

# Stochastic Rate Control for Scalable VBR Video Streaming over Wireless Networks

Guang Ji and Ben Liang

Department of Electrical and Computer Engineering

University of Toronto

Email: {gji,liang}@comm.utoronto.ca

**Abstract**—Video streaming over wireless links is a challenging problem due to both the unreliable, time-varying nature of the wireless channel and the stringent delivery requirements of media traffic. Layered encoded video can be used to improve the system performance by adapting the sending rate for different video frame layers to the varying network and playout situations. In this paper, we study the adaptive control of sending rates for both the base layer and enhancement layer based on feedback information from the wireless receiver. We formulate the problem in a framework of Markov decision processes to minimize a weighted sum of video quality and playout continuity degradation. In order to decrease the computation complexity, we then develop an online greedy algorithm, which only considers the current control time period. Simulation results show that the propose adaptive rate control provides significantly improved video quality and playout smoothness. Furthermore, when rate control is not performed very frequently, the greedy algorithm achieves a video distortion rate nearly matching that of the ideal optimal Dynamic Programming policy.

## I. INTRODUCTION

Media streaming applications have distinctive Quality of Service (QoS) requirements, such as delay sensitiveness and loss tolerance[1]. In addition, the varying wireless environment brings in dramatic fluctuation of network bandwidth [2], which makes streaming even more challenging. A typical wireless video streaming system is presented in Figure 1, where the video sources are stored in the remote *Media Server*. The video sequences are transmitted through the backbone network to the Access Point (AP) and then sent to the users through a wireless network. In order to protect against the influence of the wireless link fluctuation, transmitted video packets are temporarily stored at the user's receiver buffer, which sustains streaming when the network throughput is low.

Advances in video coding technology and standardization are enabling an increasing number of video applications. The Scalable Video Coding (SVC) [3] standard as an extension of H.264/AVC [4] allows efficient, standard-based scalability of temporal, spatial, and quality resolution of a decoded video signal through adaptation of the bit stream. The scalable video encodes each frame into several layers, which includes one Base Layer and several Enhancement Layers. Without the existence of base layer, the corresponding video frame cannot be decoded, which is one of the most important traits for the video streaming system. On the other hand, the enhancement layers are used to supplement the base layer to improve the quality of the video pictures. Therefore, such scalability of a

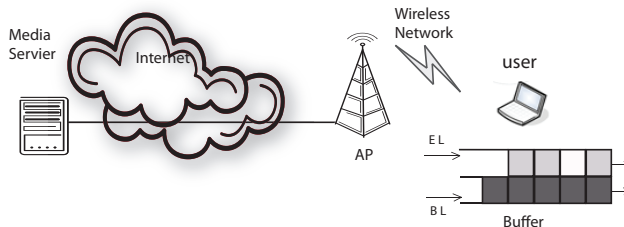


Fig. 1. Video streaming system and the wireless receiver's buffer model.

video stream allows for media bit rate and device capability adaptation without the need of transcoding or re-encoding [5]. We may protect the important part of the scalable media (the base layer) and give less protection to the enhancement layer in order to overcome the most typical deteriorated network situations. Intuitively, when the transmission rate is low, we can drop some of the enhancement layers and transmit more base layer packets in order to guarantee playout continuity, while if high bandwidth is available, we can increase the enhancement layers ratio so as to enhance the video quality. Therefore, proper control of the transmission rates for both of the layers provides a way for the media streaming system to adapt to the vacillating wireless network conditions.

In this paper, we consider the transmission of on-demand scalable *variable-bit-rate* (VBR) video over wireless networks<sup>1</sup>. We initiate an analytical model for the end user's receiver buffer. Based on the knowledge of the current buffered data and the estimation of the future network condition, the receiver can send signals to the sender through a feedback channel to control the sending rate. Furthermore, in order to evaluate the user-perceived streaming media quality, we adopt a cost function combining the base layer loss and the enhancement layer loss. Particularly, the base layer loss models the continuity of video transmission and the enhancement layer loss characterizes the video quality degradation. A Dynamic Programming (DP) based algorithm is applied to achieve the optimal policy for the choices of the sending rates. We also propose a greedy based algorithm that takes less time of execution but at the expenses of performance declination.

The rest of the this paper is organized as follows. We

<sup>1</sup>VBR media provides better quality for the same average bit rate by adapting quantization and compression to the time-varying entropy of the media [6].

discuss the related work in Section II. The system model is presented and the problem is formulated in Section III. We derive the dynamic programming policy in Section IV. Section V presents the greedy policy. Simulation results and further discuss are provided in Section VI. We conclude the paper in Section VII.

## II. RELATED WORK

Several rate control techniques dealing with wired network link transfers of stored video streams are based on the TCP Friendly Rate Control (TFRC) [7] [8], which is designed to be fair to TCP flows and involves lower fluctuation than TCP. For example, Chen and Zakhor in [9] propose the use of multiple TFRC connections as an end-to-end rate control solution for wireless video streaming. However, TFRC cannot distinguish between packet loss due to buffer overflow and that due to bit errors, so these schemes are not suitable for error-prone wireless network streaming systems. A number of efforts have been made to improve TCP, such as combining packet inter-arrival times and relative one-way delay to differentiate the losses, or to use end-to-end statistics to detect congestion.

A large majority of rate control solutions focus on the source-rate control, which is often performed at the frame level or the macroblock level. Cabrera *et al* in [10] introduce a rate control scheme based on *a priori* stochastic models for both the source and the underlying channel, and a solution based on stochastic dynamic programming is proposed. The video considered there is not layered encoded, and the authors mainly explore the rate-distortion model as the performance metric. Atzori *et al* in [11] propose a joint source-rate/channel-code control scheme for streaming VBR-encoded video over a wireless channel. The rate control is performed on a cycle basis which characterizes the “good” and “bad” states of the channel. The scheme is designed to maximize the source rate and considers the user’s buffer by guaranteeing an upper bound of starvation probability. They do not use layered encoded video, and they use starvation probability as the performance metric.

Saparilla and Ross in [12] study the most closely related system setting compared to our work. They prove that for an infinitely-long video, the optimal policy takes on a static form. However, the video considered is constant-bit-rate encoded, and the loss of enhancement layer is simply modeled as a fixed proportion of the base layer loss.

In this work, we focus on a joint video sender-receiver control scheme for transporting scalable VBR encoded video over wireless channels. The main objective is to adapt the sending rates of different video layers based on the combined knowledge of receiver buffer, playback progressing stage, and estimated network condition. To the best of our knowledge, this paper represents the first attempt to analyze stochastic rate control mechanisms for layered encoded VBR media streaming over wireless networks.

## III. SYSTEM MODEL AND PROBLEM FORMULATION

### A. System Model

As shown in Figure 1, the video is stored in a remote server and transmitted to wireless users through an AP. We consider the adaptive transmission of VBR layered encoded video, so the encoding rates of the base and enhancement layers are varying from time to time, denoted as  $C_b(t)$  and  $C_e(t)$  respectively. We consider the video streaming process as time-discrete with equal time slots. Specifically, time is divided into slots  $[t_k, t_{k+1})$ , where  $k \in [0, n - 1]$  and  $n$  is the total number of time slots for the streaming process. Also, we have  $t_0 = 0$ . Suppose before the video starts playing out, an *initial delay* is employed and at time  $t_0$  there are  $\Delta_b$  Base Layer (BL) packets and  $\Delta_e$  Enhancement Layer (EL) packets in the receiver buffer already. Thereafter, multiple video packets are sent to the user in each time slot.

In this paper, we assume the transport channel to provide error-free service, possibly through an ideal error control mechanism, such as coding or ARQ, so that video packet losses may only occur by missing the playout deadline. The feature we are interested in capturing is the time-varying nature of the wireless channel. Let  $R(t)$  be the wireless network transmission rate at time  $t$ , which is also the maximum rate our server streams the video at  $t$ . At any point in time, the sender devotes a certain percentage  $\pi_b$  of the wireless network bandwidth to transmit the BL packets, and  $\pi_e$  for EL packets. Apparently,  $\pi_b + \pi_e = 1$ , but note that  $\pi_b$  and  $\pi_e$  can vary over time.

On the user side, the video receiver consists of a BL playout buffer and an EL playout buffer, which is also shown in Figure 1. The playout buffers are used to temporarily store the incoming video packets. Noting the delay-sensitivity of the video streaming technology, the fluctuations in transmission rate may possibly lead to late packet arrivals, packet losses, significant playback interruption, and video quality degradation. In one time slot, if the required base layer packets do not arrive by the playback schedule time, we cannot decode the corresponding frame and we denote the missing amount of BL as  $L_b$ . Encountering such occurrence of interruptions, we assume that certain concealment technology [13] is incorporated to guarantee continuous playback, but the users’ viewing experience will be degraded.

The base layer provides a basic level of quality and can be decoded independently of the enhancement layer. However, without the corresponding base layer, the received enhancement layer cannot be used for decoding and thereby adds to the loss. The loss amount of the enhancement layer is denoted as  $L_e$ . The layered structure of the scalable video stream and different priorities between the layers provide two sources of enhancement layer loss:  $L_e^1$ , due to the late arrival of EL packets, and  $L_e^2$ , due to missing corresponding BL packets in the same frame.

Clearly, there exists a tradeoff between the video continuity and video picture quality, when we decide the allocation of network resources for BL and EL transmissions. A superior

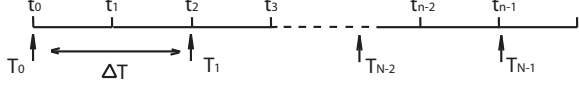


Fig. 2. Illustration of the time slots.

rate control scheme should strike a balance between these two factors to achieve optimized overall user satisfaction. Toward this goal, we introduce a cost function  $C$  as the weighted sum of  $L_b$  and  $L_e$ :

$$C = \omega_b L_b + \omega_e L_e = \omega_b L_b + \omega_e L_e^1 + \omega_e L_e^2. \quad (1)$$

where  $\omega_b$  and  $\omega_e$  are weights decided by the video viewer's preference and both range within  $[0, 1]$ .

Denote  $B_b(t_k)$  and  $B_e(t_k)$  as the buffered BL and EL data in the receiver buffers at  $t_k$ . We also define  $D_b(t_k)$  and  $D_e^1(t_k)$  as the "virtual" buffered data at  $t_k$ , which is calculated as the remained BL/EL data from the previous time slot  $[t_{k-1}, t_k]$  plus the newly incoming data minus the consumed data during  $[t_k, t_{k+1}]$ :

$$D_b(t_k) = B_b(t_{k-1}) + \int_{t_{k-1}}^{t_k} R(t)\pi_b dt - \int_{t_{k-1}}^{t_k} C_b(t)dt,$$

$$D_e^1(t_k) = \underbrace{B_e(t_{k-1})}_{\text{Remained}} + \underbrace{\int_{t_{k-1}}^{t_k} R(t)\pi_e dt}_{\text{New}} - \underbrace{\int_{t_{k-1}}^{t_k} C_e(t)dt}_{\text{Consumed}}.$$

It is obvious that  $D_b(t_k)$  and  $D_e^1(t_k)$  can be positive or negative. If they are positive, it means that there is no video packet loss. Otherwise, the losses are expressed as the absolute value of  $D_b(t_k)$  and  $D_e^1(t_k)$ .

### B. Rate Control Problem Formulation

The receiver is assumed to be aware of the video consumption information. In order to adapt to the varying network transmission rate, we adopt a rate control policy, in which the receiver periodically sends feedback signals to the sender to adjust the value of  $(\pi_b, \pi_e)$  based on the combined knowledge of the buffer level, consumption rates, and the estimated future network condition. The goal is to achieve the minimized total cost defined in (1). We denote  $N$  as the total number of rate control intervals, and  $T_0, T_1, \dots, T_{N-1}$  as the epochs of control executions, where  $T_0 = t_0$ . The time duration between two consecutive control executions is fixed:  $\Delta T = \lceil T/N \rceil$ <sup>2</sup>. Figure 2 illustrates the relationship between the control period and the basic transmission time units.

<sup>2</sup> $\lceil x \rceil$  returns the smallest integer greater than or equal to  $x$ .

We formulate the rate control problem as follows:

$$\text{Minimize } \sum_{k=0}^{n-1} \{\omega_b L_b(t_k) + \omega_e L_e(t_k)\}$$

$$\text{subject to } \pi_b(T_k) + \pi_e(T_k) = 1, \quad 0 \leq k \leq n-1$$

$$\text{where } L_b(t_k) = \begin{cases} |D_b(t_k)|, & \text{if } D_b(t_k) < 0 \\ 0, & \text{otherwise} \end{cases}$$

$$L_e(t_k) = \begin{cases} |D_e^1(t_k)|, & \text{if } D_e^1(t_k) < 0 \text{ and the} \\ & \text{corresponding BL is} \\ & \text{received successfully} \\ \Gamma(|L_b(t_k)|), & \text{if the corresponding} \\ & \text{BL is lost} \\ 0, & \text{if } D_b(t_k) \geq 0 \text{ and} \\ & D_e(t_k) \geq 0 \end{cases}$$

and  $\Gamma(x)$  calculates the traffic amount of enhancement layer corresponding to the base layer  $x$ , which is determined by the video coding scheme. In this optimization problem,  $\pi_b$  and  $\pi_e$  are the control parameters. At each control epoch, we adjust the values of  $\pi_b$  and  $\pi_e$  for the next period of  $\Delta T$ . Denote the control action as  $a = (\pi_b, \pi_e)$ . Then the transmission policy is  $A = (a(0), a(1), \dots, a(N-1))$ , where  $a(t)$  is the control decision made at epoch  $t$ . Our objective is to find the optimal policy  $A$  which minimizes the total loss.

## IV. DYNAMIC PROGRAMMING BASED ALGORITHM

### A. DP Formulation and Algorithm

In this section, we study the problem in a DP framework [14]. We define the state of the system as  $s = (B_b, B_e)$ , where  $B_b$  and  $B_e$  are the BL and EL video packet numbers in the buffer at the current time stamp. Let  $\Lambda$  be the set of control actions  $a$ , which contains all of the possible combinations of  $\pi_b$  and  $\pi_e$ .

We denote  $q_{s,s'}(a; T_k)$  as the transition probability from the state  $s : (B_b(T_k) = i, B_e(T_k) = j)$  at  $T_k$  to the next state  $s' : (B_b(T_{k+1}) = h, B_e(T_{k+1}) = k)$  at  $T_{k+1}$  while taking the action of  $a$ . The value of  $q_{s,s'}(a; T_k)$  not only depends on the states and the action, but on the time  $T_k$  as well, because the video is VBR encoded and the consumption speed of data may not be stationary over time. Therefore, the transition probability matrix is non-homogeneous.

Let  $l_s(a; T_k)$  denote the loss from  $T_k$  to  $T_{k+1}$  with  $s = (B_b(T_k), B_e(T_k))$  at  $T_k$  and using the action  $a$ .  $J_s(T_k)$  is defined as the optimal cost to go, i.e., the minimized total loss starting at  $T_k$  from the current state  $s$  to the end of the video stream assuming that optimal control is used at every control epoch. Then the system becomes a finite-horizon controlled Markov decision process [15] and the optimal control policy  $A$  can be obtained from the following optimality equation:

$$J_s(T_k) = \min_{a \in \Lambda} \{l_s(a; T_k) + \sum_{s'} q_{s,s'}(a; T_k) J_{s'}(T_{k+1})\}, \quad (2)$$

where  $k$  ranges from 0 to  $N-1$ .

The system starts at  $T_0$  with the state  $s_0 = (\Delta_b, \Delta_e)$  and evolves according to (2). The DP computation is based on using backward induction to recursively evaluate the expected costs. We present the algorithm that solves (2) in Algorithm 1.

---

**Algorithm 1** Find the optimal policy:  $A = (a(0), a(1), \dots, a(N-1))$

---

**Require:**  $N, \Lambda, C_b(t), C_e(t)$

```

1:  $k = N - 1$ 
2: for all states  $s$  do
3:    $J_s(T_{N-1}) = \min_{a \in \Lambda} l_s(a; T_{N-1})$ 
4: end for
5: while  $k \geq 0$  do
6:    $k \leftarrow k - 1$ 
7:    $t \leftarrow T_k$ 
8:   for all states  $s$  do
9:      $J_s(T_k) = \min_{a \in \Lambda} \{l_s(a; T_k) + Q\}$ ,
10:    where  $Q = \sum_{s'} q_{s,s'}(a; T_k) J_{s'}(T_{k+1})$ 
11:   end for
12:   set  $a_s(T_k) = \arg_{a \in \Lambda} J_s(T_k)$ 
13: end while

```

---

### B. Use of Policy

Solving the DP formulation involves recursively computing and filling up two tables, in a bottom-up way: one table stores the optimal controls, and another stores the resulting cost  $J_s(T_k)$  for every system state  $s$  at every control epoch. The optimal policy should be computed and stored in a table at the receiver. At each of the control epoch, the receiver collects joint information of the current buffer state, video progress, and network condition. A lookup is next performed in the table to find the optimal control for the current state of the system. Then the receiver sends signals back to the sender providing the updated values of  $\pi_b$  and  $\pi_e$ .

The DP based algorithm is able to find the optimal control policy for the Markov decision process in terms of expectations. The algorithm uses recursive computation that depends on the sizes of the buffer state space and control action set. Depending on the magnitude and granularity of the above parameters, the computation complexity can be quite high. Therefore, in the following section, we propose a greedy algorithm which only focuses on minimizing the total loss during the current control period.

## V. GREEDY ALGORITHM

We propose a greedy algorithm to be executed at each control epoch. The goal is to minimize the total loss in  $[T_k, T_{k+1})$ , denoted as  $L(T_k)$ :

$$L(T_k) = \omega_b L_b(T_k) + \omega_e L_e(T_k) \quad (3)$$

$$= \sum_{t_k=T_k}^{T_k+\Delta T} \{\omega_b L_b(t_k) + \omega_e L_e(t_k)\} \quad (4)$$

To simplify, we denote  $\int_{T_k}^{T_{k+1}} R(t) dt$  as  $\bar{R}(T_k)$ ,  $\int_{T_k}^{T_{k+1}} C_b(t) dt$  as  $\bar{C}_b(T_k)$ , and  $\int_{T_k}^{T_{k+1}} C_e(t) dt$  as  $\bar{C}_e(T_k)$ . Then, we have

$$D_b(T_{k+1}) = \bar{R}(T_k) \pi_b + \underbrace{D_b(T_k) - \bar{C}_b(T_k)}_{N_b(T_k)} \quad (5)$$

$$D_e^1(T_{k+1}) = \bar{R}(T_k) \pi_e + \underbrace{D_e(T_k) - \bar{C}_e(T_k)}_{N_e(T_k)} \quad (6)$$

Here, the signs of  $N_b(T_k)$  and  $N_e(T_k)$  show whether the remaining data in the buffer alone are sufficient for future payout.

The greedy algorithm accommodates different system situations shown as follows. For simplicity, we omit  $T_k$  in  $N_b(T_k)$ ,  $N_e(T_k)$ , and  $\bar{R}(T_k)$ . The algorithm uses a function  $\text{GetPib}(N_b, N_e, \bar{R}, T_k)$ , which returns the value of  $\pi_b$  such that the ratio between  $\bar{R}\pi_b + N_b$  and  $\bar{R}\pi_e + N_e$  is the same as the ratio between the average BL and EL consumption rates.

1) Calculate current  $N_b$  and  $N_e$ .

2) Estimate the total incoming packet amount  $\bar{R}$  based on previous wireless network statistics.

3) Control actions under different situations:

(3.1) If  $N_b \geq 0$  and  $N_e \geq 0$ , the remained packets alone are already enough to sustain the future payout, so  $L(T_k) = 0$ .  
 $\pi_b = \text{GetPib}(N_b, N_e, \bar{R}, T_k)$ .

(3.2) If  $N_b \geq 0$  and  $N_e < 0$ , then  $L_b(T_k) = 0$  and  $L_e(T_k) \geq 0$ .  
 If  $\bar{R} + N_e < 0$ ,  $L_e(T_k) > 0$ , choose  $\pi_e = 1$ ;  
 if  $\bar{R} + N_e \geq 0$ ,  $\pi_b = \text{GetPib}(N_b, N_e, \bar{R}, T_k)$ .

(3.3) If  $N_b < 0$  and  $N_e \geq 0$ , the  $L_e(T_k) \geq 0$  because of EL's dependence on BL, and  $L_b(T_k) \geq 0$ .  
 If  $\bar{R} + N_b < 0$ , choose  $\pi_b = 1$ ;  
 if  $\bar{R} + N_b \geq 0$ ,  $\pi_b = \text{GetPib}(N_b, N_e, \bar{R}, T_k)$ .

(3.4) If  $N_b < 0$  and  $N_e < 0$ , the remained data cannot sustain future payout without new data.

(3.4.a) If  $\bar{R}(t) + N_b + N_e \geq 0$ , then  $L(T_k) = 0$ . It requires  $-\frac{N_b}{\bar{R}(t)} \leq \pi_b \leq 1 + \frac{N_e}{\bar{R}(t)}$ .

(3.4.b) If  $\bar{R}(t) + N_b + N_e < 0$  but  $\bar{R}(t) + N_b > 0$ , then  $L_e(T_k) > 0$  and  $L_b(T_k) < 0$ . We choose  $\pi_b = \frac{-N_b}{\bar{R}(t)}$ .  
 If  $\bar{R}(t) + N_b + N_e < 0$  and  $\bar{R}(t) + N_b < 0$ , then  $L_e(T_k) > 0$  and  $L_b(T_k) > 0$ . We choose  $\pi_b = 1$ .

This greedy algorithm can be easily implemented and consume little computation time. It usually does not provide optimal performance. However, our numerical results in Section VI show that it nearly matches the performance of the ideal optimal DP policy when the channel rate is low.

## VI. SIMULATION RESULTS

In this section, we show results from simulations by which we evaluate the proposed rate-control algorithms and analyze the impact of system parameters, including the wireless channel statistics, the weights for BL and EL losses, and the control interval length.

### A. Simulation Setup

We use MPEG-4 variable-bit-rate video traces provided by [16]. The video sequences are encoded in the Common Image Format (CIF) resolution. The video frames are encoded into two layers: base layer and enhancement layer, which exhibit temporal scalability. Table I summarizes the main parameters of the video trace used in the presented simulation results. The packet size is set to 2100 bytes and the transmission time slot duration is 800 ms.

Parameter	Value
Sequence Length	120 frames
Video Size	$6.1e + 05$ bytes
Format	CIF ( $352 \times 288$ pixels)
Video Run Time	$9.6 \times 10^5$ msec
Mean Bit Rate	$7.5 \times 10^4$ bps
Peak Bit Rate	$1.5 \times 10^5$ bps

TABLE I  
VIDEO PARAMETERS IN SIMULATION

For the wireless channel model, we use  $R$  as the set of possible channel rates and  $Pr$  as the corresponding probability set for different rates. At each time step, the system will randomly choose one rate from  $R$  based on  $Pr$ . We apply 3 sets of wireless channel parameters, in which  $R_1 = [0, 2, 4, 6]$  packets per unit time slot with  $Pr_1 = [0.45, 0.45, 0.05, 0.05]$ ;  $R_2 = [0, 2, 4, 6]$  packets per unit time slot with  $Pr_2 = [0.1, 0.2, 0.5, 0.2]$ ;  $R_3 = [0, 2, 6, 10]$  packets per unit time slot with  $Pr_3 = [0.3, 0.17, 0.3, 0.5]$ . The ratios between the average channel rate and the average video consumption rate are 0.3, 1.0, and 2.0. The discrete action set  $a = [0, 0.1, 0.2, \dots, 0.9, 1]$  is used for  $\pi_b$  and  $\pi_e$ . The initial packets in the buffer  $\Delta_b$  and  $\Delta_e$  are set to  $4.2 \times 10^3$  bytes and  $2.1 \times 10^3$  bytes. We simulate the transmission and playback for the target video sequence in Matlab for over 500 realizations of the random VBR channel and obtain the results by implementing different algorithms.

We define the performance metric as the weighted sum of lost BL and EL packet divided by the weighted sum of total BL and EL packets. We term this ratio the video distortion rate, since given a layered encoding scheme, it is possible to map it to other video distortion metrics such as the peak signal-to-noise ratio.

### B. Constant Algorithms

Before presenting the performances of the proposed DP algorithm and greedy algorithm, we introduce the “non-control” algorithm for the purpose of comparison with the adaptive rate control algorithms. In the constant algorithm the values of  $\pi_b$  and  $\pi_e$  are not changing during the entire streaming process. We set  $\pi_b = 0.4$  and  $\pi_e = 0.6$ , whose ratio is approximately the same as the ratio between the BL and EL average consumption rates.

### C. Impact of Control Duration $\Delta T$ and Channel Rate

The impact of control time length  $\Delta T$  and the channel rate for the DP based algorithm is shown in Figure 3. We

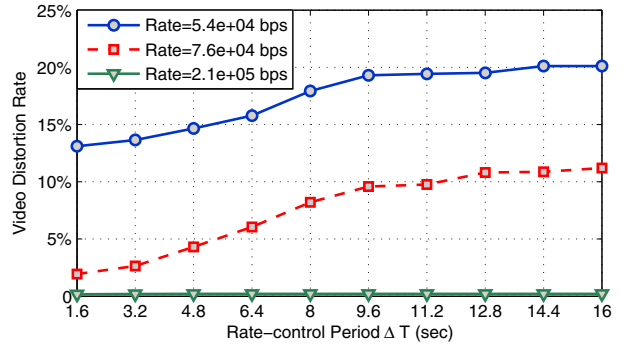


Fig. 3. The impact of control time period on DP algorithm for different network settings.  $\omega_b = 0.8$ .

observe that when the average channel rate is lower than or approximately equal to the average video consumption rate, the distortion rate decreases as  $\Delta T$  becomes smaller. In other words, to receive improved video streaming quality we should increase the frequency of rate control in the DP algorithm. However, with a high channel rate, the video distortion rate is below 1%. This suggests that when there is sufficient network resource, the buffer starvation probability is insignificant, and shortening the control period will not impact much on the overall system performance.

Furthermore, we compare the performance of the DP, greedy, and constant algorithms with various control time durations and channel rates. The results are plotted in Figure 4. In particular, from Figure 4(a), we see that the DP algorithm and the greedy algorithm largely improve the system’s performance compared to the “non-control” algorithm when the network bandwidth is insufficient. Moreover, as  $\Delta T$  increases, the performance of the greedy algorithm almost matches that of DP algorithm. When the average channel rate and the video consumption rate is equal as shown in Figure 4(b), the DP and greedy algorithms still significantly outperform the constant algorithm, but there is a substantial gap between DP and greedy.

### D. Impact of the Loss Weights

We compare the algorithms with different  $(\omega_b, \omega_e)$  in Figure 5 and Figure 6. Since in practice the weight of BL losses is generally larger than that of the EL losses, we set the values of  $\omega_b$  from 0.6 to 1. We observe that as  $\omega_b$  increases, the video distortion rates of the DP and greedy algorithms are decreasing accordingly, which implies that the proposed algorithms are favoring the BL packets. The larger weight we put on the BL losses, the better system performance we will achieve by using the DP and greedy algorithms.

## VII. CONCLUSION

We have studied the problem of stochastic rate control policies for VBR layer encoded video streaming over wireless networks, by formulating the system as a Markov decision process. We combine the video receiver’s information of current buffer state, video consumption rate, and the estimation



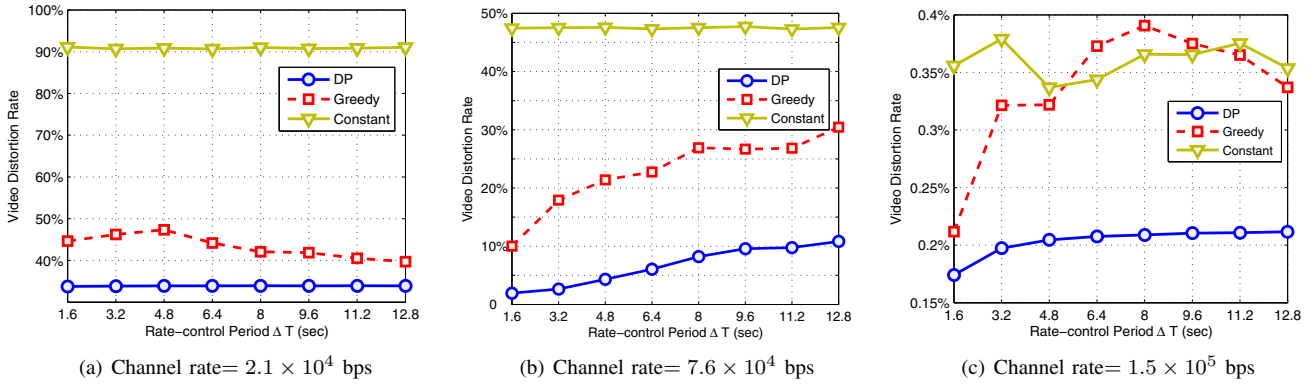


Fig. 4. Comparison of algorithms with different control time durations.  $\omega_b = 0.8$ .

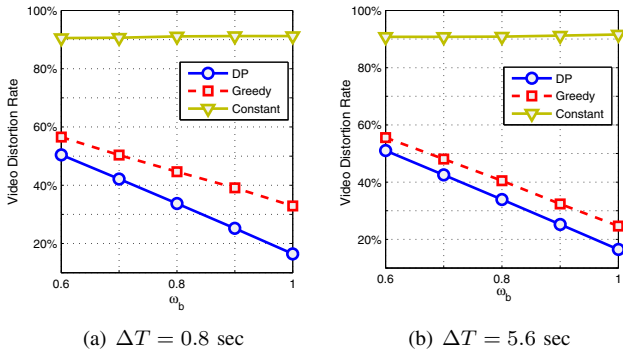


Fig. 5. Comparison algorithms with different  $\omega_b$ . The average channel rate is  $2.1 \times 10^4$  bps.

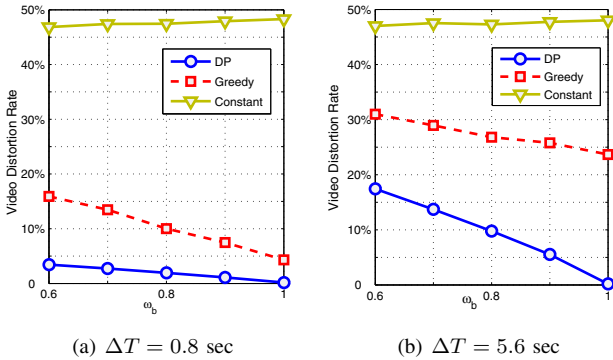


Fig. 6. Comparison algorithms with different  $\omega_b$ . The average channel rate is  $7.6 \times 10^4$  bps.

of wireless network condition to obtain the optimal control policy, from which the sender can adjust the proportion of network resource for sending base layer and enhancement layer packets based on the feedback signals. Both a DP based algorithm and a greedy algorithm are studied. Simulation results show that the DP and the greedy algorithms generally provide significantly improved system performance compared to the “non-control” algorithm. However, we also observe a substantial performance gap between the DP and greedy algorithms when the channel rate is close to or above the

video data rate, which suggests an important open problem for further research.

#### ACKNOWLEDGMENT

This research was funded in part by Bell Canada through its Bell University Laboratories program.

#### REFERENCES

- [1] J. Chakareski and P. Frossard, “Adaptive systems for improved media streaming experience,” *IEEE Communications Magazine*, vol. 45, no. 1, pp. 77–83, Jan. 2007.
- [2] M. Etoh and T. Yoshimura, “Advances in wireless video delivery,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 111–122, Jan. 2005.
- [3] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of h.264/avc,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
- [4] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the h.264/avc video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [5] T. Schierl, T. Stockhammer, and T. Wiegand, “Mobile video transmission using scalable video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1204–1217, Sep. 2007.
- [6] T. Lakshman, A. Ortega, and A. Reibman, “Variable bit rate (VBR) video: Tradeoffs and potentials,” *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–973, May 1998.
- [7] S. Flyod, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” *Proc. ACM SIGCOMM*, pp. 43–56, Aug. 2000.
- [8] W. Tan and A. Zakhor, “Real-time internet video using error resilient scalable compression and tcp-friendly transport protocol,” *IEEE Trans. Multimedia*, vol. 1, no. 2, pp. 172–186, Jun. 1999.
- [9] M. Chen and A. Zakhor, “Rate control for streaming over wireless,” in *Proc. IEEE INFOCOM*, Apr. 2004.
- [10] J. Cabrera, A. Ortega, and J. Ronda, “Stochastic rate-control of video coders for wireless channels,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 496–510, Jun. 2002.
- [11] L. Atzori, M. Krunz, and M. Hassan, “Cycle-based rate control for one-way and interactive video communications over wireless channels,” *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 176–184, Jan. 2007.
- [12] D. Saporilla and K. Ross, “Optimal streaming of layered video,” in *Proc. IEEE INFOCOM*, Mar. 2000.
- [13] Y. Wang and Q. Zhu, “Error control and concealment for video communication: A review,” *Proceedings of the IEEE*, vol. 86, pp. 974–997, May 1998.
- [14] D. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2000, vol. 1 & 2.
- [15] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: John Wiley & Sons, 1994.
- [16] P. Seeling, F. H. P. Fitzek, and M. Reisslein, *Video Traces for Network Performance Evaluation*. Springer, Nov. 2006.