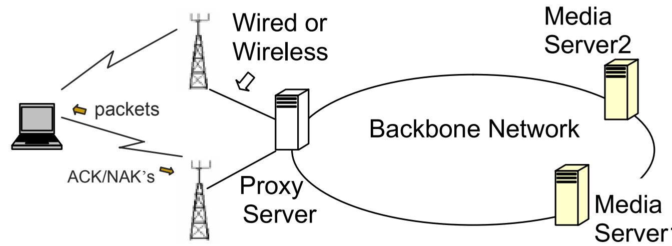

Fig. 1. Streaming media through a proxy server to a mobile host via multiple wireless interfaces.

media representation, the proxy server performs a real-time transmission policy selection algorithm to determine which data units should be delivered over which forward channels and at which time slots. The simulation results show that the proposed method can effectively utilize multiple wireless channels and significantly reduce the playback distortion at the client.

The authors in [3], [4] and [5] address a similar problem to the one we are investigating in this paper. In [3], Chakareski et al. considered streaming packetized media over a lossy packet network through an intermediate server to a wireless client. A hybrid receiver/sender driven transmission scheme was employed to coordinate the communication between the media server and the client. Sehgal et al. [4] proposed a cost-distortion optimized caching algorithm to minimize the cost of storage at the proxy as well as to minimze the distortion at the client. Chakareski et al. [5] presented a rate-distortion optimized sender-driven streaming framework over the Internet using path diversity. All these methods develop a sensitivity adjustment (SA) algorithm, which is computationally expensive, to obtain an optimal (or locally optimal) solution in an iterative way [6]. In this paper, we further address the scenario of the simultaneous use of multiple wireless channels from a proxy server to a mobile client, which has not been considered in the literature. In addition, the proposed method has less computational complexity. Note that we do not consider the problem of streaming media data over the backbone network, which has been addressed in [4] and can be easily integrated with our method. Moreover, the proxy server may pre-store the entire media presentation because of its frequent request by other clients. In such a case, we do not need to consider the problem of streaming media data from the media servers to the proxy.

The rest of this paper is organized as follows. Section II

 0-7803-8533-0/04/$20.00 (c) 2004 IEEE

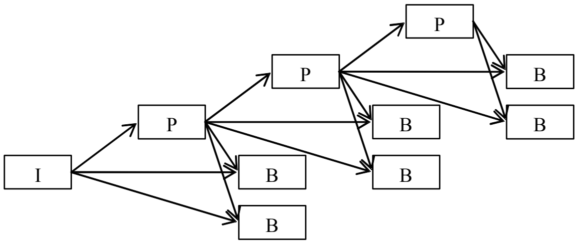

Fig. 2. Dependencies between IBBPBBPBBP video frames.

introduces the models used for the media source and the wireless channel. In Section III, we present the basic rate-distortion optimized streaming framework. The transmission policy selection algorithm is described in Section Section IV. The experimental results are presented in Section V and Section VI concludes the paper.

II. PRELIMINARIES

In this section, we briefly describe how we model the predictively coded media source and the wireless channel, in preparation for developing our major framework hereafter.

A. Source model

In a media streaming system, the encoded data is packetized into *data units* and stored at one or many media servers. Because of predictive coding, all of these data units have interdependencies and can be expressed by a directed acyclic graph [6]. Figure 2 shows the dependency graph of a typical encoded video as an example. Each node in the graph corresponds to a data unit, and each edge with the direction from data unit k to k' implies that the data unit k' cannot be decoded unless data unit k is first decoded. For each data unit k, we define three quantities:

- B_k: the size of data unit *k* in bytes,
- $t_{DTS,k}$: the decoding deadline for data unit *k*, and
- ΔD_k: the *importance* of data unit k.

The importance, ΔD_k, is the amount by which the distortion at the receiver will *decrease* if the data unit is decoded on time. The decoding deadline, $t_{DTS,k}$, for data unit k is the time by which the data unit k must arrive at the client, or be too late to be usefully decoded. If a packet arrives at the destination after its decoding deadline, they will simply be discarded.

B. Channel model

The block error process in a wireless fading channel can be well approximated by means of a first-order Markov model [7], [8]. The system parameters which affect the Markov description are defined by a transition matrix

$$M(x) = \begin{pmatrix} p(x) & q(x) \\ r(x) & s(x) \end{pmatrix} = \begin{pmatrix} p & q \\ r & s \end{pmatrix}^x,$$

where $p(x) = 1 - q(x)(r(x) = 1 - s(x))$ is the probability that the i-th block is successfully transmitted, given that the transmission of $(i-x)$-th block was successful (unsuccessful).

There are essentially two independent parameters for completely specifying this first-order Markov model. In most of the literature, the transition probability $r = P\{success|failure\}$, and the steady-state packet error probability,

$$\varepsilon = 1 - \frac{r}{1-p+r},$$

are studied instead of evaluating p directly. This choice is motivated by the fact that these two parameters have an immediate physical significance: ε, as mentioned above, is the average packet error rate, measuring how often a packet is corrupted; $1/r$ is the average error burst length, and gives an idea about how clustered the errors tend to be.

Consider that a mobile client has L connections to the proxy server. In other words, L pairs (forward and backward) of channels are available for the proxy server to transmit the data units and receive the ACK/NAK's from the mobile client. Assuming the error processes in the channels are independent from each other, we denote the average error probabilities for each pair (forward and backward) of channels as $(\varepsilon_{l,f}, \varepsilon_{l,b}),\ l = 1, 2, \cdots, L$, and the corresponding transition matrices by $(M_{l,f}, M_{l,b}),\ l = 1, 2, \cdots, L$.

III. RATE-DISTORTION OPTIMIZATION

The proxy server caches the data units received from the media servers and transmits them to the mobile client via multiple wireless channels. Suppose that there are K data units being cached for transmission and L channels are available for delivery. For simplicity, throughout this paper, we assume that each channel can carry only one data unit at each transmission opportunity.

Each data unit k is implicitly labeled by a set of transmission opportunities, $\{t_{0,k}, t_{1,k}, ..., t_{N_k-1,k}\}$, prior to its decoding deadline, $t_{DTS,k}$. That is, each data unit k can be considered to be transmitted at most N_k times until either it reaches its decoding deadline or the proxy receives the ACK. At every transmission opportunity, L transmission slots are available. Typically, the number of data units is larger than the number of available transmission slots, *i.e.,* not every cached data unit can be sent at each transmission opportunity. The proxy server needs to choose the data units to be sent at each transmission opportunity to maximize the quality on the client side. It also has to decide on which channels those data units should be sent. In order to achieve the best service quality, we propose that this decision should be made in a rate-distortion optimized way, by taking into account the characteristics of each channel, the feedback information, the importance and the decoding deadline of each data unit.

A general framework is given as follows. Let $\boldsymbol{\Pi} = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_K\}$ be the *transmission policy set* for K data units, where $\boldsymbol{\pi}_k = (\pi_{k,1}, ..., \pi_{k,L}), k \in \{1, ..., K\}$, is the *transmission policy vector* for data unit k on all channels. The *transmission policy*, $\pi_{k,l}$, contains the transmission actions for data unit k on channel l at all transmission opportunities prior to its decoding deadline. A transmission action is either to send or not to send, denoted by 1 and 0, respectively. For example, the transmission policy, $\pi_{k,l} = \{1, 0, 1, 0, 0\}$, means that data unit k will be sent on channel l at times t_0 and t_2, and will not be transmitted on channel l at times t_1, t_3, and t_4. Therefore, every given transmission policy set $\mathbf{\Pi}$ induces an expected playback distortion $D(\mathbf{\Pi})$ and an expected transmission rate $R(\mathbf{\Pi})$. Our goal is to seek the transmission policy set $\mathbf{\Pi}$ that minimizes $D(\mathbf{\Pi})$ subject to a constraint on $R(\mathbf{\Pi})$. This can be achieved by minimizing the following Lagrangian with a Lagrange multiplier λ,

$$J(\mathbf{\Pi}) = D(\mathbf{\Pi}) + \lambda R(\mathbf{\Pi}),\ \lambda > 0. \tag{1}$$

The optimal transmission policy set is the one that minimizes $J(\mathbf{\Pi})$. That is,

$$\mathbf{\Pi}^* = \arg\min_{\mathbf{\Pi}} J(\mathbf{\Pi}). \tag{2}$$

We now compute the expected distortion, $D(\mathbf{\Pi})$, and the expected transmission rate $R(\mathbf{\Pi})$ for a given transmission policy set. The expected transmission rate $R(\mathbf{\Pi})$ is the sum of the expected number of bytes transmitted for each data unit $k \in \{1, 2, \cdots, K\}$ on each channel $l \in \{1, 2, \cdots, L\}$,

$$R(\mathbf{\Pi}) = \sum_{k=1}^{K} \sum_{l=1}^{L} B_k \rho(\pi_{k,l}), \tag{3}$$

where $\rho(\pi_{k,l})$ is the expected cost per source byte (number of transmissions) under transmission policy $\pi_{k,l}$.

To express the expected distortion $D(\mathbf{\Pi})$, we first assume that D_0 is the initial distortion if no data units are received. If data unit k is decoded on time, the distortion will be reduced by the quantity ΔD_k. The expected distortion reduction for data unit k with transmission policy vector $\boldsymbol{\pi}_k$ is

$$D(\boldsymbol{\pi}_k) = \Delta D_k \prod_{k' \preceq k} (1 - \prod_{l=1}^{L} \epsilon(\pi_{k',l})), \tag{4}$$

where $k' \preceq k$ means that data unit k' must be decoded before data unit k can be decoded. $\epsilon(\pi_{k',l})$ is the probability that data unit k' is corrupted if it is transmitted over channel l under policy $\pi_{k',l}$. Therefore, the term $\prod_{k' \preceq k} (1 - \prod_{l=1}^{L} \epsilon(\pi_{k',l}))$ in Equation (4) gives the probability that the receiver decodes the data unit k on time.

The expected distortion can be computed by subtracting the expected distortion reduction from D_0 as follows:

$$D(\mathbf{\Pi}) = D_0 - \sum_{k=1}^{K} D(\boldsymbol{\pi}_k). \tag{5}$$

Now, we can find the transmission policy set by minimizing the Lagrangian $J(\mathbf{\Pi})$, where

$$\begin{aligned} J(\mathbf{\Pi}) &= D(\mathbf{\Pi}) + \lambda R(\mathbf{\Pi}) \\ &= D_0 - \sum_{k=1}^{K} \Delta D_k \prod_{k' \preceq k} (1 - \prod_{l=1}^{L} \epsilon(\pi_{k',l})) \\ &\quad + \lambda \sum_{k=1}^{K} \sum_{l=1}^{L} B_k \rho(\pi_{k,l}). \end{aligned} \tag{6}$$

The optimization in Equation (2), however, is difficult because the terms involving the individual policies $\pi_{k,l}$ in $J(\mathbf{\Pi})$ are not independent. An iterative descent algorithm proposed in [6], called sensitivity adjustment (SA) algorithm, could be extended and applied to find a (suboptimal) solution in Equation (2). However, this iterative algorithm has high computational complexity and therefore is not implementable at proxy servers. We instead propose a less complex transmission method as detailed in the following section.

IV. The Proposed Transmission Policy Selection Algorithm

In this section, we develop a transmission policy selection algorithm with low computational complexity based on a rate-distortion optimization method. We estimate the expected distortion for each data unit using the expected run-time distortion ($ERTD$). This distortion measure depends on the current channel statistics, the decoding deadline, and the data units importance. At time slot i, $ERTD$ value associated with data unit k on channel l is defined as follows:

$$d_{k,l}^{(i)} = \eta_{k,l}^{(i)} \sum_{k \preceq k'} \Delta D_{k'}, \tag{7}$$

where $k = 1, \cdots, K,\ l = 1, \cdots, L$, and

$$\eta_{k,l}^{(i)} = M_{l,f}^{x_l}(\varsigma_{l,f}^{(i-x_l)}, 2) \cdot (M_{l,f}(2,2))^{(N_k - 1)}$$

is the corruption probability for data unit k if it was always chosen to be transmitted on channel l but was finally expired before success. Time slot $(i - x_l)$ is the most recent slot in which the proxy receives an acknowledgment (ACK or NAK). $\varsigma_{l,f}^{(i-x_l)} = 1$ implies that the transmission on channel l at slot $(i - x_l)$ is successful and $\varsigma_{l,f}^{(i-x_l)} = 2$ refers to the case when the transmission on channel l at slot $(i - x_l)$ is unsuccessful. N_k is the number of transmission opportunities that data unit k can be considered to be transmitted.

For each data unit k, we can find the minimum expected run-time distortion ($ERTD_{min}$) among its all $ERTD$ values.

$$\overline{d}_k^{(i)} \triangleq d_{k,l_k}^{(i)} = \min_{l \in 1,\dots L} d_{k,l}^{(i)}, \tag{8}$$

where $l_k = \arg \min_{l \in 1,\dots L} d_{k,l}^{(i)}$ is the corresponding channel on which data unit k has the minimum $ERTD$.

With the set of $ERTD_{min}$ of all data units, we order the list in (9) below, and then determine the transmission policies of the data units. In other words, data unit k_1, whose $ERTD_{min}$ ranks first in (9), will first be considered to find its optimal transmission policy on the corresponding channel l_{k_1}.

$$\overline{d}_{k_1}^{(i)} \geq \overline{d}_{k_2}^{(i)} \geq \cdots \geq \overline{d}_{k_K}^{(i)}, \quad k_j \in \{1, 2, ..., K\}, \tag{9}$$

The problem now is to find the optimal transmission policy for a single data unit k on a single channel l. We use a Markov decision process, the details of which will be later explained

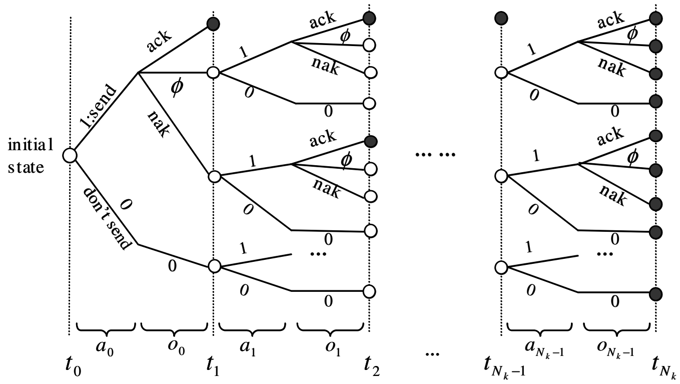

Fig. 3. Markov decision tree for a single data unit.

in this section, to compute the optimal transmission policy. We achieve this by minimizing

$$\begin{aligned} J(\pi_{k,l}) &= D(\pi_{k,l}) + \lambda R(\pi_{k,l}) \\ &= \Delta D_k \epsilon(\pi_{k,l}) + \lambda B_k \rho(\pi_{k,l}). \end{aligned} \quad (10)$$

The optimal transmission policy, $\pi_{k,l}^*$, is obtained correspondingly by

$$\begin{aligned} \pi_{k,l}^* &= \arg\min_{\pi_{k,l}} \left(\Delta D_k \epsilon(\pi_{k,l}) + \lambda B_k \rho(\pi_{k,l}) \right) \\ &= \arg\min_{\pi_{k,l}} \left(\epsilon(\pi_{k,l}) + \lambda_k \rho(\pi_{k,l}) \right), \end{aligned} \quad (11)$$

where $\lambda_k = \lambda \frac{B_k}{\Delta D_k}$.

With the optimal transmission policy $\pi_{k,l}^*$, the transmission action for data unit k on channel l at the current time slot i is determined, either to send or not to send. Suppose that data unit k_j is the first data unit in list (9) which has "Send" as the transmission action on channel l_{k_j} at the current time slot. Then, slot i on channel l_{k_j} will be allocated to data unit k_j and will not be considered for the other data units. Furthermore, the $ERTD$ value of data unit k_j on the rest channels will be updated using

$$d_{k_j,l}^{(i)} = d_{k_j,l}^{(i)} \epsilon(\pi_{k_j,l_{k_j}}), \quad l \neq l_{k_j}, \quad (12)$$

which implies that the importance of data unit k_j decreases because it has been selected to be sent on channel l_{k_j}.

Based on the updated expected run-time distortions of the data units and the available channels, the $ERTD_{min}$ list (9) is updated accordingly. This selection process will be repeated until all channels are allocated.

A. Finding the transmission policy for a single data unit

In the remaining of this section, we explain in detail how to find the optimal transmission policy for data unit k on channel l. A Markov decision process, as shown in Figure 3, is used to determine the best transmission actions at any transmission opportunity.

At each transmission opportunity $t_i, i = 0, 1, ..., N_k - 1$, the sender can choose either to send ($a_i = 1$) or not to send ($a_i = 0$) data unit k on channel l. If the sender chooses to send the data unit, then just prior to the next transmission opportunity t_{i+1}, the server can observe that either an acknowledgement (ACK or NAK) for that data unit has been received, or no packet has been acknowledged, in which case it is concluded that the transmission on the feedback channel was corrupted.

Observing an ACK at t_i makes the process enter a final state (solid circles in Figure 3) at time t_{i+1}. In addition, any state at time $t_{N_k} = t_{DTS,k}$ is a final state. In other cases the process enters a non-final state at t_{i+1}, and the sender can once again choose either to send or not to send the data unit.

Each state in the tree captures the action-observation history leading up to that state from the initial state. That is, a state q_i at time t_i represents a sequence of i action-observation pairs, $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{i-1}, o_{i-1})$. The action a_i at a non-final state q_i determines the transition probability $P(q_{i+1}|q_i, a_i)$ to the next state q_{i+1}. From Fig. 3, it is apparent that $P(q_{i+1}|q_i, a_i = 0) = 1$ since q_{i+1} will be the unique resulting state from state q_i if action a_i is 0.

The expression of transition probability $P(q_{i+1}|q_i, a_i = 1)$ is conditioned on the observed feedback on the backward channel. A computational example is given next and more expressions can be derived likewise. Assume $o_{i-1} = \phi$ and $o_i = ACK$ is observed before entering next state q_{i+1}, which implies that in time slot i the transmission on both (forward and backward) channels is correct, whereas the transmission on the forward channel in slot $(i-1)$ is unknown due to the corruption on feedback. The transition probability is then given by

$$P(q_{i+1}|q_i, a_i = 1) = M_f^{1+x_l}(2,1) \cdot M_b(2,1),$$

where time slot $(i - x_l)$ is the most recent slot in which the transmission result on the forward channel is already known.

Let q_F be a final state with history $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{F-1}, o_{F-1})$, and let $q_{i+1} = q_i \circ (a_i, o_i)$, $i = 1, ..., F-1$, be the sequence of states leading up to q_F. Then, q_F has the transmission cost $\rho(q_F) = \sum_{i=0}^{F-1} a_i$ and error $\epsilon(q_F) = 0$ if o_{F-1} is an ACK; otherwise, $\epsilon(q_F)$ is equal to the probability that none of the transmitted packets arrives the client. In mathematical form,

$$\epsilon(q_F) = \begin{cases} 0 & \text{if } o_{F-1} = ACK; \\ \prod_{j=0}^{F-1}{}_{a_j=1 \& o_j=\phi} M_f^{x_j}(\varsigma_f^{(x_j)}, 2) & \text{otherwise,} \end{cases} \quad (13)$$

where, for slot j in which the action is "Send" but no acknowledgement was observed, slot $(j - x_j)$ is the most recent slot in which the transmission result on the forward channel $\varsigma_f^{(x_j)}$ is implicitly given with an acknowledgement observed on the feedback channel.

With the expected error and transition probabilities, we can use dynamic programming [9] to find the optimal transmission policy π^* that minimizes the expected Lagrangian:

$$J^*(q_i) = \begin{cases} \epsilon(q_F) + \lambda \rho(q_F) \\ \qquad \text{if } q_i = q_F \text{ is a final state;} \\ \min_a \sum_{q_{i+1}} P(q_{i+1}|q_i, a) J^*(q_{i+1}) \\ \qquad \text{otherwise,} \end{cases} \quad (14)$$

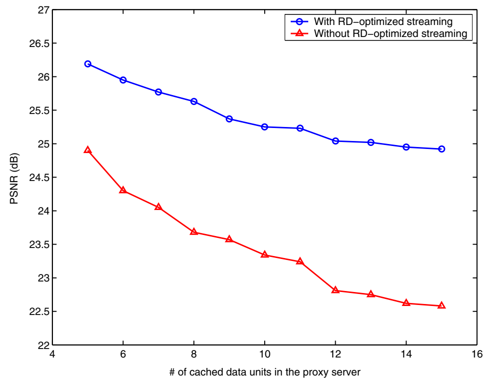

Fig. 4. Playback quality with respect to various numbers of cached data units (the number of available channels is fixed to 2 and PLR = 0.2).

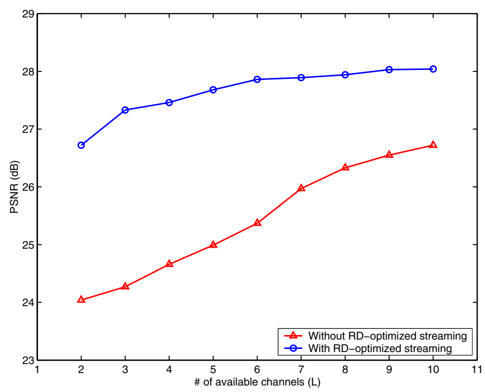

Fig. 5. Playback quality with respect to various numbers of available channels (30 data units are cached for transmission and PLR = 0.1).

where

$$\pi^*(q_i) = \arg\min_a \sum_{q_{i+1}} P(q_{i+1}|q_i, a) J^*(q_{i+1}) \tag{15}$$

is the optimal action at the current transmission slot i.

V. Experimental Results

We present the experiments conducted on one video sequence, COASTGUARD, to evaluate the performance of the proposed algorithm. This video sequence is in the format 4:2:0, 176×144 pixels per frame, and 30 frames per second. We encode this sequence using MPEG-1, targeting at a constant bit rate of 512 Kbps. Peak signal-to-noise ratio (PSNR) is used to perform objective comparisons.

In order to demonstrate the effectiveness of the proposed method, we design two sets of experiments. In the first set, we fix the number of available channels to two but vary the size of grouped data units in the proxy server. Large group size implies that more data units are cached, which should be considered for transmission to the receiver before their decoding deadlines are expired. The average packet loss rate of each channel is 20%. Figure 4 shows the PSNR comparisons of the proposed method and the conventional heuristic method, which simply chooses the data units whose decoding deadlines are the closest and transmits them over the channels with the lowest error probabilities at the current transmission slot. As we can see, our method improves the playback quality by roughly 1.3-2.5 dB as compared to the heuristic method. One may observe that as the number of cached data units increases, the performance for both methods decreases. This is because when the cached data units increases, more data units need to be transmitted, but only two data units could be sent at each transmission slot. As a result, some data units may fail to be delivered because of the expiration of their decoding deadlines. Nonetheless, the proposed method provides more graceful quality degradation than the conventional system as demonstrated in Figure 4.

In the second set of experiments, we fix the number of cached data units to thirty and vary the number of active channels from the proxy server to the mobile client. The mean packet loss rate of each channel is 10%. Figure 5 shows the PSNR comparisons. The proposed method outperforms the heuristic method by about 1.3-3.0 dB. It is noted that as the number of connections between the proxy and the mobile client increases, the performance of both methods increases. However, the performance for the proposed method saturates when the number of available channels is larger than six. Therefore, by adopting the proposed transmission mechanism, the streaming media system can achieve higher quality with fewer connections between the proxy server and the client.

VI. Conclusions

In this paper, we investigated the problem of streaming packetized media through a proxy server to a mobile client over multiple wireless interfaces. A real-time sender-driven transmission method that operates at the proxy server was proposed to minimize the end-to-end distortion in a rate-distortion optimized way. The experimental results illustrated that the proposed method can significantly improve the video quality by 1.3-3.0 dB.

References

[1] M. Stemm and R. Katz, "Vertical Handoffs in Wireless Overlay Networks, *ACM/Baltzer Mobile Networking and Applications*, vol. 3, No. 4, pp. 319-334, Jan. 1999.

[2] G. Cheung and T. Yoshimura, "Streaming agent: A network proxy for media streaming in 3G wireless networks," in *Proc. Packet Video Workshop*, Pittsburgh, Apr. 2002.

[3] J. Chakareski and P. A. Chou, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients", Microsoft Research Technical Report MSR-TR-2002-81, August 2002.

[4] A. Sehgal and P. A. Chou, "Cost-distortion optimized caching of streaming media", in *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, Orlando, FL, May 2002.

[5] J. Chakareski and B. Girod, "Rate-Distortion Optimized Packet Scheduling and Routing for Media Streaming with Path Diversity", in *Proc. IEEE Data Compression Conference*, Snowbird, UT, Apr. 2003.

[6] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of pacetized media", *IEEE Trans. on Multimedia*, submitted.

[7] M. Zorzi, R. R. Rao, L. B. Milstein, "On the accuracy of a first-order Markov model for data transmission on fading channels", in *Proc. Universal Personal Communications, Fourth IEEE International Conference on*, Nov 1995.

[8] M. Zorzi, R. R. Rao, "Throughput analysis of ARQ selective-repeat protocol with time diversity in Markov channels", in *Proc. Global Telecommunications Conference, GLOBECOM '95., IEEE*, Nov 1995.

[9] T. K. Moon and W. C. Srirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, 1999.