A Control-Theoretic Approach to Rate Control for Streaming Videos

Yingsong Huang, Shiwen Mao, *Senior Member, IEEE*, and Scott F. Midkiff, *Senior Member, IEEE*

***Abstract*—As streaming videos are becoming increasingly popular, it is important to understand the end-to-end streaming system and to develop effective algorithms for quality control. In this paper, we address the problem of rate control for streaming videos with a control-theoretic approach. Among the various control knobs, video bit rate is one of the most effective in the sense that it has a direct impact on the interaction between the video coder and network system. While increasing rate reduces the coder-induced distortion, it may also cause congestion at a bottleneck link. The packet loss due to congestion will, then, increase the distortion of the decoded video. We model end-to-end video steaming as a feedback control system, taking into account video codec and sequence characteristics, rate control, active queue management, and receiver feedback. We then develop effective proportional (P) controllers to stabilize the received video quality as well as the bottleneck link queue, for both homogeneous and heterogeneous video systems. Simulation results are presented to demonstrate the efficacy of the *P* controllers and the viability of the proposed control-theoretic approach.**

***Index Terms*—Congestion control, feedback control, rate control, stability, video streaming.**

I. Introduction

Due to the rapidly increasing demand of multimedia applications over the Internet, video communications has become an important problem area attracting considerable efforts from both networking and signal processing research communities. While delivering video data over a wireline or wireless network, its stringent quality-of-service (QoS) requirements necessitate a careful reexamination of the originally data-driven network architecture and underscores the need for a new family of "multimedia-aware" protocols or enhancements.

To provide QoS guarantees for video, an effective and indispensable approach is cross-layer design, which allows overcoming the barriers created by layering and more effectively utilizing the available bandwidth in resource-limited networks. However, many existing cross-layer designs for QoS provisioning are of the "network-centric" nature, focusing on general performance metrics such as throughput and delay. As observed in a few works [1]–[3], video quality is a complex function of multiple coupled network layer metrics; optimizing one or more of them does not necessarily lead to superior video quality. A possibly more effective approach is to directly optimize video quality when developing new QoS mechanisms/protocols, in order to fully exploit the potential of cross-layer design. It is expected that such a *quality-driven cross-layer design* approach will enable a broad spectrum of emerging video communication applications.

In this paper, we address the problem of rate (and congestion) control for streaming videos. In prior work, the rate control problem has been well studied at an isolated encoder (see Section VI). But, existing congestion control algorithms within networks, such as realized in TCP and DCCP [4], are inherently data-oriented. They may not be suitable for video since the video characteristics are not explicitly considered. Jointly designing the video codec (i.e., encoder rate control) and the transport layer (i.e., congestion control) opens a new dimension of freedom for high-quality streaming videos.

We consider online coding where the coder can change the video rate on the fly. Such adaptive coder is necessary to react to network status changes as well as dynamics in the video sequence itself (since video frames usually contain different amount of information and hence different amount of bits). For superior perceived quality, we need to stabilize the video quality via a proper control mechanism. Consider an example where five video streams share a bottleneck link, each being a *Foreman* QCIF sequence coded at 15 frames/s. Assume that the servers are not congestion-aware. As shown in Fig. 1, initially the peak signal-to-noise ratio (PSNR) of a received video stays around 31 dB. At 10 s, the capacity of the bottleneck link drops to 77% of its original value because of the activity of a higher priority user. As a result, the PSNR curve degrades sharply after 10 s. At 20 s, the higher priority user leaves the system and the capacity for the video streams return to 100%. However, the PSNR cannot return to the original value (even 10 s after the link rate is restored), due to the lack of an effective control algorithm that is adaptive to network congestion.

Motivated by the seminal work on applying feedback control theory to TCP and active queue management (AQM) [5], [6], we propose a control-theoretic approach to rate control for streaming videos. We model the overall streaming system, including the codec, the bottleneck link where AQM is enforced, and feedback from receiver as a *feedback control system*. We

Manuscript received October 07, 2008; revised April 07, 2009. Current version published September 16, 2009. This work was supported in part by the National Science Foundation (NSF) under Grant ECCS-0802113 and in part through the Wireless Internet Center for Advanced Technology at Auburn University. This material was also based on work supported in part by the NSF, while S.F. Midkiff was working at the Foundation. Any opinion, finding, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aggelos K. Katsaggelos.

Y. Huang and S. Mao are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201 USA (e-mail: yzh0002@auburn.edu; smao@ieee.org).

S. F. Midkiff is with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: midkiff@vt.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2009.2026085

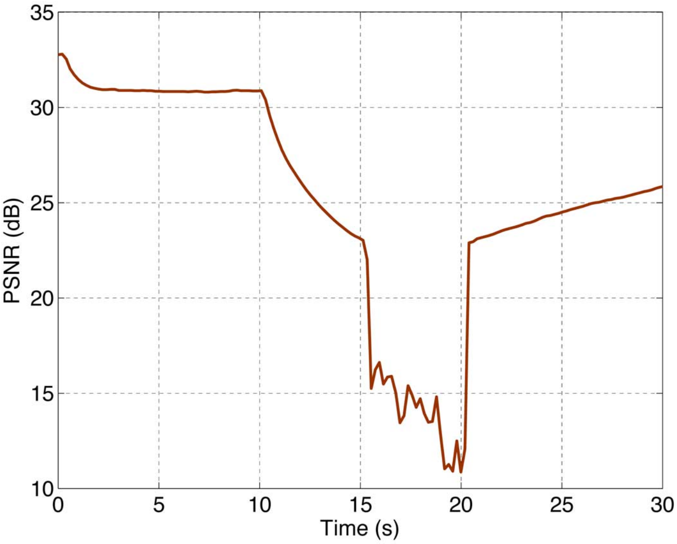


Fig. 1. Without a properly designed rate controller, the PSNR of the received video degrades severely after the bottleneck capacity C drops at 10 s and does not return to the previous level after C resumes its previous value at 20 s.

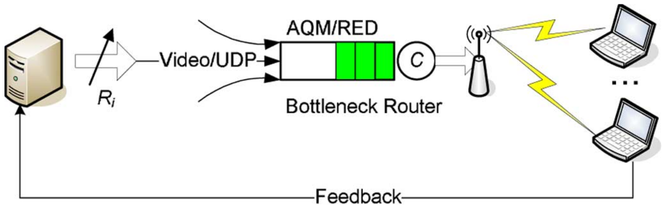


Fig. 2. Network and video session model.

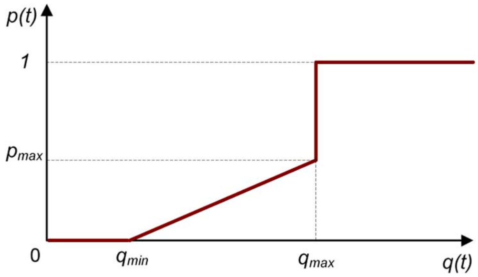


Fig. 3. RED drop probability $p(t)$ as a function of queue length $q(t)$.

first consider the case of homogeneous videos. Based on a properly chosen operating point, we can linearize the complex system and develop a proportional (P) controller to stabilize the video quality as well as the bottleneck link queue. We also derive the conditions for stability, and the phase and gain margins of the proposed controller. We next consider the more interesting and challenging case of heterogeneous videos. Applying *Lyapunov stability analysis*, we show that the P controller can also effectively stabilize the heterogeneous video system. Unlike prior work on TCP over wireless networks, where much effort has been made to distinguish packet loss due to wireless transmission errors or congestion, the proposed controller only requires feedback of the overall packet loss rate for tuning the video rate. We evaluate the performance of the proposed controller using a simulator developed with C++ and SystemC [7], which integrates an H.264 codec with network simulation modules. Our simulation results verify the accuracy of our analysis, and clearly 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 P controller in Section III. The case of heterogeneous videos is examined in Section IV. We present simulation results in Section V and discuss related work in Section VI. Section VII concludes this paper.

II. System Model

A. Network and Bottleneck Router

Consider N video sessions sharing a bottleneck router. The video flows are transmitted over a wired network, possibly with a wireless last hop, where the video is transmitted to client from a wireless access point. We assume the UDP transport protocol for video data. For streaming videos, it is highly important to be adaptive to network status. Otherwise, either a large portion of video data will be dropped under congestion, or the available network bandwidth will be wasted. We assume the video encoder on the server side can adjust its bit rate R according to feedback from the receiver to react to congestion in the network.

We assume a bottleneck link where N video sessions share the link bandwidth. The bottleneck could be a router link along the path, or the last-hop wireless link, as shown in Fig. 2. Loss of video packets occur at the bottleneck link due to congestion and at the wireless link due to interference or collisions. The bottleneck router is assumed to enforce a drop probability $p(t)$ when congestion occurs, and $p(t)$ is determined according to an AQM algorithm, i.e., random early detection (RED) [8].

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 small (i.e., 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 $q_{\max}$, $p(t)$ reaches the maximum drop probability $p_{\max}$. Beyond $q_{\max}$, all incoming packets will be dropped. Without loss of generality, we consider the *proportional (P) scheme* among the RED variations, where $p(t)$ is proportional to the queue length [5], [9]. The relation between drop probability $p(t)$ and queue length $q(t)$ is illustrated in Fig. 3. We are specifically interested in the following linear region:

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

where K_r is the RED parameter.

For video applications, packet losses cause considerable degradation in received video quality and should be avoided as much as possible [10]. Therefore, an effective control algorithm should keep the packet loss rate $p(t)$ close to zero [i.e., $q(t)$ close to $q_{\min}$]. We also take into account loss at all the other links along the path (e.g., loss at the wireless link). We assume the aggregate loss rate on all the non-bottleneck links is $p_w \in [0, 1]$. Usually p_w is a small value; otherwise, it may be hard to provide acceptable video quality. An estimate of the overall packet loss rate is

$$p_{tot} = p(t) + [1 - p(t)] \cdot p_w. \tag{2}$$

At the bottleneck link, assume the capacity reserved for video traffic is C.[1] Since it is the bottleneck link, the queue is in the busy state [i.e., $q(t) > 0$] with a probability close to 1. We first consider in this section the case of homogeneous video sessions, and will examine the more general case of heterogeneous videos in Section IV. According to Lindley's equation, the buffer dynamics can be written as [13]

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

where N is the load factor [5].

B. Video Rate-Distortion Model

We consider the distortion of the received video, which largely consists of *encoder distortion* D_e caused by quantization and *channel distortion* D_c due to network packet loss [10], [14], [15]. That is

$$D = D_e + D_c. \tag{4}$$

Based on the rate-distortion theory and experimental studies, many empirical models were proposed [14], [16], [17]. Without loss of generality, we adopt the empirical model introduced in [14], Under this model, the encoder distortion, measured as mean-square error (MSE), can be evaluated by

$$D_e = I_0 + \frac{\theta}{R - W_0} \tag{5}$$

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, and the *Foreman* and *Football* test sequences. The rate-distortion curves are plotted in Fig. 4, where the marks are measured distortions and the curves are computed from the model (5). We find the curves fit well with the measurements for both video sequences.

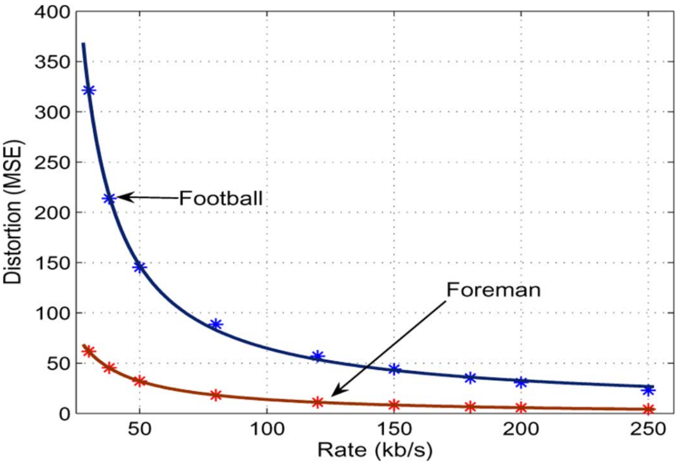


Fig. 4. Rate-distortion curves at the encoder for test sequence *Football* and *Foreman* both encoded at 15 frames/s with 11% intra rate.

The second component, channel distortion, is caused by packet loss within the network. We adopt the general model presented in [15] for channel distortion. Under this model, the decoder performs error concealments for macroblocks (MBs) affected by loss. Due to decoding dependencies, distortion in an earlier frame will propagate to the current frame and cause further distortion, even if the current frame is correctly received. Such error propagation can be stopped by coding the corrupted MBs in the *intra mode*. The channel distortion of the current frame is the sum of the error concealment distortion and the distortion of the earlier frame multiplied by a factor depending on the packet loss rate and the intra rate [15], as $D_{c,n} = p_{tot} \cdot D_{ECP,n} + \kappa(\beta_n, p_{tot}) \cdot D_{c,n-1}$, where $D_{ECP,n}$ is the average concealment distortion of frame n, and κ is the error propagation factor determined by the intra rate β_n and the packet loss probability p_{tot}.

It has been shown in [15] that with predetermined average D_{ECP} and β, $D_{c,n}$ is generally an exponentially increasing function of and converges to $D_c = \lim_{n\to\infty} D_{c,n} = p_{tot}/1 - \kappa(\beta, p_{tot})D_{ECP}$. Without intraprediction or with constrained intraprediction, in the special case of integer-pel motion vectors (MV) and no deblocking filtering, we have $\kappa = 1 - \beta \cdot (1 - p_{tot})$. The converged value is

$$D_c = \frac{p_{tot}}{\beta(1 - p_{tot})} D_{ECP}. \tag{6}$$

In (6), D_{ECP} is a constant and can be predetermined by training. In the following analysis, we use (6) to approximate the average channel distortion of a video frame.

C. Overall System Model and Linearization

Based on the buffer mode and video rate-distortion performance measure, we can derive the model for the overall video streaming system as (7), where $\dot{x}$ denotes the *time-derivative* of x, i.e., dx/dt. It can be seen that video distortion D is a function of R and p_{tot}. To improve video quality, we may use a larger coding rate R (i.e., to reduce the encoder distortion D_e). However, a larger rate may cause congestion at the bottleneck link and result in an increased p_{tot}, which will increase the channel distortion D_c. Such dynamics are modeled in (7). For varying network conditions, the tension between rate and loss dictates the interaction between the encoder and RED.

The model described in (7) is a nonlinear system, with the block diagram shown in Fig. 5. We next derive a linearized version of this system by considering the dynamics around a properly chosen operating point. Consider $q(t)$ as state, $R(t)$ as input, and $D(t)$ as output of the system.

$$\begin{cases} \dot{q(t)} = N \cdot R(t) \cdot [1 - p(t)] - C \\ D(t) = I_0 + \frac{\theta}{R(t) - W_0} + \frac{p_{tot}(t)}{\beta \cdot [1 - p_{tot}(t)]} D_{ECP} \\ p_{tot} = p(t) + p_w - p(t) \cdot p_w \\ p(t) = K_r \cdot [q(t) - q_{\min}]. \end{cases} \tag{7}$$

[1] We can assume generalized processor sharing (GPS) schedulers adopted at the routers in order to guarantee reserved bandwidth for video sessions [11]. In addition, a receiver may identify a bottleneck and estimate the bottleneck link capacity by adopting some network measurement techniques (see [12] and other network measurement literature).

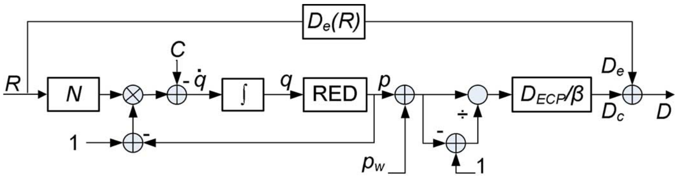


Fig. 5. Block diagram of the overall system model.

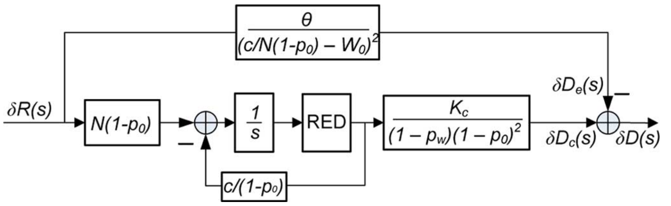


Fig. 6. Block diagram of the linearized model.

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. From (7), we have

$$\begin{cases} \dot{q} = 0 \Rightarrow N \cdot R_0 \cdot (1 - p_0) = C \\ D_0 = I_0 + \frac{\theta}{R_0 - W_0} + \frac{D_{ECP} \cdot (p_0 + p_w - p_0 \cdot {}_w)}{\beta \cdot (1 - p_0 - p_w + p_0 \cdot {}_w)} \cdot \end{cases} \tag{8}$$

Note that p_0 and p_w are usually small to meet the minimum QoS requirements of the video application (e.g., $p_0 + p_w \leq 0.1$[14]), which can be achieved by, e.g., adopting a QoS routing algorithm [1], [18] or cross-layer optimizations [19]. Letting $K_c = D_{ECP}/\beta$, we have from (8)

$$\begin{cases} R_0 = \frac{C}{N(1-p_0)} \\ D_0 = I_0 + \frac{\theta}{R_0 - W_0} + \frac{K_c \cdot (p_0 + p_w - p_0 \cdot p_w)}{(1 - p_0 - p_w + p_0 \cdot p_w)}. \end{cases} \tag{9}$$

We next linearize (7) around the operating point p_0. The following linearized state and output equations are obtained for the feedback system:

$$\begin{cases} \delta(\dot{q}(t)) = N(1 - p_0)\delta(R(t)) - \frac{C}{1 - p_0}\delta(p(t)) \\ \delta(D(t)) = -\frac{\theta}{(R_0 - W_0)^2}\delta(R(t)) + \frac{K_c}{(1-p_w)(1-p_0)^2}\delta(p(t)) \end{cases} \tag{10}$$

where $\delta(R) = R - R_0, \delta(p) = p - p_0$, and $\delta(D) = D - D_0$. Applying Laplace transform to the differential equations, we derive the linearized system block diagram as shown in Fig. 6. Recall that we adopt the *P scheme* for RED [5], [9]. With some manipulation of the block-diagram, we obtain the transfer function of the plant as

$$G_p(s) = \frac{N \cdot K_r \cdot K_c}{(1 - p_w)(1 - p_0)(s + \frac{K_r C}{(1 - p_0)})} - K_e. \tag{11}$$

In Fig. 6, the block diagram consists of two forward paths. Let $K_e = \theta/(R_0 - W_0)^2$ 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 $\delta D(s)$ is the difference between these two components, indicating that the change of bit rate R affects D_e and D_c on opposite directions. As bit rate increases, encoder distortion D_e decreases. But the increased bit rate usually leads to network congestion, resulting in 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 adjusting the bit rate. The net effect depends on the actual amount of increase/decrease in the two components.

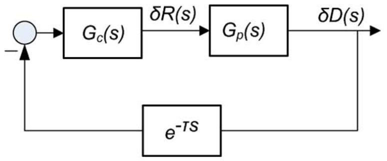


Fig. 7. Block diagram of the simplified linearized feedback control system.

Remark 1 (Stability): The pole of the linearized system is $s = -K_r \cdot C/(1 - p_0)$. Since K_r, C, and $1 - p_0$ are all positive values, the pole is located in the left phase plane. This indicates that the equilibrium state of the dynamics of the system is locally asymptotically stable.

To achieve an acceptable video quality, the encoder distortion D_e should be small. Generally, a video rate-distortion curve becomes quite flat when the rate is moderate or large (see Fig. 4). Around such an operating point, a moderate variation of the bit rate does not greatly affect the encoder distortion. However, the increased packet loss rate at the bottleneck router caused by the increased rate will cause relatively larger distortion. In other words, except for the very small rate region, video distortion is more sensitive to loss than rate. Based on this observation, we can further simplify the system model by only considering the lower path in Fig. 6 while omitting the upper path. The simplified system transfer function is

$$G_p(s) = \frac{N \cdot K_r \cdot \alpha}{s + K_r \cdot \frac{C}{(1 - p_0)}} \tag{12}$$

where $\alpha = K_c/[(1 - p_w)(1 - p_0)]$. The pole of (12) is $s = -K_r \cdot C/(1 - p_0)$, which is located in the left phase plane. The asymptotic stability property holds true.

III. Rate Control Problem

In this section, we design an effective controller for the linearized feedback control system (12). We redraw the block diagram in Fig. 7, where $G_p(s)$ is given in (12). Let τ denote the round-trip delay, while the $e^{-\tau s}$ term in Fig. 7 is the Laplace transform of τ. The $G_c(s)$ block represents the rate control component to be designed. 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. The goal of $G_c(s)$ is to stabilize the feedback loop system.

A. Performance Objectives

As discussed, packet loss is a main cause of video quality degradation, especially in real-time 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 for a consistent distortion level.

Furthermore, distortion at a client consists of encoder distortion and channel distortion. Encoder distortion is monotonically decreasing with the encoding bit rate. Because the $D(R)$ curve usually takes the form similar to a negative exponential function (see Fig. 4), using a very large bit rate, when exceeding some threshold, only achieves a small reduction of video distortion. On the contrary, this large bit rate may lead to serious congestion and high packet drop probability, which increases the channel distortion. Therefore, the rate controller should maintain a suitable encoding bit rate to achieve an acceptable distortion, but without causing congestion. To achieve this goal, the controller should take a small p_0 as operating point and maintain stable bottleneck queue length around $q_{\min}$.

Finally, the network status, e.g., the reserved capacity for video traffic, may change over time. The rate controller should be adaptive and robust to such disturbances. It should maintain the system stability and, more importantly, be able to drive it back to the operating point when disturbances disappear.

B. Stabilizing Control Law

Consider the system shown in Fig. 7, where $G_c(s)$ is the transfer function for the controller and $G_p(s)e^{-s\tau}$ is the transfer function of the plant. The closed-loop feedback control system should be stable to meet the above-mentioned performance objective. The overall system transfer function is

$$G(s) = \frac{G_p(s)G_c(s)}{1 + G_p(s)G_c(s)e^{-\tau s}}. \tag{13}$$

The following proposition is for the stabilizing control law.

Proposition 1: Given feasible network parameters $\{N, C\}$ with operating point $\{p_0, R_0, D_0\}$, the feedback control system in Fig. 7 is stable if $G_c(s)$ stabilizes the plant $G_p(s)e^{-s\tau}$.

C. Rate Control Using a Proportional Controller

A P controller for the feedback system simply performs scaling of the distortion errors. The classic *P* controller can be designed with the following transfer function:

$$G_c^P(s) = K. \tag{14}$$

This controller is implemented at the server side. It first evaluates the current distortion based on feedback of packet losses from the client. Then the server computes the error between the desired distortion and the evaluated distortion. The new bit rate for the video is then determined based on the error. The open loop transfer function is

$$L(s) = \frac{K \cdot N \cdot K_r \cdot \alpha}{s + K_r \cdot \frac{C}{(1-p_0)}} e^{-s\tau}. \tag{15}$$

We have the following propositions for the *P* controller.

Proposition 2: The feedback system in Fig. 7 is stabilized by the *P* controller, if the parameters satisfy the following conditions:

$$\begin{cases} \omega_c = \frac{\gamma}{\tau}, \text{for } 0 < \gamma \leq 0.59 \\ K_r \cdot \frac{C}{(1-p_0)} < \omega_c \\ K = \frac{C}{N \cdot \alpha \cdot (1-p_0)} \sqrt{\left[\frac{\gamma \cdot (1-p_0)}{K_r \cdot C \cdot \tau}\right]^2 + 1}. \end{cases} \tag{16}$$

Proof: Consider the amplitude of $L(s)$

$$|L(j\omega)| = \left| \frac{KN\alpha(1-p_0)}{\frac{(1-p_0)j\omega}{K_r} + C} e^{-j\omega\tau} \right|.$$

Letting $|L(j\omega_c)| = 1$, we can solve for K to have $K = C/N\alpha(1-p_0)\sqrt{[\gamma(1-p_0)/K_r C\tau]^2 + 1}$. From Proposition 2, the unit-gain crossover frequency is limited by $\omega_c = \gamma/\tau$. We have

$$\begin{aligned} \angle L(j\omega_c) &= -\arctan\left(\frac{\omega_c}{K_r \frac{C}{(1-p_0)}}\right) - \tau\omega_c \\ &\geq -\frac{\pi}{2} - \gamma_{\max} = -\frac{\pi}{2} - 0.59 > -\pi. \end{aligned}$$

Applying the Nyquist stability criterion for the feedback system, we prove that the system in Fig. 7 is stable. ■

A video server chooses a P controller [i.e., (14)] by computing value K as given in (16). For this purpose, the server only needs its own D_{ECP} and β, while N is returned by the router during the session setup phase. As will be shown in the heterogeneous video case, each video server computes its *P* controller value K_{c_i} also using (16) with local information individually. There is no need for the servers to exchange $D_{ECP,i}$ and β_i parameters.

Proposition 3: For a P controller satisfying Proposition 2, the gain margin (GM) and phase margin (PM) of the feedback control system satisfy: $PM \geq 56°$ and $GM > 6.06$ dB.

Proof: The phase at the crossover frequency ω_c is

$$\begin{aligned} \angle L(j\omega_c) &= -\arctan\left(\frac{\omega_c}{K_r \frac{C}{(1-p_0)}}\right) - \tau\omega_c \\ &\geq -\frac{\pi}{2} - \gamma_{\max} = -\frac{\pi}{2} - 0.59 \approx -124°. \end{aligned}$$

The phase margin is $PM = \angle L(j\omega_c) - (-180°) \geq 56°$.

Let ω_g be the frequency at which the phase reaches $-180°$. We have $\angle(L(j\omega_g)) = -\pi$. Since $\angle(KN\alpha(1-p_0)/[j\omega(1-p_0)/K_r + C]) \geq -\pi/2$, we have $\angle(e^{-j\omega_g\tau}) \leq -\pi/2$. Therefore, we obtain $\omega_g \geq \pi/2\tau$. From the first two constraints in Proposition 2, we have

$$\begin{aligned} |L(j\omega_g)| \leq \left|L\left(j\frac{\pi}{2\tau}\right)\right| &= \frac{\sqrt{\left(\frac{\gamma(1-p_0)}{K_r C\tau}\right)^2 + 1}}{\sqrt{\left(\frac{\frac{\pi}{2}(1-p_0)}{K_r C\tau}\right)^2 + 1}} \\ &< \frac{\sqrt{2}}{\sqrt{(\frac{\frac{\pi}{2}}{0.59})^2 + 1}} \approx 0.497. \end{aligned}$$

Then the $GM = 1/|L(j\omega_g)| > 2.01 = 6.06$ dB. ■

As a numerical example, consider a system with parameters $C = 1.5 \times 10^4$ kb/s, $N = 60$, $K_r = 5.862 \times 10^{-5}$, $p_w = 0.01$, and $\tau = 0.2$ s. The video sequence is *Foreman* coded using an H.264 codec at 15 f/s. We choose $p_0 = 0.006$, $\gamma = 0.5$ and draw the Bode plot of $L(j\omega)$ in Fig. 8. It can be verified

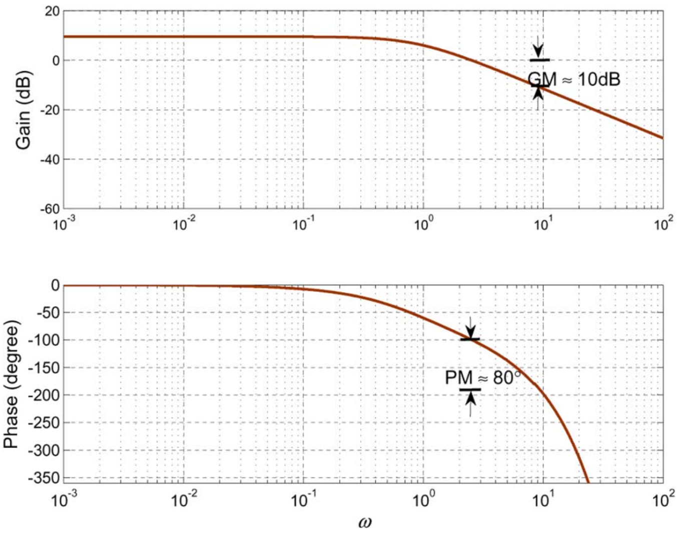


Fig. 8. Bode plot for the example system with a P controller satisfying Proposition 2.

that $GM \approx 10$ dB and $PM \approx 80°$ from Fig. 8. The feedback control system is stable with sufficient stability margins.

Proposition 4: For a system satisfying Proposition 2, the decrease of capacity C does not affect its stability.

Proof: Let $\hat{C} = \eta \cdot C$ be the new capacity value, where $0 < \eta < 1$, and $\hat{L}(j\omega) = K \cdot N \cdot K_r \cdot \alpha/[j\omega + K_r \cdot \hat{C}/(1-p_0)]e^{-j\omega\tau}$ be the new open loop transfer function. Following the same proof as in Proposition 3, we have

$$|\hat{L}(j\omega_g)| \leq \left|\hat{L}\left(j\frac{\pi}{2\tau}\right)\right| = \frac{\sqrt{\left(\frac{\gamma(1-p_0)}{K_r C\tau}\right)^2 + 1}}{\sqrt{\left(\frac{\frac{\pi}{2}(1-p_0)}{K_r C\tau}\right)^2 + \eta^2}}.$$

Since $\eta > 0$ and $0 < \gamma \leq 0.59$, $|\hat{L}(j\omega_g)| < \sqrt{2}/\sqrt{(\pi/2/0.59)^2} \approx 0.5312$. We have $GM = 1/|\hat{L}(j\omega_g)| > 1.88 = 5.49$ dB. Applying the Nyquist stability criterion, if $G_c^P(s)$ can stabilize $L(j\omega)$, then $G_c^P(s)$ can also stabilize $\hat{L}(j\omega)$. ■

Remark 2 (Choice of ω_c): The coefficient γ in ω_c is chosen for providing sufficient stability margin. We can choose a value that satisfies $PM > 45°$ and $GM > 6$ dB. We have

$$\angle L(j\omega_c) = -\arctan\left(\frac{\omega_c}{K_r C}\right) - \tau\omega_c$$
$$\geq -\frac{\pi}{2} - \gamma > -\frac{3\pi}{4}$$

and

$$L(j\omega_g) < \frac{\sqrt{2}}{\sqrt{(\frac{\frac{\pi}{2}}{\gamma})^2 + 1}} \leq \frac{1}{2}.$$

From the above equations, we obtain $\gamma \leq 0.59$. A larger γ provides higher system gain, with faster system response. But a larger γ will also produce a lower stability margin. In practice, we may choose γ according to the network situation to balance responsiveness and stability margin.

Remark 3 (Impact of Delay): Delay is an important factor for stability of the feedback system. A large τ may decrease the gain, thus making the controller less responsive. This is particularly harmful for real-time video streaming. A larger γ yields a smaller additional delay that the system can tolerate. For example, if we choose $\gamma = 0.1$, the phase margin is $PM \geq 84°$, the phase lag due to an additional delay $\delta(\tau)$ is $-\omega_c \cdot \delta(\tau)$. From $\omega_c \cdot \delta(\tau) = 84 \cdot \pi/180$, we have $\delta(\tau) \leq 14.6\tau$. If we choose $\gamma = 0.5$, we have $\delta(\tau) \leq 1.95\tau$.

Remark 4 (Proportional-Integral Controller): It is well known that proportional control cannot eliminate steady-state error. Generally, steady-state error decreases with the increase of system gain. However, a larger system gain will make the feedback control system less stable. An alternative approach is to adopt a classic proportional-integral (PI) controller, as

$$G_c^{PI} = \frac{K_{PI}}{s}\left(s + \frac{1}{T_{PI}}\right). \tag{17}$$

The integral part in (17) will cancel the disturbance and reduce the steady-state error. We will examine the design of a PI controller in our future work.

Remark 5 (Practical Considerations): The proposed algorithm can be implemented in a protocol as follows.

1) Clients send video streaming requests to servers. A server then sets up a video session using signaling messages. By doing so, the server estimates the bottleneck capacity, round-trip time, and number of video sessions at the bottleneck router. Routers are involved in this procedure by piggybacking N and C in the signaling messages.[2]
2) When the setup phase is over, the server extracts video parameters from the requested video. According to estimated network status, the server determines the target channel video distortion.
3) The server calculates the controller parameter according to Proposition 2 and starts the video session.
4) The client measures the average loss rate for video packets and returns it to the server as feedback.
5) The server computes the current distortion based on feedback and adjusts the video bit rate using the P controller, according to the difference between the target distortion and the computed distortion.

This procedure is implemented in C++ and SystemC and integrated with the H.264 codec in our simulation studies [7]. See Section V for more details.

IV. Case of Heterogeneous Videos

We considered the case of homogeneous videos in the previous section. In reality, there could be different video sessions sharing the bottleneck link. This more general scenario is considered in this section. We assume that the video sessions can be classified into M difference classes, while within each class, the video sessions are homogeneous. To simplify this challenging problem, we assume that the round-trip delays are small and can be ignored. We analyze such a system in this section, and apply *Lyapunov stability criterion* to prove that it is stable if Proposition 2 is satisfied [23].

[2] Such involvement is minimal as compared with the class of explicit congestion control approaches, such as Explicit Congestion Notification (ECN) [20], eXplicit Control Protocol (XCP) [21], and the Rate Control Protocol (RCP) [22].

In Lyapunov stability analysis, a *Lyapunov function* $V : \mathcal{R}^n \rightarrow \mathcal{R}$ is an energy-like *positive definite* function [23]. Roughly speaking, if we can find a suitable Lyapunov function that always has a negative "drift," we can conclude that the system is stable [24], [25]. In the following, we first show how to apply Lyapunov stability criterion for the relatively simple case of $M = 1$ and then show how to extend the result to the more general case of $M \geq 2$.

Consider the case of $M = 1$ for which (10) provides a linear model. Adopting a proportional controller $G_c^P(s) = K$ satisfying Proposition 2 and omitting the small D_e part, we can obtain the feedback system equation as follows:

$$\begin{cases} \delta(\dot{q(t)}) = N(1-p_0)\delta(R(t)) - \frac{C}{1-p_0}K_r\delta(q(t)) \\ \delta(D_c) = K_r\delta(q(t))\frac{K_c}{(1-p_w)(1-p_0)^2} \\ \delta(R) = -K\delta(D_c). \end{cases}$$

After manipulating the terms, we have

$$\delta(\dot{q(t)}) = \left[-\frac{K_rNK_cK}{(1-p_0)(1-p_w)} - \frac{K_rC}{1-p_0}\right]\delta(q(t)).$$

We choose a positive-definite Lyapunov function as

$$V(\delta(q)) = \frac{1}{2}\delta(q)^2. \tag{18}$$

It follows that

$$\dot{V} = \delta(q)\delta(\dot{q}) = \left[-\frac{K_rNK_cK}{(1-p_0)(1-p_w)} - \frac{K_rC}{1-p_0}\right]\delta(q(t))^2 < 0.$$

Since $\dot{V} < 0$, the feedback control system is asymptotically stable according to the Lyapunov stability criterion.

We next extend the proof to the general case of $M > 1$. Consider M classes of video sessions sharing a bottleneck link. Each class i has N_i homogeneous video sessions, each characterized by a set of parameters $\{I_i, \theta_i, W_i, K_{c_i}\}$, $i = 1, 2, \ldots, M$. We have the following system of equations:

$$\begin{cases} \dot{q(t)} = [N_1R_1(t) + \cdots + N_MR_M(t)][1-p(t)] - C \\ D_{c_1}(t) = \frac{p_{tot}(t)}{1-p_{tot}(t)}K_{c_1} \\ \cdots \\ D_{c_M}(t) = \frac{p_{tot}(t)}{1-p_{tot}(t)}K_{c_M} \\ p_{tot}(t) = p(t) + p_w - p(t)p_w \end{cases}$$

where $D_{c_i}(t)$ is the channel distortion for a class i video. We have $\sum_{i=1}^{M} N_iR_i(1-p_0) = C$ to fully utilize the available bandwidth at the bottleneck link.

We linearize the system of equations at the operating point p_0 and choose the Lyapunov function (18). Then we have

$$\dot{V} = \left[\frac{-K_r}{(1-p_0)(1-p_w)}\sum_{i=1}^{M} N_iK_{c_i}K_i - \frac{K_rC}{1-p_0}\right]\delta(q)^2 < 0.$$

Therefore, we conclude that the system with M classes of heterogeneous videos is asymptotically stable with the *P* controller satisfying Proposition 2.

V. Simulation Results

A. Simulation Setting

We evaluate the performance of the proposed feedback control system using a customized C++ simulator built on SystemC [7]. SystemC is a set of library routines and macros implemented in C++ for simulation of concurrent processes. Although often perceived as a *hardware description language* like VHDL, SystemC is more aptly described as a *system description language*, with its unique strength in transaction-level modeling and behavioral modeling.

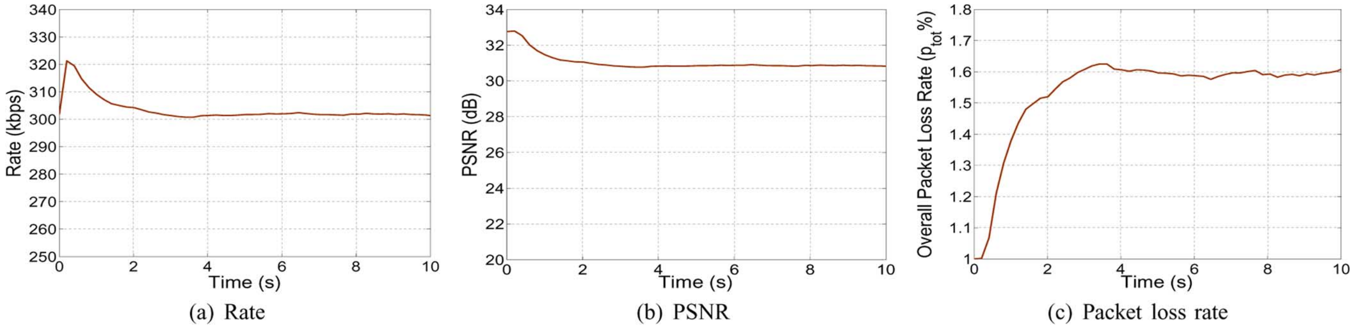
Although the P controller is designed under a linearized model, the real video streaming system is much more complex and nonlinear in nature. Therefore it is important to examine the impact of the approximations made in the analysis via truthful simulations. For this purpose, we integrate the H.264 (JM) codec [as a *dynamic-link library* (DLL)] with our SystemC network model, rather than using some dummy/simplified source models. Each server invokes the DLL to code the video sequence and encapsulates coded frames into UDP datagrams. The packets are transmitted to the corresponding client through the bottleneck router, where RED is implemented. Each client sends measured loss rate feedback to the server with delay τ. Note that the server only needs the overall packet loss rate p_{tot} for rate control. There is no need to distinguish between the RED-induced loss $p(t)$ and the aggregated loss at other non-bottleneck links p_w.

In the following simulations, we consider both the homogeneous video case (with five sessions) and the heterogeneous video case (with two classes).[3] The H.264 reference software encoder (JM) in the Baseline profile is used to generate the test sequence. Each encoded frame is carried in one UDP packet [15]. 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. For each sequence, the first frame of each group of 60 frames is coded in the intra-mode and all the remaining frames are P frames. The intra rate is 0.33 for *Football* and 0.03 for *Foreman*. The link bandwidth of the bottleneck router is set to $C = 1.5$ Mb/s. The bottleneck router buffer size is 2000 packets. The RED parameter is set to $K_r = 5.862 \times 10^{-5}$ [6]. The aggregate loss rate at all the non-bottleneck links is $p_w = 0.01$.

B. Simulation With Homogeneous Videos

We simulate a video streaming system where five sessions share the bottleneck link. The *Foreman* sequence is used in these simulations. We choose an operating point with RED drop probability $p_0 = 0.006$. The corresponding P controller for the feedback control system is $G_c^P(s) = 1.032$, when $\gamma = 0.5$. The study reported in [26] shows that 85% of the Internet traffic has round-trip delays between 5 and 500 ms, and 60% of the traffic has round-trip delay below 200 ms. We set the round-trip delay to $\tau = 200$ ms in the following simulations.

[3] The main limiting factor here is simulation time, since running multiple encoders simultaneously in a computer greatly slows down the simulation.



(a) Rate (b) PSNR (c) Packet loss rate

Fig. 9. Simulation results for the homogeneous video case with the *Foreman* sequence.

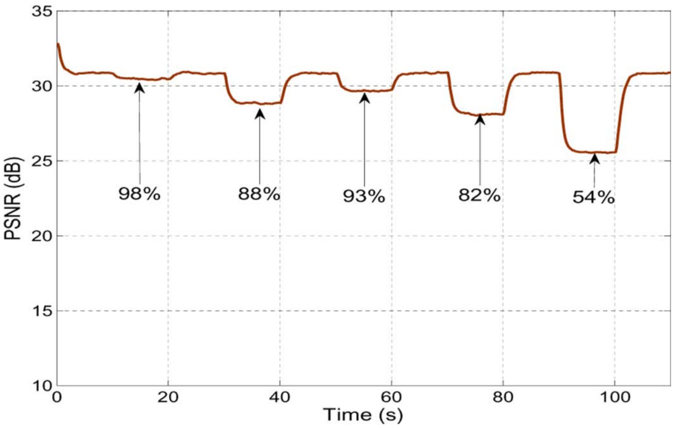


Fig. 10. PSNR of the *Foreman* sequence when capacity C changes in the homogeneous video simulation.

In Fig. 9, we plot the simulation results for the homogeneous video system. We find the bit rate, PSNR, and the overall packet loss rate all converge to the neighborhood of the corresponding steady-state values after about 4 s. Specifically, the total packet loss rate converges to the neighborhood of $p_{tot} = 0.016$, the PSNR converges to the neighborhood of 31 dB, and the encoding bit rate converges to the neighborhood of $R_0 = 301.8 \text{ kb/s}$. After convergence, the rate, distortion, and loss rate fluctuate around the corresponding steady-state values, but the differences are negligible.

We next examine the performance of the P 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 $\tau = 200 \text{ ms}$. We assume the high-priority user begins to consume bandwidth at 10 s and remains active for 10 s. During the period where the high-priority user is active, the bottleneck link bandwidth for video is dropped to a fraction of its original value uniform distributed in $[0.5, 1.0]$. After the 10 s period, the high-priority user leaves the network, the bottleneck link capacity for video assumes its original value.

The PSNR at a client is plotted in Fig. 10. For example, at 30 s, the decrease of bandwidth C leads to a decrease in PSNR from about 31 dB to 28 dB. The system is then stabilized at the new PSNR value. After the 10 s disturbance, the high-priority user becomes idle at 40 s, and the controller drives the system back to the original operating point, as the distortion value returns to 31 dB and stays there. Compared with Fig. 1, it can be seen the *P* controller is quite effective in stabilizing the video streaming system under network dynamics.

C. Simulation With Heterogeneous Videos

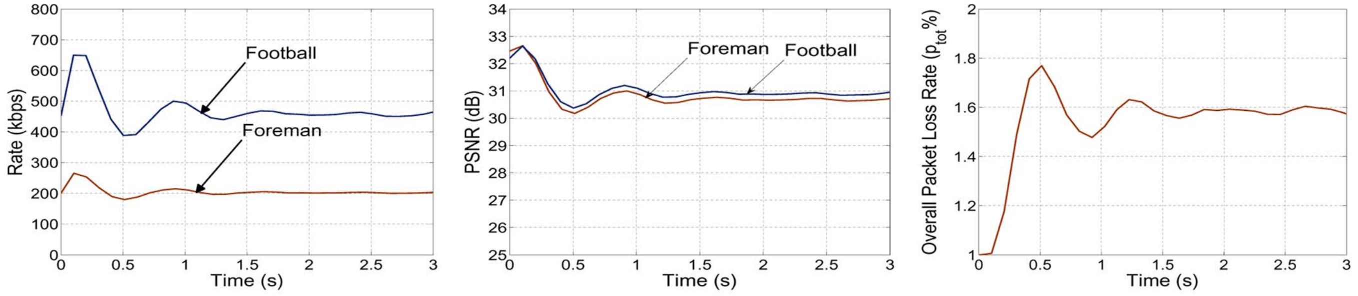
We next consider the case of heterogeneous videos. In the simulations, there are two classes of video streams: one consists of two sessions streaming the *Football* sequence, the other consists of three sessions streaming the *Foreman* sequence. Each video is encoded at the corresponding server as discussed in Section V-A. We choose an operating point $p_0 = 0.006$, such that the operating point for the overall packet loss probability is $p_{tot,0} \approx 0.016$. The round-trip delay is $\tau = 100 \text{ ms}$. We derive the controller for the *Foreman* class as $G_{c,1}^P(s) = 3.44$ and the controller for the *Football* class as $G_{c,2}^P(s) = 14.09$.

The heterogeneous video simulation results are plotted in Fig. 11. Again, fast convergence is observed. We find the heterogeneous video system achieves the desired distortion and drop rate in about 1.5 s. The overall packet loss rate converges to the neighborhood of $p_{tot} = 0.0158$. The PSNR of the *Football* class converges around 31 dB, and the PSNR of the *Foreman* class converges around 30.7 dB. The *P* controllers stabilize the heterogeneous video streaming network.

Finally, we examine the heterogeneous video system under network dynamics. Similar to the homogeneous video case, we add a disturbance to the bottleneck link capacity C to reduce it to a fraction uniformly distributed in $[0.5, 1.0]$ for 3 s and then resume the original C value for the next 3 s, and so forth. The PSNRs for the two received videos are plotted in Fig. 12. We find that for both videos, the reduced C causes degradation in received video quality. Furthermore, for both videos, the PSNRs return to the designed operating points when C resumes its original value.

VI. Related Work

Rate control has received considerable interest from the video coding community (e.g., see [17], [27]–[31], and references therein). Rate control algorithms have been incorporated in video coding standards, such as H.261/263/264 and MPEG-1/2/4. Most existing work concerns how to allocate a given bit rate budget to the frames or MBs at the video coder, while a given bit rate can be met by adjusting the quantization parameter or by skipping certain frames. We revisit this problem by jointly considering the encoder rate control and network congestion control, with a control-theoretic approach.



(a) Rate (b) PSNR (c) Packet loss rate

Fig. 11. Simulation results for the heterogeneous video case with the *Football* and *Foreman* sequences.

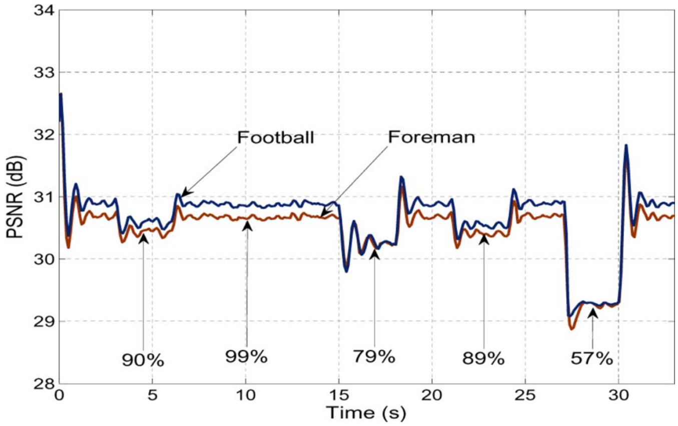


Fig. 12. PSNR of the *Football* and *Foreman* sequences when capacity C changes in the heterogeneous video simulation.

Our work actually complements this class of work by finding the bit rate that is allowed by the network condition, while we do not assume any specific encoder rate control mechanism.

Many networking problems can be formulated as control problems. In the seminal works [5], [6], the authors take a control-theoretic approach to the Internet congestion control problem by modeling TCP and RED as a feedback control system. Through the control models, the authors derive useful insights on TCP stability and guidelines for setting RED parameters. This line of work is extended in [9] to the general additive-increase-multiplicative-decrease (AIMD) mechanism with feedback delays. In [32], Choi and Park propose a stable feedback control algorithm for the encoder buffer using the controlled Lagrange multiplier approach. The authors focus on the video coder without considering the more challenging network congestion control problem. The stability of wireless networks is studied using Lyapunov stability analysis in [24] and [25], where capacity region-achieving distributed scheduling algorithms are developed. These successful stories clearly demonstrate the strength of the control-theoretic approach to solving networking problems, and motivated this work on video streaming systems.

Cross-layer design and optimization of streaming video has recently become an active research area (see [2], [3], and references therein). Much of this existing work focuses on video communications in wireless networks, with joint design and optimization of PHY, MAC, routing, and video quality [1], [18], [19]. The problem considered in this work involves application layer (video) and transport layer (congestion control). There is a general trade-off between the achievable performance gain and complexity for this class of work.

VII. 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 derived a P controller that stabilizes both homogeneous video and heterogeneous video systems. We developed a SystemC-based simulator that integrates an H.264 codec with the networking simulation modules. Through our simulation studies, we found the proposed approach highly effective in rate control for streaming videos.

References

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

[2] S. Ci, H. Wang, and D. Wu, “A theoretical framework for quality-aware cross-layer optimized wireless multimedia communications,” *Adv. Multimedia*, vol. 2008, pp. 1–10.

[3] Q. Zhang and Y.-Q. Zhang, “Cross-layer design for QoS support in multihop wireless networks,” *Proc. IEEE*, vol. 96, no. 1, pp. 64–76, Jan. 2008.

[4] V. Balan, L. Eggert, S. Niccolini, and M. Brunner, “An experimental evaluation of voice quality over the datagram congestion control protocol,” in *Proc. IEEE INFOCOM’07*, Anchorage, AK, May 2007, pp. 2009–2017.

[5] C. Hollot, V. Misra, D. Towsley, and W. Gong, “A control theoretic analysis of red,” in *Proc. IEEE INFOCOM’01*, Anchorage, AK, Apr. 2001, pp. 1510–1519.

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

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

[8] B. Braden *et al.*, Recommendations on Queue Management and Congestion Avoidance in the Internet, 1998, IETF RFC 2309.

[9] L. Wang, L. Cai, X. Liu, and X. Shen, “Stability and TCP-friendly of AIMD/RED system with feedback delays,” *Elsevier Comput. Netw. J.*, vol. 51, no. 15, pp. 4475–4491, Oct. 2007.

[10] S. Mao, S. Kompella, Y. Hou, H. Sherali, and S. Midkiff, “Routing for concurrent video sessions in ad hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 317–327, Jan. 2006.

[11] 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.

[12] 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.

[13] D. Lindley, "The theory of queues with a single server," *Proc. Cambridge Phil. Soc.*, vol. 48, no. 2, pp. 277–289, Apr. 1952.

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

[15] 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, Jun. 2006.

[16] S. Ma, W. Gao, and Y. Lu, "Rate-distortion analysis for H.264/AVC video coding and its application to rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1533–1544, Dec. 2005.

[17] 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.

[18] S. Kompella, S. Mao, Y. Hou, and H. Sherali, "On path selection and rate allocation for video in wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 212–224, Feb. 2009.

[19] D. Wu, S. Ci, and H. Wang, "Cross-layer optimization for video summary transmission over wireless networks," *IEEE J. Select. Areas Commun.*, vol. 25, no. 4, pp. 841–850, May 2007.

[20] K. Ramakrishnan, S. Floyd, and D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, 2001, IETF RFC 3168.

[21] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proc. ACM SIGCOMM'02*, Pittsburgh, PA, Aug. 2002, pp. 89–102.

[22] C.-H. Tai, J. Zhu, and N. Dukkipati, "Making large scale deployment of RCP practical for real networks," in *Proc. IEEE INFOCOM'08 Mini-Conf.*, Phoenix, AZ, Apr. 2008, pp. 2180–2188.

[23] K. Astrom and R. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ: Princeton Univ. Press, 2008.

[24] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[25] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[26] M. Allman, "A web server's view of the transport layer," *ACM Comput. Commun. Rev.*, vol. 30, no. 5, pp. 10–20, Oct. 2000.

[27] 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.

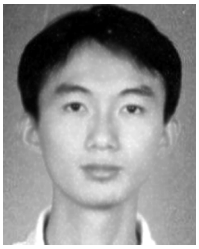
[28] S. Aramvith, I. Pao, and M. Sun, "A rate-control scheme for video transport over wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 569–580, May 2001.

[29] 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.

[30] 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.

[31] 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.

[32] J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled Lagrange multiplier method," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 546–558, Sep. 1994.



Yingsong Huang received the B.S. and M.S. degrees from Chongqing University, Chongqing, China, in 2000 and 2003, respectively. Since 2007, he has been pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL.

From 2003 to 2007, he was with Advantech Co., Ltd. R&D Center, Beijing, China. His research interests include network congestion control and video communications.

Shiwen Mao (S'99–M'04–SM'09) received the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, in 2004.

He was a Research Member with IBM China Research Lab, Beijing, China, from 1997 to 1998 and a Research Intern with Avaya Labs-Research, Holmdel, NJ, in summer 2001. He was a Research Scientist in the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, from December 2003 to April 2006. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL. His research interests include cognitive radio networks, cross-layer design and optimization of wireless networks, and multimedia communications. He is a coauthor of *TCP/IP Essentials: A Lab-Based Approach* (Cambridge, U.K.: Cambridge Univ. Press, 2004).

Dr. Mao is on the Editorial Board of *Advances in Multimedia* and *International Journal of Communication Systems*. He received the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems and the Best Paper Runner-up Award from the Fifth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine) 2008.

Scott F. Midkiff (S'82–M'85–SM'92) received the B.S.E. and Ph.D. degrees from Duke University, Durham, NC, and the M.S. degree from Stanford University, Stanford, CA, all in electrical engineering.

He worked at Bell Laboratories and held a visiting position at Carnegie Mellon University, Pittsburgh, PA. In 1986, he joined the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, where he is now a Professor. He is now with the National Science Foundation (NSF) as a Program Director for the Integrative, Hybrid, and Complex Systems (IHCS) Program of the Electrical, Communications, and Cyber Systems (ECCS) Division in the Directorate for Engineering (ENG). His research interests include system issues in wireless and ad hoc networks, network services for pervasive computing, and performance modeling of mobile ad hoc networks.