ADAPTIVE POLICIES FOR REAL-TIME VIDEO TRANSMISSION: A MARKOV DECISION PROCESS FRAMEWORK

Chao Chen, Robert W. Heath Jr., Alan C. Bovik and Gustavo de Veciana

The University of Texas at Austin
Department of Electrical and Computer Engineering
1 University Station C0803, Austin TX - 78712-0240, USA

ABSTRACT

We study the problem of adaptive video data scheduling over wireless channels. We prove that, under certain assumptions, adaptive video scheduling can be reduced to a Markov decision process over a finite state space. Therefore, the scheduling policy can be optimized via standard stochastic control techniques using a Markov decision formulation. Simulation results show that significant performance improvement can be achieved over heuristic transmission schemes.

1. INTRODUCTION

The problem of efficient real-time video transmission over wireless channels is challenging. In the first place, the data throughput is varying over time. In the second place, real-time video delivery can be highly delay-sensitive. To improve the receiver/decoder video quality, the transmitter should optimally allocate bandwidth among current and future frames. In the third place, the video packets are structured. Due to the nature of predictive video coding algorithm, a video frame can be decoded only when its predictor is available. Hence, the prediction structure of the video codec enforces an order on the video packets.

A system with finite state space is called a controlled Markovian system if its state transition probability only depends on the current state and the control action taken at the state. If a instantaneous service quality associated with the system is solely determined by the state, this system is called a Markov decision process (MDP) [1]. The average service quality of the system can be maximized by optimizing the control policy. The MDP-based control framework has previously been proposed in the scenario of real-time video transmission. Indeed in [2], a MDP based formulation was introduced for the problem of real-time encoder rate control. The derived optimal control policy operates at the video encoder adapting the video rate according to the channel conditions and video rate-distortion characteristics. In [3], an MDP formulation was proposed for adaptive video play out and scheduling. The controller controls the play out speed according to the receiver buffer state and channel state to optimize the receiver visual quality. Neither of the above two works considered the adaptive real-time video scheduling. The most closely related work to our paper is by Zhang *et al.* [4], in which a reinforcement learning framework was studied for adaptive video transmission. The optimal transmission policy is obtained via reinforcement learning rather than MDP-based optimization. Hence, the transmitter need to learn a "good" policy from trying those "bad" policies. For real time video delivery, it will degrade the visual quality until the learning is finished.

In this paper, we propose to apply MDP-based stochastic control to real-time video scheduling. Under certain assumptions, the real-time video scheduling can be formulated as a Markov decision process over finite state space. Hence, standard policy optimization algorithms can be employed to derive video scheduling strategies. Different from [4], the scheduling policy is derived off-line and thus is suitable for real-time applications. Simulations results show that substantial gains can be achieved by the optimized scheduling policy.

2. SYSTEM MODEL

We consider a real-time wireless video transmission system with the compressed video stored on a server. The video is sent through a stable TCP/IP network to a wireless router which forwards the video to a mobile user. We assume that the wireless channel between the wireless router and the user is the bottleneck of the link. Our adaptive control policy operates on the wireless router in a frame by frame basis. At the beginning of each frame slot, one frame is displayed and the wireless transmitter schedules a collection of video data for transmission. Video sequences are encoded by an H.264 compatible scalable video encoder and the prediction structure is "I-P-P-P...". We adopt this prediction structure rather than the "Hierarchical B" structure because no structural delay is introduced and this is the most widely used structure for real-time video transmission. Each frame of the video sequence is compressed into L quality layers.

This research was supported in part by Intel Inc. and Cisco Corp. under the VAWN program

Rate-distortion Model For each frame, let ΔR_m be the data rate in the mth layer and Δq_m be its contribution to the visual quality measured in PSNR. For a real video sequence, ΔR_m and Δq_m varies from frame to frame. For simplicity, we only use their average values as brief approximations.

Channel Model and System State As shown in [5], The dynamics of a wireless channel can be modeled by a finite state Markov channel. In this paper, the channel state space is defined as $\mathcal{C} = \{(R_1, p_1), ..., (R_{|\mathcal{C}|}, p_{|\mathcal{C}|})\}$, where (R_i, p_i) is the transmission rate and packet error probability of the ith state. The state transition matrix P is a $|\mathcal{C}| \times |\mathcal{C}|$ matrix with entry $P_{s,t}$ as the transition probability from state (R_s, p_s) to (R_t, p_t). We define the receiver buffer state space as the set of L dimensional vectors $\mathcal{L} = \{(l_1, \cdots, l_L) | l_m \geq 0, 1 \leq m \leq L\}$, in which l_m is the number of received but not displayed frames in the mth layer. At the beginning of each time slot t, the first frame in the window is decoded and the reconstruction visual quality is

$$Q_t(s_t) = \sum_{m=1}^{L} \Delta q_m \times \mathbb{1}(l_m > 0), \tag{1}$$

where $\mathbb{1}(\cdot)$ is the indicator function. The system state $\mathcal{S}$ is defined as the product of the channel state and the receiver buffer state, i.e., $\mathcal{S} = \mathcal{C} \times \mathcal{L}$.

Control Set and Policy For each state $s \in \mathcal{S}$, we define a feasible control set $\mathcal{U}(s)$. Each control $u \in \mathcal{U}(s)$ is a L-dimensional vector $(u_1, \cdots, u_L)$. The entries are the number of frames scheduled for transmission in each layer when action u is taken. The control policy $\mu(s)$ is defined as the mapping from the system state s to an control in set $\mathcal{U}(s)$. Once the scheduler select video data, the data will be transmitted in a frame by frame order as shown in Fig.1. Every video packet is repeatedly transmitted until received. In the following, we assume that the scheduler never schedule the enhancement layers of a frame before its base layer is received because the enhancement layers are decoded based on the base layer.

3. PROBLEM FORMULATION

At each time slot, the scheduler can schedule any subset of video data not previously received. This makes the feasible control set $\mathcal{U}(\cdot)$ very large and optimization intractable. Intuitively, the more enhancement layers that are scheduled, the better instantaneous visual quality is obtained. Meanwhile, the more base layers that are scheduled, the less receiver buffer drainage is likely to happen. It is observed that, when a lot of video data are buffered at the receiver, the receiver buffer is less susceptible to drainage. Hence, it would be beneficial to schedule as many enhancement layers as possible. To this end, we define a window of size W. For any slot t, the scheduler schedules the video frames which to be displayed in the interval $[t, t + W]$ with higher priority. Specifically, the scheduler operates according to the following rules.

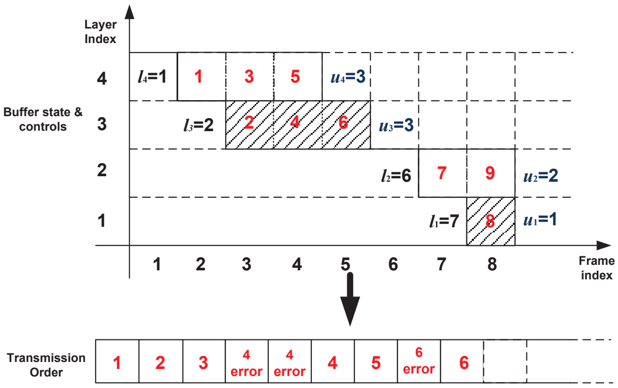

Fig. 1. Receiver buffer state, control and corresponding transmission order.

If all the data within the window are transmitted, the transmitter should schedule as many enhancement layers as possible. If the receiver buffer is empty, the transmitter only schedules the base layer. In other cases, the transmitter chooses data units within the window according to policy $\mu(\cdot)$.

Here, the window size W provides a tradeoff between complexity and optimality. The larger the window, the better the performance and the higher the complexity. By using this window, although the state space is still infinite, we fix the actions outside a finite state space. In other words, we only need to find the optimal policy when the window is neither empty nor fulfilled.

Let $s_t = (C_t, L_t)$ and $\mathcal{U}_(s_t)$ be the system state and the corresponding feasible control set at slot t, respectively. If one frame is decoded at the beginning of the slot and there are $\Delta L_t = (\Delta l_1, \cdots, \Delta l_L)$ frames transmitted for each layer by the end of the slot, we have $L_{t+1} = \lceil L_t - \mathbf{e} \rceil^+ + \Delta L_t$,[1] where $\mathbf{e} = (1, \cdots, 1)$. Assuming the packet length is L_{pkt}, there will be $N = \lceil \frac{\Delta T \times R_t}{L_{pkt}} \rceil$ packet transmissions during a time slot ΔT. Assuming that the packet loss happens independently, at the end of the time slot, the number of successfully transmitted packets is distributed binomially. At time $t + 1$, the number of successfully transmitted packets is at least $N_l = \lceil \frac{\sum_{m=1}^{4} \Delta l_m \Delta R_m \Delta T}{L_{pkt}} \rceil$ but is less than $N_h = \lceil \frac{(\sum_{m=1}^{4} \Delta l_m \Delta R_m + \Delta \tilde{R}) \Delta T}{L_{pkt}} \rceil$, where $\Delta \tilde{R}$ is the data rate in the frame which is scheduled but is not completely received. Hence, the state transition probability from $s_t = (C_t, L_t)$ to $s_{t+1} = (C_{t+1}, L_{t+1})$ is approximately

$$\mathbb{P}_{s_t, s_{t+1}} \approx \left[\sum_{n=N_l}^{N_h - 1} \binom{N}{n} p_t^{N-n} (1 - p_t)^n \right] \times P_{C_t, C_{t+1}}, \tag{2}$$

where the first multiplicative term is the transition probability

[1] $\lceil x \rceil^+ = max\{x, 0\}$

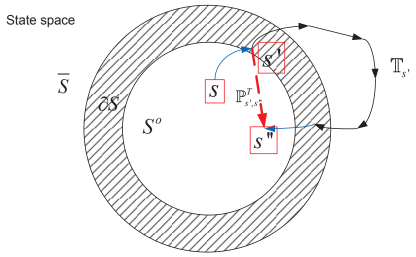

Fig. 2. The behaviors of the original system $\mathcal{A}$.

of receiver buffer state from L_t to L_{t+1} and the second term is the transition probability of channel state from C_t to C_{t+1}.

Our aim is to find the optimal policy $\mu^*(\cdot)$ which maximizes the average visual quality

$$J_\mu(s) = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left\{\sum_{t=0}^{N-1} Q_t(s_t)|s_0 = s\right\}, \forall s \in S. \quad (3)$$

3.1. State Space Reduction

Using the window defined in section 3, we reduced the feasible control set. But, the system state moves in the infinite state space and the MDP algorithm can only operate on a finite state space. To this end, we need to further reduce the state space to a finite one. We define a partition of the state space as follows:

$$\overline{\mathcal{S}} = \{(C,L)|C \in \mathcal{C}; \lceil l_m - 1\rceil^+ \geq W, \forall 1 \leq m \leq L\}\}$$
$$\mathcal{S}^o = \{(C,L)|C \in \mathcal{C}, \lceil l_m - 1\rceil^+ \leq W, \forall 1 \leq m \leq L\}.$$

Given a policy $\mu(\cdot)$, the state will transit as a controlled Markov chain in set $\mathcal{S}^o \cup \overline{\mathcal{S}}$. Let set $\partial\mathcal{S}$ be the subset of $\overline{\mathcal{S}}$ which could be reached from the states in $\mathcal{S}^o$. Because the bandwidth is limited, $\partial\mathcal{S}$ is a finite set. As shown in Fig. 2, once the system moves onto state $\overline{\mathcal{S}}$, it will first visit some state $s' \in \partial\mathcal{S}$ and traverse in $\overline{\mathcal{S}}$ for some time before it visit to some state $s'' \in \mathcal{S}^o$. During this period, the decoded video quality will always be $\hat{Q} = \sum_{m=1}^{L} \Delta q_m$ because the window is always full. Let $\mathbb{T}_{s'}$ be the expected time the system spends in $\overline{\mathcal{S}}$ if it enters $\overline{\mathcal{S}}$ at state $s' \in \partial\mathcal{S}$. Let $\mathbb{P}^T_{s',s''}$ be the probability that the state jumps back to $\mathcal{S}^o$ at state s'' if it enters $\overline{\mathcal{S}}$ from state $S' \in \partial\mathcal{S}$. The following theorem shows that this infinite state problem can be equated to a finite state problem.

Theorem 1. *Given a policy $\mu(\cdot)$, if the associated jump chain*[2] *of the original infinite-state Markov chain is positive recurrent, then the average video quality of the original system $\mathcal{A}$ is the same as the following finite state system $\tilde{\mathcal{A}}$:*[3]

1. *The system is a Markov process over state space $\mathcal{S}^o \cup \overline{\mathcal{S}}$;*

2. *When the system is in one of the states in $s \in \mathcal{S}^o$, it acts according to policy $\tilde{\mu}(s) = \mu(s)$.*

3. *When the system jumps to a state in $s' \in \partial\mathcal{S}$ from $\mathcal{S}^o$, it spends $\mathbb{T}_{s'}$ slots there.After that, the system $\tilde{\mathcal{A}}$ jumps to state $s'' \in \mathcal{S}^o$ with probability $\mathbb{P}^T_{s',s''}$.*

Proof Sketch of Theorem 1. If the jump chain is positive recurrent, the jump from $\mathcal{S}^o$ to $\partial\mathcal{S}$ can partition the Markov process into i.i.d segments. We only need to optimize the policy $\mu(\cdot)$ to maximize the average quality in each segment. Every segment consists of two consecutive subsegments. During the first subsegment, the state $s_t \in \bar{\mathcal{S}}$. In the other subsegment, $s_t \in \mathcal{S}^o$. Because every state in $\bar{\mathcal{S}}$ provide same visual quality $\sum_{m=1}^{L} \Delta q_m$, we can abstract the first subsegment as a single state with transition probability $\mathbb{P}^T_{s',s''}$. This simplified system provide the same average quality as the original system. The detailed proof is not included for lack of space. □

3.2. Computing $\mathbb{T}_s$ and $\mathbb{P}^T_{s,s'}$

Before we apply the standard MDP results to identyfy optimal policies, $\mathbb{T}_s$ and $\mathbb{P}^T_{s,s'}$ need to be determined. When the system moves in $\overline{\mathcal{S}}$, the system always schedules as many enhancement layers as possible, so we can have a one to one mapping between L_t and the quantity $\tilde{k}_t = \sum_{n=1}^{4}(l_n - W)$, i.e., the received video data outside the window. Hence, the state transitions of the system can be modeled as a Markov chain with $(C_t, \tilde{k}_t)$ as the state. All the states in $\overline{\mathcal{S}}$ correspond to some state $\tilde{k}_t > 0$. All the state in $\mathcal{S}^o$ corresponds to some state $\tilde{k}_t <= 0$.

At the beginning of each time slot, the state $\tilde{k}_t$ reduces by $\hat{R} = \sum_{m=1}^{4} \Delta R_m$ because one frame is displayed. Then, the encoder schedules the video data with the best possible quality. At the end of the slot, $\tilde{k}$ is changed by a certain amount that is solely dependent on the channel state C_t with the probability specified in equation (2). Because C_t is Markovian, the state $\tilde{k}_t$ will vary like a random walk but with Markovian step-size. This process can be described by a quasi-birth-death process (QBDP). Hence, determining $\mathbb{T}_s$ and $\mathbb{P}^T_{s,s'}$ is actually the hitting time problem of the quasi-birth-death process. The problem for continuous time QBDP was essentially solved in [6, p. 96]. The discrete time case can also be solved similarly. Due to the limit of the space, we do not elaborate it here. Given the formulation, the optimal policy for a MDP can be determined for the simplified system $\tilde{\mathcal{A}}$, which is also the optimal policy of $\mathcal{A}$. A standard policy optimization algorithm for semi-Markov system can be employed to derive the optimal policy [1, p. 435].

3.3. Modified Policy Iteration Algorithm

Let s_0 be a state in $\mathcal{S}^o \cup \partial\mathcal{S}$. The hitting time to state s_0 can partition the process into into i.i.d cycles. Maximizing the

[2] The jump chain associated with a Markov chain is a Markov chain with the state transitions as its state space.

[3] The simplified system is not coupled with the original system. They just share certain statistical properties.

Table 1. Performance Comparison between Optimized Policy and Heuristic Policy

	Bus		Foreman	
	PSNR	Lost Frames	PSNR	Lost Frames
O	34.8491	0	37.0902	6
H1	34.8897	88	36.9953	112
H2	34.3468	0	36.3332	6
	Mobile		Flower	
	PSNR	Lost Frames	PSNR	Lost Frames
O	33.3675	0	35.3217	0
H1	33.2382	48	35.6844	48
H2	32.9873	0	34.5415	0

average video quality λ in the cycles by optimizing the policy $\mu(\cdot)$, will maximize the average video quality of the system. This is equivalent to the stochastic optimal path problem with stage costs $g(s) - \tau(s)\lambda$, where

$$g(s) = \begin{cases} Q(s) & : \quad s \in \mathcal{S}^o \\ \mathbb{T}_s \hat{Q} & : \quad s \in \partial\mathcal{S}, \end{cases}$$

and

$$\tau(s) = \begin{cases} 1 & : \quad s \in \mathcal{S}^o \\ \mathbb{T}_s & : \quad s \in \partial\mathcal{S}. \end{cases}$$

The optimal policy can be determined via policy iteration, see e.g. [1].

4. SIMULATION RESULTS

The proposed adaptive scheduling algorithm is evaluated on the test sequence of “foreman”, “bus”, “flower” and “mobile”. These video sequences are encoded using H.264\SVC reference software JSVM into 4 layers. The GOP length is set as $L_{GOP} = 16$. We employ a 4-states Markov channel to test the performance of the proposed scheduling algorithm. The state transition matrix is

$$\begin{bmatrix} \frac{1}{5} & \frac{4}{5} & 0 & 0 \\ \frac{1}{5} & \frac{3}{5} & \frac{1}{5} & 0 \\ 0 & \frac{3}{5} & \frac{1}{5} & \frac{1}{5} \\ 0 & 0 & \frac{4}{5} & \frac{1}{5} \end{bmatrix}$$

and the steady state distribution is $\pi = [0.15, 0.60, 0.20, 0.05]$. Denote the throughput of each state by r_1, r_2, r_3 and r_4. The state parameters are designed such that $\mathrm{r}_1 < R_1 < r_2 < R_2 < r_3 < R_3 < r_4 < R_4$ in which R_i is the average video data rate up to the ith layer. Hence, the channel throughput will fluctuate about the average rate of each layer. The average throughput of the channel is higher than the base layer but not enough to support the first enhancement layer.

The policy iteration algorithm was used for policy optimization and the window size W was set to 5. Empirically, the algorithm converged to the optimal policy within 10 iterations. Two heuristic policies were compared with the optimized policy (O). The first one (H1) always tries to send data to maximally improve the video quality of the frame which will be displayed in the next slot. The second policy (H2), only transmits the video data in the first two layers because the average throughput is just enough to transmit the first two layers. Each sequence was transmitted over the channel 20 times. The number of lost frames and the average PSNR of the received frames are presented in Table 1. We compare the PSNR and frame loss separately because the degradation of video visual quality also depends on the adopted concealment algorithm. The simulation results shows that the optimized policy can alleviate frame loss while achieving a better video quality. For the received frames, a video quality improvement of 0.4-0.8dB in PSNR is observed.

5. CONCLUSIONS

In this paper, we proposed an MDP formulation for adaptive video scheduling over a wireless channel. Simulation results demonstrate its power in scheduling policy optimization.

6. REFERENCES

[1] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2, Athena Scientific, 3rd edition, 2005.

[2] J. Cabrera, A. Ortega, and J.I. Ronda, “Stochastic rate-control of video coders for wireless channels,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 496 –510, June 2002.

[3] Yan Li, A. Markopoulou, J. Apostolopoulos, and N. Bambos, “Content-aware playout and packet scheduling for video streaming over wireless links,” *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 885 –895, Aug. 2008.

[4] Yu Zhang, Fangwen Fu, and M. van der Schaar, “Online learning and optimization for wireless video transmission,” *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3108 –3124, June 2010.

[5] Qinqing Zhang and S.A. Kassam, “Finite-state markov model for rayleigh fading channels,” *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1688 –1692, nov 1999.

[6] M.F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithm Approach*, The Johns Hopkins University Press, 1981.