RATE-DISTORTION OPTIMIZED VIDEO STREAMING FOR SCALABLE H.264

Sangho Yoon, Mark Mao and Mark Kalman

Information Systems Lab., Department of Electrical Engineering, Stanford University
{holyoon,markmao,mkalman}@stanford.edu

ABSTRACT

We propose a new real-time packet scheduling algorithm for streaming scalable H.264. Our algorithm makes use of a packet importance measure, which we define, that takes into consideration transmission history, channel conditions, and the unique decoding dependencies due to the temporal wavelet encoding. Our algorithm utilizes this importance measure to minimize the expected reconstruction distortion at the decoder under a certain rate constraint. In our experimental results we show gains of more than 3 dB in decoded video quality when transmissions are controlled with our algorithm as compared to existing schedulers.

1. INTRODUCTION

In a real-time Internet streaming system, encoded media data are packetized and transmitted over the Internet from a server to a client which, after a short delay, plays the media in real time. Problems occur because of the best-effort nature of the Internet. Packets transmitted over the Internet may be lost or not delivered to the client in time. In addition, transmissions are often limited by rate constraints imposed by transmission speeds of links or because of congestion control guidelines.

When transmission rate is constrained, when the media encoding offers rate-scalability, and when retransmission is used to handle lost packets, it is not obvious what packets should be transmitted when. We need to decide which packets to transmit and when to transmit them to optimize the playback quality at the client given the rate constraint and the ongoing delivery performance of the channel.

In rate-distortion optimized streaming, we want to minimize the expected distortion given a rate constraint [1]. This problem can be formulated by the following Lagrangian cost function:

$$J(\pi) \quad = \quad D(\pi) + \lambda R(\pi) \qquad (1)$$

where, π is a policy governing the transmission of L data units, and $D(\pi)$ and $R(\pi)$ are the expected distortion and expected transmission rate for the transmission policy π.

Finding the optimal policy that minimizes $J(\pi)$ in (1) can easily be intractable. For example, suppose we have L packets and for each packet the transmission policy governs whether the packet will be transmitted or not over the course of a time horizon of N discrete transmission opportunities. Then π in (1) can be expressed as a policy vector ($\pi = (\pi_1, \pi_2, .., \pi_L)$) with each component policy π_i governing the transmissions of a particular packet. The π_i can be expressed, in turn, as length-N binary vectors with the elements indicating whether the packet will be transmitted or not at each of the N opportunities. In this case, there are 2^{LN} possible policies π, and the number of policies and thus the complexity of (1) grows exponentially in the number of packets as well as the number of transmission opportunities [4]. Chou et al. [1] solved this problem by using an iterative descent algorithm where they optimize the Lagrangian of each π_l shown below separately until (1) converges.

$$J_l(\pi_l) \quad = \quad d(\pi_l) + \lambda_l r(\pi_l) \qquad (2)$$

This effectively decouples the packet dependencies and can reduce the complexity of (1) to be roughly proportional to the number of packets. However, the algorithm may still not be feasible for real-time streaming because (2) is still exponential in the number of transmission opportunities (=N) and λ_l is adjusted by iterations to meet a rate constraint.

To reduce searching complexity in (1) and realize real time streaming, in our algorithm we only consider transmitting one packet at a time. To further reduce searching complexity, we transmit the packet with the highest importance measure (defined later) instead of searching the entire policy space exhaustively to transmit one packet at each transmission time. An important feature of our algorithm is that it can take into account the unique packet decoding dependency structure of scalable H.264.

Similar work was done by Miao et al. [3]. They proposed an algorithm for scalable media streaming. Their algorithm tries to maximize the quality of reconstructed media at the client by on-line packet scheduling. Their scheduling algorithm computes the expected distortion for each packet based on the transmission history and packet dependencies. Their distortion measure is simple and fast. However, in their distortion model, they simply modelled the channel as a fixed value of packet loss probability without consideration of channel delay. More importantly, the rate constraint was not incorporated into their distortion measure.

1-4244-0367-7/06/$20.00 ©2006 IEEE

2. BACKGROUND

The new H.264 standard promises higher quality video transmission for both high and low bandwidth networks. In order to improve the performance in case of varying link quality, a scalable version [5] of this standard has been recently proposed. The scalable H.264 partitions the compressed video data into layers so that different qualities of video can be transmitted according to the availability of network bandwidth. This scalability makes it ideal for video streaming over the Internet or wireless networks where available bandwidth fluctuates over time.

The scalable H.264 coder provides three aspects of scalability: temporal, spatial and quality (SNR). For simplicity, in our work the spatial scalability provided by the coder is not used. See [5] for more details.

In our work, an open GOP (Group Of Pictures) structure as in [5] is used. For each SNR layer we have the following GOP structure. The first picture is always coded as a single I (or more accurately IDR [Instantaneous Decoder Refresh]) picture. The remainder of the stream is coded in groups of 16 pictures, with anchor pictures at the end and 15 hierarchically coded B pictures between each pair of anchor pictures.

I N × [B4 B3 B4 B2 B4 B3 B4 B1 B4 B3 B4 B2 B4 B3 B4 I]

Fig. 1. GOP structure: the first picture is coded as a single I and is followed by N GOPs, each having 16 pictures.

Fig. 2 shows the structure of one GOP for three SNR layers (one base layer and 2 enhancement layers). At the decoder side, for the base layer, B1 packets can only be decoded when the I packet of the current GOP and the I packet of the previous GOP are decoded. The B2 packets can only be decoded when the closest I packet and B1 packet are decoded. The dependency is the same for B3 and B4 packets. Each packet depends on the closest lower level packets that precede and follow it.

```
SNR L2         B4  B4  B4  B4  B4  B4  B4  B4
                 B3      B3      B3      B3
                     B2              B2
                             B1
             I                                 I
SNR L1         B4  B4  B4  B4  B4  B4  B4  B4
                 B3      B3      B3      B3
                     B2              B2
                             B1
             I                                 I
SNR L0         B4  B4  B4  B4  B4  B4  B4  B4
                 B3      B3      B3      B3
                     B2              B2
                             B1
             I                                 I
```

Fig. 2. GOP structure with three SNR layers

For the enhancement layers, the dependency is somewhat different. For example, if a B3 packet in the second enhancement layer is lost, the adjacent B4 packets can still be decoded as long as the corresponding B4 packets in the base and first enhancement layers and the corresponding B3 packet in the base layer are decoded. The distortion removed as a result of the second enhancement layer B4 packet being decoded will be less, however, than when the B3 packet in the second enhancement is available. This is an example of an indirect dependence relationship, discussed in Sec. 3.

3. SCHEDULING ALGORITHM BASED ON IMPORTANCE MEASURE

3.1. Weighted Distortion Reduction

If packet l is decodable by the receiver on time, then the reconstruction distortion is reduced by Δd_l. For packet l to be decodable, all packets that packet l is dependent on must arrive on time. Otherwise, packet l cannot be decoded even if it is delivered on time. Thus, the expected reconstruction error resulting from transmitting packets based on a policy π becomes

$$\begin{aligned} D(\pi) &= D_0 - \Sigma_l \Delta d_l \prod_{l' \in M(l)} (1 - P_e(l', \pi)) \quad (3) \\ &= D_0 - D_c(\pi) \quad (4) \end{aligned}$$

where, $M(l)$ is a set of packets that packet l depends on, $P_e(l', \pi)$ is the loss probability of packet l' under policy π, and we assume π transmits one packet at a time.

The average transmission rate for π is

$$R(\pi) = \Sigma_l B_l \rho(l, \pi) \quad (5)$$

where B_l is the packet size in bytes, and $\rho(l, \pi)$ is the number of transmissions of packet l under policy π.

The optimal policy π^* subject to rate constraint is

$$\begin{aligned} \pi^* &= \text{argmin}_{\pi, R(\pi) \leq R} D(\pi) \quad (6) \\ &= \text{argmax}_{\pi, R(\pi) \leq R} D_c(\pi) \quad (7) \end{aligned}$$

As discussed earlier, exhaustive search is not suitable for real-time scheduling. In addition, the quantity Δd_l can not be uniquely defined when there is a highly complicated dependency structure. For the scalable H.264 considered in this paper, there can be two kinds of dependencies between packets: direct and indirect. If there is a direct dependency between packets, then a child packet can only be decoded when the parent packet is received. In an indirect relationship, however, a child packet can still be decoded without the indirect parent. The missing indirect parent packet will affect only the amount of distortion reduction of the child packet. This would necessitate defining multiple Δd_l for packet l depending on which direct and indirect parent packets are available to decode packet l.

To avoid having multiple Δd_l for each packet, we define a weighted distortion reduction. We first measure the distortion reduction Δd_l as the decrease in distortion by decoding packet l assuming all of its direct and indirect parents are available. Then, to take indirect dependencies between packets into account, Δd_l is weighted by w_l ($0 \leq w_l \leq 1$):

$$\Delta \widehat{d_l} \quad = \quad w_l \Delta d_l \tag{8}$$

Suppose packet l corresponds to the i^{th} layer of a certain picture in Fig. 2. We denote $k(l)$ as the picture which packet l belongs to. Assuming all parent packets are available, we call the distortion reduction of i^{th} layer of picture $k(l)$, which is packet l, $\Delta d_i(k(l))$, and we define the following weight for the scalable H.264 dependency structure shown in Fig. 2:

$$w_l \quad = \quad \prod_{l' \in M(l)} p_{l'} \tag{9}$$

where influence factor $p_{l'}$ is

$$p_{l'} \quad = \quad \begin{cases} \frac{\Delta d_0(k(l'))}{\sum_{i=0}^{2} \Delta d_i(k(l'))}, \text{ if } l' \text{is available} \\ 0, \text{otherwise} \end{cases} \tag{10}$$

if packet l' is in the base layer,

$$p_{l'} \quad = \quad \begin{cases} \frac{\Delta d_0(k(l'))+\Delta d_1(k(l'))}{\Delta d_0(k(l'))}, \text{ if } l' \text{is available} \\ 1, \text{otherwise} \end{cases} \tag{11}$$

if packet l' is in the first enhancement layer,

$$p_{l'} \quad = \quad \begin{cases} \frac{\sum_{i=0}^{2} \Delta d_i(k(l'))}{\Delta d_0(k(l'))+\Delta d_1(k(l'))}, \text{ if } l' \text{is available} \\ 1, \text{otherwise} \end{cases} \tag{12}$$

if packet l' is in the second enhancement layer, and

$$p_{l'} \quad = \quad \begin{cases} 1, \text{ if } l' \text{is available} \\ 0, \text{otherwise} \end{cases} \tag{13}$$

if packet l' and packet l are in the same picture

w_l is the influence of parent packets on Δd_l, and we empirically estimate w_l by using the influence factors. The distortion decreases by Δd_l when we decode packet l if all of its direct and indirect parents are available. However, when some of the indirect parents are not available at decoder, the distortion decrease is less than Δd_l. By using w_l based on the transmission history, we can approximate the actual distortion reduction realized when packet l is decoded. Note that the influence of a parent picture k on w_l with all three layers available, two bottom layers available and only base layer available are 1, $\frac{\Delta d_0(k)+\Delta d_1(k)}{\sum_{i=0}^{2} \Delta d_i(k)}$, and $\frac{\Delta d_0(k)}{\sum_{i=0}^{2} \Delta d_i(k)}$, respectively.

3.2. Packet Importance Measure

In our scheduling algorithm, we define an importance measure and transmit the packet with the highest importance measure. We try to incorporate the transmission history of packet l into its importance measure. Intuitively, we do not want to re-transmit those packets sent a short time before. Thus, we compute the probability of future loss conditioned on the knowledge of feedback and the deadline of packet l:

$$P_{\text{e,future}}(l) \quad = \quad \prod_{i=1}^{n(l)} P(FTT > t_{d,l} - t_c | RTT > t_c - t_x(i))$$

where, $n(l)$ is the number of previous transmission trials of packet l, t_c is the current time, $t_{d,l}$ is the dead line of packet l, $t_x(i)$ is the time of the i^{th} transmission of packet l, FTT is the forward travel time and RTT is the round-trip travel time, assuming that the client immediately sends an acknowledgement to the server upon the reception of a media packet.

We also define another probability of loss for past transmission trials of packet l as follows:

$$P_{\text{e,past}}(l) \quad = \quad \begin{cases} 0, \text{ if packet } l \text{ ACKed} \\ 1, \text{ if packet } l \text{ Not yet sent} \\ \prod_{i=1}^{n(l)} P(FTT > t_{d,l} - t_x(i) | RTT > t_c - t_x(i)), \\ \quad \text{if packet } l, \text{ sent } n(l) \text{ times} \end{cases}$$

The importance of packet l can be increased if any of its child packets is available at decoder. Thus we define effective distortion reduction of packet l as follows:

$$\Delta \widetilde{d_l} = P_{\text{e,future}}(l) \left(\Delta \widehat{d_l} \prod_{l_p \in M(l)} (1 - P_{\text{e,past}}(l_p, \pi)) \right. \\ \left. + \sum_{l_c \in C(l)} \Delta \widehat{d}_{l_c} P_{\text{e,past}(l_c, \pi)} \right)$$

where $C(l)$ is a set of child packets of packet l, $\Delta \widehat{d_{l_c}}$ is the weighted distortion reduction in (8) assuming packet l is available at decoder.

Under a rate constraint, we can not always transmit packets with high $\Delta \widehat{d_l}$ regardless of packet size. Thus we define an importance measure $I(l)$ for packet l by normalizing $\Delta \widehat{d_l}$ by packet size B_l.

$$I(l) \quad = \quad \frac{\Delta \widetilde{d_l}}{B_l} \tag{14}$$

3.3. Scheduler with rate constraint

Since we are selecting the most important packet at each transmission trial, we achieve rate control by stopping transmission during the remainder of a given GOP's life time when the total number of bytes transmitted is greater than or equal to the maximum allowable transmission bytes (=$R\Delta T$, ΔT is the life time of a given GOP). Therefore, our algorithm not only transmits important packets, but also meets the rate constraint.

4. EXPERIMENTAL RESULTS

We simulate our algorithm for the transmission of the *foreman*, *mother-daughter*, and *carphone* video sequences encoded with the H.264 scalable coder with three SNR layers

[5] at a frame rate of 15 Hz. We assume that each layer of each frame is placed into one packet, and we use 16 frames per GOP. We model the network as in [1], with independent delay assumptions. Delay is modelled by a shifted Gamma distribution with shift $\kappa = 25$ ms, mean 125 ms, and the standard deviation 35.4 ms. Packets are delayed both in forward and backward directions. The packet loss probability in both forward and backward directions are 0.2. We use a start-up delay of 750 ms.

Our algorithm is compared with a simple heuristic algorithms and Miao's algorithm [3]. The simple heuristic algorithm transmits every packet in the following order from base layer to top enhancement layer, and may retransmit the packet once after checking for an ACK after 350 ms (=mean RTT time + $2\sigma_{RTT}$). Δd_l in (8) is measured in MSE.

I → B1→ B2→ B2→ B3→ B3→ B3→ B3→ B4→ B4→ B4→ B4→ B4→ B4→ B4→ B4

Fig. 3. Simple heuristic algorithm

Fig. 4 - 6 show results for *foreman*, *mother-daughter*, and *carphone* video sequences, and our algorithm outperforms the simple heuristic algorithm and Miao's algorithm.

5. CONCLUSIONS

We propose a real-time scheduling algorithm based on the importance measure in (14). This both reduces searching complexity and improves the decoded quality of media at the client. Weighted distortion in (8) enables our algorithm to be suitable for the complex GOP structure of scalable H.264, where there are both direct and indirect dependencies between packets. We incorporate a delay model into the scheduling algorithm. In addition to that, rate control can be simply achieved in our algorithm without extra cost. Simulation results show that our algorithm is superior to other heuristic algorithms with gains in excess of 3 dB shown.

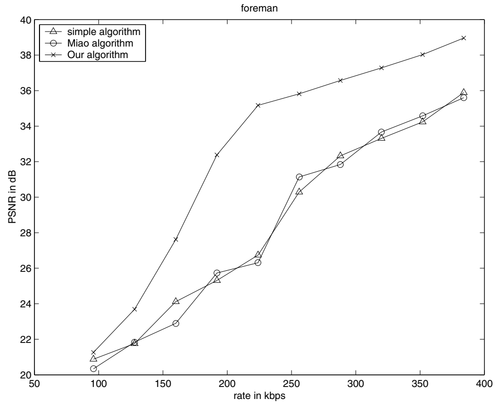

Fig. 4. Rate-Distortion performance: *forman*

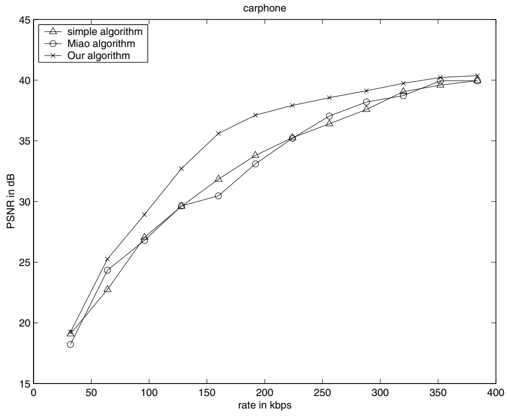

Fig. 5. Rate-Distortion performance: *carphone*

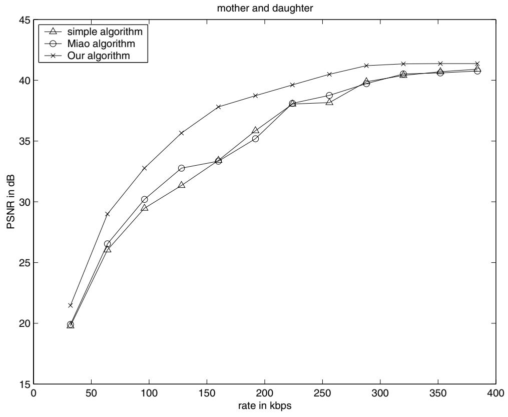

Fig. 6. Rate-Distortion performance: *mother-daughter*

6. REFERENCES

[1] P. A. Chou and Z. Miao, ”Rate-distortion optimized streaming of packetized media,” Microsoft Research, Tech. Rep. MSR-TR-2001-35, Feburary 2001.

[2] M. Kalman, and B. Girod, ”Rate-Distortion Optimized Streaming of Video With Multiple Independent Encodings,” Proc. IEEE International Conference on Image Processing, Singapore, October, 2004.

[3] Z. Miao and A. Ortega,”Optimal Scheduling for the Streaming of Scalable Media ”, Proc. of Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, Oct. 2000

[4] M. Podolsky, S. McCanne, and M. Vetterili, ”Soft ARQ for layered streaming media,” Tech. Rep. UCB/CSD-98-1024, University of California, Computer ScienceDivision, Berkeley, CA, Nov. 1998.

[5] ”Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)14th Meeting”, Hong Kong, CN, 17-21 January, 2005