Analysis and Design of a Proportional-Integral Rate Controller for Streaming Videos

Yingsong Huang and Shiwen Mao
Department of Electrical and Computer Engineering, Auburn University, Auburn, AL
Email: yzh0002@auburn.edu, smao@ieee.org

***Abstract*—In this paper, we study the problem of rate control for streaming videos by jointly considering encoder rate control and network congestion control. We adopt a control-theoretic approach that models video streaming as a feedback control system. Based on a properly chosen operating point, the model is linearized and a proportional-integral (PI) controller is designed to stabilize the streaming video quality. We derive the guidelines for choosing parameters for the proposed PI rate controller and prove its stability properties. We also show that the PI rate controller is highly robust to fluctuations in the bottleneck link capacity. Our simulation results verify the accuracy of the analysis and demonstrate the efficacy of the proposed control-theoretic approach.**

I. Introduction

Although video service is widely available in the Internet nowadays, provisioning of Quality of Service (QoS) guarantees for streaming videos remains to be an open problem. It is not atypical to have stalled video display during a live session, or to have frequent damaged frames. For streaming videos, congestion is one of the major problems that need to be addressed. During congestion, a large amount of video packets could be dropped at the bottleneck link due to buffer overflow. In addition, the high delay experienced at congested links makes many packets arrive later than their decoding deadlines. Such overdue video packets may be discarded at the receiver. For high quality video delivery, it is imperative to make video sessions responsive to congestion. Such responsiveness also makes streaming video "friendly" to other data or multimedia sessions that share the same bottleneck link.

It is well-known that the additive-increase-multiplicative-decrease (AIMD) algorithm adopted in TCP is not suitable for video streaming (i.e., not video downloading). AIMD is effective for probing and consuming all the available bandwidth for elastic data, but may be an "overkill" for streaming videos with bounded rates and limited playout buffers. That is, video data is usually generated at, and is consumed at a constant frame rate. An overly large bandwidth may not be much useful when there is no data to send, while an overly small bandwidth will certainly cause overdue frames and stalled display. The everlasting fluctuations in the AIMD rate cause fluctuations in received video quality, which are annoying for viewers. For streaming videos, it is highly desirable to have a smooth and elegantly degraded quality as network status evolves over time, thus yielding high subjective video quality.

Video rate control has been studied by many researchers (e.g., see [1]–[7] and references therein). Several rate control algorithms have been incorporated into video coding standards. Many existing work concerns with how to allocate a given bit rate budget to the frames or Macroblocks (MB) at the video coder, while a given bit rate can be met by adjusting the quantization parameter or by skipping certain frames.

In this paper, we revisit this important problem by jointly considering the encoder rate control and network congestion control with a control-theoretic approach. We consider a bottleneck link shared by multiple video sessions. The Random Early Detection (RED) mechanism is adopted at the routers to drop packets under incipient congestion [8]. A receiver may measure the received video quality as feedback to the corresponding server. When a quality degradation is detected, what should the video server do? The overall video distortion consists of the distortion due to quantization error at the encoder and that due to lost packets [9], [10]. If the coder increases its rate, the distortion is reduced due to smaller quantization error. However, the distortion may also be increased due to intensified congestion. On the other hand, if the coder decreases its rate, although certain reduction in distortion may be obtained due to mitigated congestion, the lower encoding rate will cause a larger encoder distortion. There is certainly a need for theoretical modeling and analysis for determining the optimal strategy for the coder to respond to congestion and to deliver smooth received video quality.

Motivated by the seminal work on applying feedback control theory to TCP and active queue management (AQM) [11], we propose a control theoretic-approach to rate control for streaming videos. We model the overall streaming system, including the coder, the bottleneck link where AQM is enforced, and feedback from receiver as a *feedback control system*. Based on a properly chosen operating point, we can linearize the complex system and apply classic feedback control theory to design effective rate controllers. In our prior work [12], we designed a proportional (P) controller to stabilize the video quality. In this paper, we extend our prior work by investigating the case of proportional-integral (PI) controllers [13]. PI controllers are widely used in feedback systems, which stabilize a plant using a weighted sum of the difference between the output and desired set-point, i.e., error, and its integral. The advantage of PI controller is that its integral part may drive the steady-state error to zero, which enhances the steady-state performance of the system. This is a perfect match for the objective of delivering streaming videos with smooth, or stabilized qualities. We derive the guidelines for setting

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

parameters for the proposed PI rate controller and prove its stability properties. We also show that the PI rate controller is robust to fluctuations in the bottleneck link capacity. We verify our analysis and the PI controller design by simulations with a SystemC-based simulator [14]. Our simulation results validate the accuracy of our analysis, and demonstrate the efficacy of the proposed control-theoretic approach.

The remainder of this paper is organized as follows. We first present a feedback control system model for streaming videos in Section II. We then formulate the rate control problem and derive a PI controller in Section III. We present our simulation results in Section IV. Section V concludes this paper.

II. System Model

In this section, we present a dynamic model for the video rate control problem, while considering both network congestion and video rate-distortion characteristics. The model is then simplified and linearized for the design of stabilizing rate controllers.

A. Network Congestion Model

We assume a bottleneck link where N video sessions share the link bandwidth. The bottleneck could be a router link within the wired network, or the last-hop wireless link if end users are mobile. Video packets may be dropped both at the bottleneck link due to congestion and at the wireless link due to interference or collision. The bottleneck router is assumed to incorporate an AQM mechanism, i.e., RED [8], for the purpose of congestion control.

In RED, packets are dropped randomly under incipient congestion with a probability that is a function of the queue length. If the buffer occupancy is less than a minimum buffer threshold q_{min}, all incoming packets will be accepted. As the queue length grows, the drop probability grows as well. When the buffer reaches the maximum buffer threshold q_{max}, the drop probability reaches the maximum drop probability p_{max}. Beyond q_{max}, all incoming packets will be dropped with probability one. Without loss of generality, we consider the *proportional (P) scheme* among the RED variations, where the drop probability $p(t)$ is proportional to the queue length $q(t)$ [11]. Due to the bottleneck router, we have $q(t) > q_{min}$ with a probability close to one. We then focus our analysis on the following linear region:

$$p(t) = K_r \cdot [q(t) - q_{min}], \text{ for } q_{min} \le q(t) \le q_{max}, \quad (1)$$

where K_r is the RED parameter.

We also assume that the aggregate loss rate on all the non-bottleneck links is $p_w \in [0, 1]$. Usually p_w and $p(t)$ need to be small in order to provide an acceptable video quality (i.e., guaranteed via an admission control and QoS routing mechanism). An estimate of the overall packet loss rate is

$$p_{tot} = p(t) + p_w - p(t) \cdot p_w. \quad (2)$$

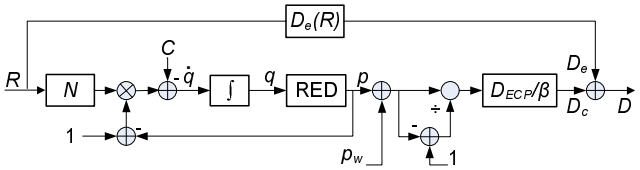

Fig. 1. Block diagram of the overall feedback system.

At the bottleneck link, assume the capacity reserved for video traffic is C.[1] According to Lindley's equation and for the linear region that is considered, the buffer dynamics can be written as

$$\frac{dq(t)}{dt} = N \cdot R(t) \cdot [1 - p(t)] - C, \quad (3)$$

where $R(t)$ is the video bit rate and N is the *load factor*.

B. Video Performance Model

We consider the distortion of received video, which largely consists of *encoder distortion* D_e caused by quantization errors and *channel distortion* D_c due to network packet losses [9], [10], [17]. That is, the overall distortion is $D = D_e + D_c$. We adopt the empirical model proposed in [9] in this paper without loss of generality, where the encoder distortion, measured as mean square error (MSE), can be evaluated as:

$$D_e = I_0 + \frac{\theta}{R - W_0}, \quad (4)$$

where I_0, θ and W_0 are constants for a specific video codec and video sequence, and R is the coding rate. We verified this model using the H.264 codec with options in the Baseline profile, using various test sequences. We find the model fits well with measurements for all the video sequences studied.

We adopt the general model presented in [10] for the channel distortion part, which can be expressed as

$$D_c = \frac{p_{tot}}{\beta \cdot (1 - p_{tot})} D_{ECP}, \quad (5)$$

where D_{ECP} is the *average concealment distortion* and β is the *intra rate* at the coder. D_c is shown to be an *upper bound* for the average channel distortion [10]. In the following, we use (5) to approximate the average channel distortion of reconstructed video frames.

C. Overall System Model and Linearization

Based on the above network and video models, we can derive the model for the overall video streaming system as

$$\begin{cases} \dot{q}(t) = N \cdot R(t) \cdot [1 - p(t)] - C \\ D(t) = I_0 + \frac{\theta}{R(t) - W_0} + \frac{p_{tot}(t)}{\beta \cdot [1 - p_{tot}(t)]} D_{ECP} \\ p_{tot} = p(t) + p_w - p(t) \cdot p_w \\ p(t) = K_r \cdot [q(t) - q_{min}], \text{ for } q(t) \in [q_{min}, q_{max}], \end{cases} \quad (6)$$

where $\dot{x}$ denotes the *time-derivative* of x, i.e., dx/dt.

[1] A router can adopt the generalize processor sharing (GPS) scheduler to guarantee the reserved bandwidth for video sessions [15]. Otherwise, the receiver may estimate C based on end-to-end measurements [16].

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

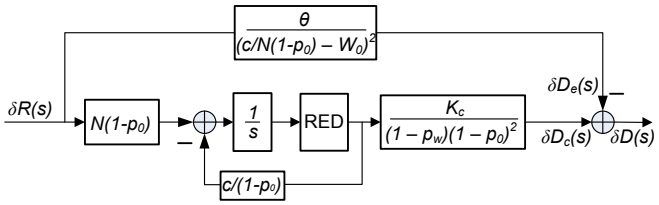

Fig. 2. Block diagram of the linearized model.

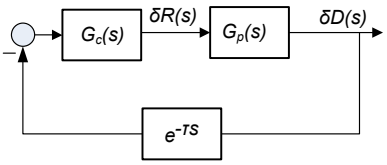

Fig. 3. Redraw the block diagram of the linearized feedback control system.

We plot the block diagram of the nonlinear system in Fig. 1. We next linearize it by focusing on the dynamics around a properly chosen operating point. Consider $q(t)$ as the state, $R(t)$ the input, and $D(t)$ the output of the feedback control system. The operating point p_0 is chosen to satisfy $\dot{q}(t) = 0$. Let the corresponding video distortion be D_0 and the corresponding rate be R_0. We have

$$\begin{cases} R_0 = \frac{C}{N(1-p_0)} \\ D_0 = I_0 + \frac{\theta}{R_0 - W_0} + \frac{K_c \cdot (p_0 + p_w - p_0 \cdot p_w)}{(1-p_0-p_w+p_0 \cdot p_w)}, \end{cases} \tag{7}$$

where $K_c = D_{ECP}/\beta$. We next linearize (6) around the operating point p_0 as

$$\begin{cases} \delta(\dot{q(t)}) = N(1-p_0)\delta(R(t)) - \frac{C}{1-p_0}\delta(p(t)) \\ \delta(D(t)) = \frac{-\theta}{(R_0-W_0)^2}\delta(R(t)) + \frac{K_c}{(1-p_w)(1-p_0)^2}\delta(p(t)), \end{cases} \tag{8}$$

where $\delta(x) = x - x_0$. Applying Laplace transform to the differential equations, we plot the linearized system block diagram in Fig. 2. With some manipulation of the block-diagram, we obtain the transfer function of the plant as

$$G_p(s) = \frac{N \cdot K_r \cdot K_c}{(1-p_w)(1-p_0)(s + K_r C/(1-p_0))} - K_e. \tag{9}$$

In Fig. 2, the block diagram consists of two forward paths. Let $K_e = \frac{\theta}{(R_0-W_0)^2}$ be the forward loop gain. K_e in the upper path represents the encoder distortion part, while the lower path represents the channel distortion part. Note that as bit rate increases, encoder distortion D_e decreases. On the other hand, the increased bit rate usually leads to network congestion, resulting more packet drops at the bottleneck router and an increased D_c. Accordingly, the total distortion $D = D_e + D_c$ will not be monotonically increased or decreased by simply adjusting the video bit rate. The net effect depends on the actual amount of increase/decrease in the two components.

Generally, a video rate-distortion curve becomes quite flat when the rate is moderate or large. To achieve an acceptable distortion, the system usually needs to work within such range, where a moderate variation of the bit rate does not greatly affect the encoder distortion. On the other hand, the increased packet loss rate at the bottleneck router caused by the increased rate will cause relative larger distortion. Based on this observation, we further simplify the system model by considering the lower path in Fig. 2, while omitting the upper path. The simplified system transfer function is

$$G_p(s) = \frac{N \cdot K_r \cdot \alpha}{s + K_r \cdot C/(1-p_0)}, \tag{10}$$

where $\alpha = K_c/[(1-p_w)(1-p_0)]$. The pole of (10) is $s = -K_r \cdot C/(1-p_0) < 0$, i.e., it is located in the left phase plane. The system is asymptotically stable.

III. Design and Analysis of the PI Controller

In this section, we design a PI controller for the linearized feedback control system (10). We then analyze its stability performance and derive guidelines on parameter setting.

A. Performance Objectives

As in any feedback control system design, we first provide our performance objectives. As discussed, packet loss is a main cause of video quality degradation, especially in realtime video streaming with tight delay requirements, where retransmission is not affordable. Variations in packet loss rate will cause large variations of received video quality, which will be annoying to viewers. The rate controller should maintain the packet loss probability around the operating point p_0 to achieve a consistent distortion level.

The network condition, e.g., the reserved capacity for video traffic, C, may change over time. The rate controller should be adaptive and robust to such disturbances. It should be able to stabilize the system under disturbances, and also be able to drive the system back to the operating point when the disturbance disappears.

B. A Proportional-Integral Rate Controller

We redraw the block diagram in Fig. 3, where $G_p(s)$ is given in (10). The $e^{-\tau s}$ term in Fig. 3 is the Laplace transform of the round trip delay τ. From the control prospective, $G_c(s)$ is the *controller* to be designed, while the combination of the other blocks is the *plant* to be controlled. A classic PI controller has the following transfer function:

$$G_c^{PI}(s) = \frac{K_{pi}}{s} \cdot \left(s + \frac{1}{T_{pi}}\right). \tag{11}$$

It is well-known that the integral part of a PI controller can greatly reduce the steady-state error [13]. The open-loop transfer function for the system with the PI controller is

$$L(s) = \frac{K_{pi}}{s}\left(s + \frac{1}{T_{pi}}\right)\frac{N \cdot K_r \cdot \alpha}{s + K_r \cdot C/(1-p_0)}e^{-s\tau}. \tag{12}$$

We design the PI controller as follows. We first choose the PI controller's break point to coincide with the corner frequency of the system dynamic, i.e.,

$$\frac{1}{T_{pi}} = \frac{K_r \cdot C}{1-p_0}. \tag{13}$$

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

We then choose the crossover frequency as

$$\omega_c = \kappa/\tau, \tag{14}$$

where κ is a constant empirically chosen to provide sufficient phase margin. A larger value for κ will provide faster system response, but also produce a lower stability margin. In practice, we may choose a proper value according to the network situation to balance between responsiveness and stability.

We have the following propositions for the PI controller.

Proposition 1: The PI controller (11) can stabilize the feedback control system in Fig. 3 if its parameters satisfy:

$$\omega_c = \frac{\kappa}{\tau} \quad \text{and} \quad K_{pi} = \frac{\kappa}{N \cdot K_r \cdot \alpha \cdot \tau}. \tag{15}$$

Proof: Consider the amplitude of the open-loop transfer function $L(s)$ at the crossover frequency. We have

$$|L(j\omega_c)| = \left| \frac{K_{pi}}{j\omega_c} \left(j\omega_c + \frac{1}{T_{pi}} \right) \frac{NK_r\alpha}{j\omega_c + K_rC/(1-p_0)} e^{-s\tau} \right|.$$

Since $|L(j\omega_c)| = 1$, we can solve for K_{pi} to have $K_{pi} = \kappa/(NK_r\alpha\tau)$. From Proposition 1, the unit-gain crossover frequency is limited by ω_c. We have $\angle L(j\omega_c) = -\frac{\pi}{2} - \tau\omega_c = -\frac{\pi}{2} - \kappa > -\pi$, for $\kappa < \pi/2$. Applying the Nyquist stability criterion [13], the feedback system in Fig. 3 is stable. ■

Proposition 2: With a PI controller satisfying Proposition 1, the gain margin (GM) and phase margin (PM) of the feedback control system in Fig. 3 satisfy: $PM = \frac{\pi}{2} - \kappa$ and $GM = \pi/(2\kappa)$.

Proof: The phase at the crossover frequency ω_c is $\angle L(j\omega_c) = -\frac{\pi}{2} - \tau\omega_c = -\frac{\pi}{2} - \kappa$. Therefore the phase margin is found to be $PM = \angle L(j\omega_c) - (-\pi) = \frac{\pi}{2} - \kappa$.

Let ω_g be the frequency at which the phase reaches $-180°$. Since $-\frac{\pi}{2} + \angle(e^{-j\omega_g\tau}) = -\pi$, we have $\angle(e^{-j\omega_g\tau}) = -\pi/2$. Therefore, we obtain that $\omega_g = \pi/(2\tau)$. According to Proposition 1, we have $|L(j\omega_g)| = |L[j\pi/(2\tau)]| = (2\tau/\pi) \cdot (\kappa/\tau)$. Therefore we have $GM = 1/|L(j\omega_g)| = \pi/(2\kappa)$. ■

As a numerical example, consider a system with parameters $C = 1.5 \times 10^4$ kb/s, $N = 30$, $K_r = 5.862 \times 10^{-5}$, $p_w = 0.01$, and $\tau = 0.200$ s. The video sequence is *Foreman* coded using an H.264 codec at 15 f/s with parameter $\beta = 0.03$ [10]. We choose $p_0 = 0.006$ and $\kappa = 0.164$, and draw the Bode plot of $L(j\omega)$ in Fig. 4. It can be verified that $GM \approx 20$ dB and $PM \approx 80°$ from Fig. 4. Therefore the feedback control system is stable with sufficient stability margins.

Next, we analyze the performance of the PI controller under disturbances in the bottleneck capacity C. We show that for a PI controller satisfying Proposition 1, the variation of the bottleneck link capacity C does not affect the stability of the system, if the new capacity is within a certain range.

Let $C' = \gamma \cdot C$. Then we have the new system transfer function as $G'_p(s) = \frac{N \cdot K_r \cdot \alpha}{s + K_r \cdot C'/(1-p_0)}$. For fixed ω, $|G_p(j\omega)|$ monotonically decreases when C is increased. Likewise, $\angle G_p(j\omega)$ monotonically increases when C is increased.

We consider two cases for γ. If $\gamma > 1$, the bottleneck link bandwidth is increased, i.e., $C' > C$. It can be shown that

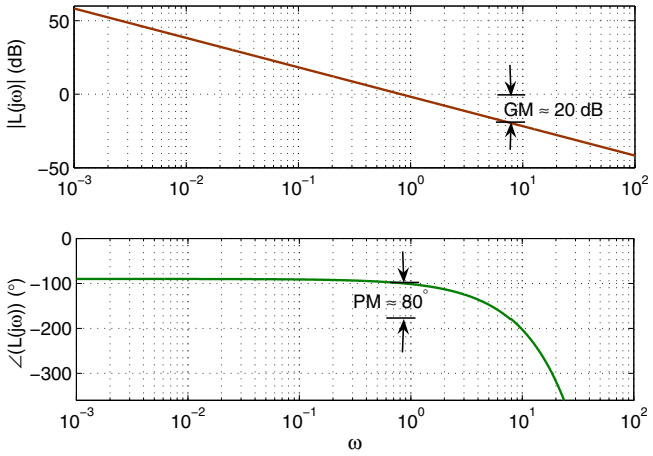

Fig. 4. Bode plot for the example with a PI controller satisfying Proposition 1.

$|G'_p(j\omega)| \leq |G_p(j\omega)|$ and $\angle G'_p(j\omega) \geq \angle G_p(j\omega)$. Applying the Nyquist stability criterion, if the PI controller $G_c^{PI}(s)$ can stabilize $G_p(j\omega)$, it can also stabilize the new plant $G'_p(j\omega)$.

If $0 < \gamma < 1$, the bottleneck link bandwidth is decreased, i.e., $C' < C$. Since $|L'(j\omega_c)| = 1$, we have $\omega_c \leq \kappa/(\tau\gamma)$.We have the following for the PM:

$$\begin{aligned} PM &= \angle L'(j\omega_c) - (-\pi) > \left| \angle L'(j\frac{\kappa}{\tau\gamma}) \right| - (-\pi) \\ &= \frac{\pi}{2} + \arctan\left\{ \left(\frac{1-p_0}{K_rC} \right) \left(\frac{\kappa}{\tau\gamma} \right) \right\} - \\ &\qquad \arctan\left\{ \left(\frac{1-p_0}{K_r\gamma C} \right) \left(\frac{\kappa}{\tau\gamma} \right) \right\} - \frac{\kappa}{\gamma}. \end{aligned}$$

It can be shown that the minimum PM is

$$\text{PM}_{min} = \frac{\pi}{2} + \arctan\left(\sqrt{\gamma}\right) - \arctan\left(\frac{1}{\sqrt{\gamma}} \right) - \frac{\kappa}{\gamma}.$$

For a given κ, we find the range of coefficient γ by solving $\text{PM}_{min} > 0$. For example, if $\kappa = 0.1$, we have $\gamma > 0.15$.

Next, consider the GM at ω_g when the phase reaches $-180°$. Letting

$$-\frac{\pi}{2} + \arctan\left\{ \frac{1-p_0}{K_rC}\omega_g \right\} - \arctan\left\{ \frac{1-p_0}{K_r\gamma C}\omega_g \right\} - \omega_g\tau = -\pi,$$

we follow the derivation as mentioned above for the PM to have $\omega_g \geq (0.7390/\tau)$ for $\gamma = 0.15$, when $\kappa = 0.1$. We then have $GM = \frac{1}{|L(j\omega_g)|} \geq \frac{1}{|L(j0.7390/\tau)|} \geq 1.1 \approx 0.89$ dB, which is positive.

Applying the Nyquist stability criterion, if the PI controller $G_c^{PI}(s)$ can stabilize $L(j\omega)$, it can also stabilize $L'(j\omega)$ under the constraint determined by κ.

Remark 1: Recall that we made the assumption in the previous section that the videos are coded at moderate or large rates. For coded videos, channel distortion is the dominant impairment factor in this region. When C is decreased, the video rates are also decreased, but we assume that this assumption is not violated (e.g., guaranteed by GPS [15]).

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

In the video streaming system, the video servers independently respond to a decrement in C: each makes a decision on adapting its rate, while the overall effect is to drive the RED drop rate at the bottleneck link back to p_0, or, the bottleneck queue length back to q_0. Since there is no exchange of information among the servers, it is possible that the video sessions reach different rates after a new convergence, although the overall queue length is kept around q_0. Such discrepancy in video rates resulting in small differences in the overall PSNRs after convergence, due to different encoder distortions. A threshold may be set for each video stream to prevent very small rates. For example, if the design objective is to stabilize video quality when $C' > \gamma C$, for $0 < \gamma < 1$, we can assume a minimum rate of C'/N for each video session. If this minimum rate still cannot guarantee stability, the video streams will be terminated, since the network condition is too poor to guarantee an acceptable video quality.

IV. Simulation Results

We evaluate the performance of the PI controller using a customized simulator written in C++ on the SystemC platform, which consists of a library of routines and macros for simulation of concurrent processes [14]. In the following simulations, we consider both the case of homogeneous videos (with 30 sessions) and the case of heterogeneous videos (with two classes). The H.264 reference software encoder (JM) in the Baseline profile is used to generate the test sequence. We use two video sequences in the Quarter Common Intermediate Format (QCIF): the high motion sequence *Football* and the medium motion sequence *Foreman*. Both videos are encoded at 15 frames/s. The intra rate is $\beta = 0.11$ for *Football* and $\beta = 0.03$ for *Foreman*. The link bandwidth of the bottleneck router is set to $C = 15$ Mb/s. The bottleneck router buffer size is 2000 packets. The RED parameter is set to $K_r = 5.862 \times 10^{-5}$ [11]. The packet loss rate at the non-bottleneck links is $p_w = 0.01$.

1) Homogeneous Videos: We simulate a video streaming system where 30 sessions share the bottleneck link. The *Foreman* sequence is used in these simulations. We choose an operating point with RED drop probability $p_0 = 0.006$, i.e., around the fixed loss rate p_w. The round trip delay is $\tau = 200$ ms. The corresponding PI controller for the feedback control system is found to be $G_c^{PI}(s) = \frac{0.0565}{s}(s + 0.885)$.

In Fig. 5, we plot the simulation results for the homogeneous video system. We find the overall packet loss rate accurately converge to the steady state value after a few seconds, i.e., to the neighborhood of $p_{tot} = 0.016$. After convergence, the loss rate fluctuates around the steady state value, with small deviations from the steady state value. This is because the integral part of the controller is effective in suppressing steady-state errors. As will be seen in Figs. 6 and 7, the PSNRs of the videos can also be stabilized with the proposed PI controller.

We next examine the performance of the PI controller under network dynamics. In this simulation, we assume that the bottleneck link capacity allocated to the video streams varies due to some high priority user traffic dynamics. The round trip time is still $\tau = 200$ ms. Since the video sessions may be set up when either the high priority users are active or idle, the available capacity could be under C during the streaming phase. Without loss of generality, we assume the available capacity changes every 120 s during the simulation. When the high priority users are active, we assume that the bottleneck link bandwidth for video sessions is uniformly distributed in $[50\%, 99\%]$ of the original value. After 120 s, the high priority users leave the network and the bottleneck link capacity for the video sessions is assumed to return to the original value.

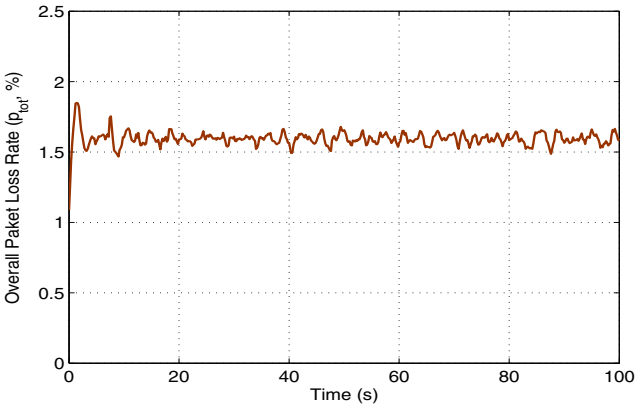

Fig. 5. Simulation results for 30 homogeneous video sessions streaming the *Foreman* sequence.

The PSNR experienced at a client is plotted in Fig. 6. The system is stabilized around the target PSNR $31dB$, even though the available capacity is reduced to 70.6%. This indicates that the received video quality is highly smooth and stable. For many streaming applications, such smooth video quality is more desirable than a high average PSNR value, since for human viewers large variations in the video quality is very annoying, even though the average quality may be better. It can also be observed that there is a deep drop in PSNR right after the capacity decrease. This is mainly due to fact that the capacity changes are instantaneous as compare to the round trip delay. The router buffer will be quickly congested after a decrease in C, since it takes at least one round trip delay for the sources to detect the congestion and respond to it. After the controller receives feedbacks, the controller will keep on driving the packet loss rate back to the operating point by adjusting the sending rate. In a real network deployment, the change in bottleneck link bandwidth may not be instantaneous, giving the servers to respond to the changes. We expect the spikes will be gone or be much smaller in such cases. On the other hand, there is no PSNR fade when C is increased. A larger C causes buffer underflow, and the servers will increase the video rates to reach the operating point p_0, resulting in an temporary increase in PSNR, as shown in Fig. 6.

2) Heterogeneous Videos: We next consider the more challenging case of heterogeneous videos. In the simulations, there are two classes of video streams, one consists of 10 sessions streaming the *Football* sequence, the other consists of 10 sessions streaming the *Foreman* sequence. Each video

978-1-4244-4148-8/09/$25.00 ©2009

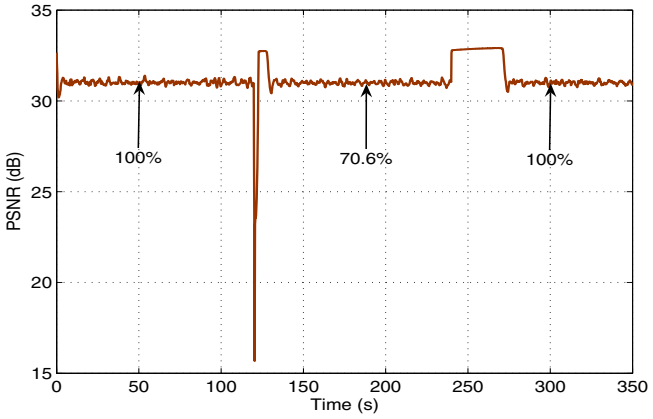

Fig. 6. PSNR of a reconstructed *Foreman* sequence when the bottleneck capacity C changes in the homogeneous video simulation.

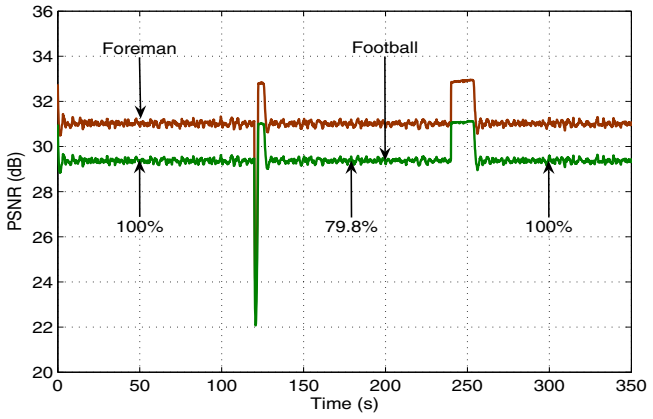

Fig. 7. PSNR of the reconstructed *Football* and *Foreman* sequences when bottleneck capacity C changes in the heterogeneous video simulation.

is encoded as discussed earlier in this section. We choose an operating point with RED drop probability $p_0 = 0.006$. The round trip delay is 100 ms. We derive the controller for the *Foreman* class as $G^P_{c,1}(s) = \frac{0.169}{s}(s+0.295)$ and the controller for the *Football* class as $G^P_{c,2}(s) = \frac{0.085}{s}(s+0.59)$.

Similar to the homogeneous video case, we add a disturbance to the bottleneck link capacity allocated for video every 120 s. The PNSR for the two videos are plotted in Fig. 7. As in the homogeneous video case, both the *Foreman* video streams and the *Football* video streams are stabilized around the target PSNRs $31dB$ and $29.4dB$, respectively, although the bottleneck capacity is reduced to 79.8%. We observe that the qualities of both video classes are smooth and stable, a desire result for the subjective video quality. Similar to the simulation with homogeneous videos, there are spikes right after the decrease in C. The decrease in PSNR are caused by the instantaneous capacity decreases. After several feedbacks, the servers will detect the change and respond by reducing video rates, and finally the PSNR are stabilized.

V. Conclusions

In this paper, we developed a feedback control system model for video streaming systems, which takes into account the interactions among video rate control, RED active queue management, and received video quality. We designed a PI rate controller for streaming videos to stabilize the received video quality. We derived the stability properties of the proposed PI rate controller and used a SystemC-based simulator for performance evaluation. We found the proposed approach highly effective in rate and congestion control for both homogeneous and heterogeneous video systems.

Acknowledgment

This work is supported in part by the National Science Foundation under Grant ECCS-0802113 and through the Wireless Internet Center for Advanced Technology at Auburn University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Foundation.

References

[1] I. Ahmad and J. Luo, "On using game theory to optimize the rate control in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 209–219, Feb. 2006.

[2] S. Aramvith, I. Pao, and M. Sun, "A rate-control scheme for video transport over wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 569–580, May 2001.

[3] W. Ding, "Joint encoder and channel rate control of VBR video over ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 266–278, Apr. 1997.

[4] Z. He and S. Mitra, "A linear source model and a unified rate control algorithm for DCT video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 970–982, Nov. 2002.

[5] M. Jiang and N. Ling, "On Lagrange multiplier and quantizer adjustment for h.264 frame-level video rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 663–669, May 2006.

[6] Y. Liu, Z. Li, and Y. Soh, "A novel rate control scheme for low delay video communication of H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 1, pp. 68–78, Jan. 2007.

[7] Y. Yang and S. Hemami, "Rate control for VBR video over ATM: Simplification and implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 9, pp. 1045–1058, Sept. 2001.

[8] B. Braden and et al., "Recommendations on queue management and congestion avoidance in the Internet," Apr. 1998, IETF RFC 2309.

[9] K. Stulmuller, N.Farberand, M.Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, June 2000.

[10] Y. Wang, Z. Wu, and J. Boyce, "Modeling of transmission-loss-induced distortion in decoded video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 6, pp. 716–732, June 2006.

[11] C. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. Automatic Control.*, vol. 47, no. 6, pp. 945–959, June 2002.

[12] Y. Huang, S. Mao, and S. Midkiff, "A control-theoretic approach to rate control for streaming videos," *IEEE Trans. Multimedia*, vol. 11, no. 5, Aug. 2009.

[13] G. Franklin, J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2005.

[14] D. Black and J. Donovan, *SystemC: From the Ground Up*. Boston, MA: Kluwer Academic Publishers, 2004.

[15] C. Oottamakorn, S. Mao, and S. Panwar, "On generalized processor sharing with regulated multimedia traffic flows," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1209–1218, Dec. 2006.

[16] R. Carter and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Perform. Eval.*, vol. 27-28, no. 4, pp. 297–318, Oct. 1996.

[17] S. Kompella, S. Mao, Y. Hou, and H. Sherali, "Cross-layer optimized multipath routing for video communications in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 831–840, May 2007.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.