Delay-Constrained Rate Control for Real-Time Video Streaming over Wireless Networks

Yufeng Geng [1], Xinggong Zhang [1,2], Tong Niu [1], Chao Zhou [1], Zongming Guo [1,2,*]

[1] *Institute of Computer Science & Technology, Peking University, Beijing, 100871, P.R. China*

[2] *Cooperative Medianet Innovation Center, Shanghai, China*

guozongming@pku.edu.cn

***Abstract*—Rate control is a big challenge for real-time video streaming on the internet with the needs of low latency, bandwidth-consuming and stable video rate. However, most of the existing Internet congestion control protocols ignore these needs, and some of them use the packet loss event as congestion signal which is deviation especially over error-prone wireless networks. In this paper, we propose a delay-constrained rate control algorithm by locking queueing delay onto a desired objective. The shadow price of video rate is controlled by queueing delay. All flows adapt video rate according to distortion weight and shadow price so as to achieve a distributed bandwidth sharing with low latency, efficient utilization, and distortion fairness. A closed-loop rate control system is designed for the purpose of stable and agile control. The control parameters are analyzed using control-theoretic approach. Additionally, we construct a real-time wireless video streaming test-bed and conduct extensive experiments over it. Compared with the current widely used methods, the experimental results show that the proposed algorithm can achieve 3dB or more gains in PSNR, and better performance on bandwidth utilization, flow stability with well guaranteed multi-flow fairness.**

***Index Terms*—Real-time Video Streaming, Rate Adaptation, Congestion Control, Delay-constrained, Control Theory**

I. Introduction

With the popularity of camera-ready mobile devices and 3G/4G/WiFi wireless infrastructure, interactive video applications with wireless Internet access links are growing exponentially. However, real-time video streaming over wireless networks are still facing many challenges such as varying latency, bandwidth fluctuation, and wireless physical packet losses. It is critical for video streaming to adapt video rate to avoid network congestion such as "drop-tail" packet losses, queueing delay or bandwidth under-utilization. But wireless physical errors make it hard for real-time streaming to use only packet-loss events as the congestion control signal.

Video rate adaptation over wireless networks is still an open issue. Although there are already many prior works about Internet congestion control, such as TCP Reno[1], TCP Vegas[2] and Cubic[3] etc. Most of these TCP style schemes maintain a congestion window that increases additive to probe the bandwidth and decreases multiplicative when congestion detected. Since it typically introduces intolerable delay and rate oscillation, it's not usually employed by real-time video streaming. On the other hand, Rate-based Congestion control[4] sets video rate according to the methods like Additive Increase Multiplicative Decrease (AIMD), or by explicitly some mathematical model, like TCP-Friendly Rate Control (TFRC)[5]. However, these algorithms use packet loss events as the congestion signal, which result in bandwidth under-utilization as wireless physical errors also cause packet losses. Motivated by these and other potential benefits,delay-based congestion control remains an active area of research and new algorithms continue to be developed, recent examples include Fast TCP[6] and AIMD variants[7][8]. But the final delay of these schemes is not constrained, which may be too large for real-time applications.

Our aim is to design a specialized rate adaptation algorithm for real-time video streaming over wireless networks, which takes into consideration the factors of wireless physical losses, low streaming latency, and rate stability. The algorithm should achieve the following goals: (1) packet loss insensitivity; (2) low transmission delay; (3) stable video rate; (4) distortion-fair bandwidth sharing.

In this paper, we propose a delay-constrained rate control algorithm for real-time video streaming over wireless networks. First we introduce a fluid-queueing model to investigate the insight of network congestion. By maintaining queueing delay under the desired objective, we adjust the shadow price for all flows to avoid network congestion and maximize network utilizations. Using queueing delay as feedback congestion signal, we build a closed-loop rate control system which is insensitive to packet losses. To make the controller more stable and agile, a proportional controller is introduced and the control parameters are analyzed using control-theoretic approach. Additionally, we construct a fully-functional real-time wireless video streaming test-bed and conduct extensive experiments over it. Compared with current widely used protocols, such as TFRC and CTCP[9], the experimental results show that our algorithm can achieve 3 dB or more gains in PSNR, and better performance on bandwidth utilization and flow stability.

The main contributions of this paper are three folds:

- We propose a fluid-queueing model to explore the insight of network congestion, which enables us to design a distributed rate control algorithm with low latency and distortion fairness.
- By controlling the delay under the desired objective,

*corresponding author

This work was supported by National High-tech Technology R&D Program (863 Program) of China under Grant 2013AA013504 and National Natural Science Foundation of China under contract No. 61471009.

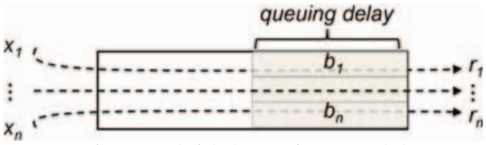


Fig. 1: Fluid-Queueing Model

we build a loss-insensitive rate control system for video streaming over wireless networks.

- We employ a control-theoretic approach to analyze the control parameters.

The rest of the paper is organized as follows. The algorithm design would be presented in section 2. And section 3 will give the implementations. To evaluate our algorithm, we conduct extensive experiments on our wireless streaming test-bed, and the results will be shown in section 4. We conclude our work in section 5.

II. Algorithm Design

In this section, we first introduce a fluid-queueing model to explore the design space of rate adaptation in wireless video streaming with low delay, packet loss insensitivity and flow stability. Then we propose a closed-loop rate control scheme to adapt video rate to network congestion, and employ control theory to tune the rate controller.

A. Fluid-queueing Model

Network congestion occurs when there are too many packets flooding into the network, which in turn results in queuing delay, or even more packet losses when network devices overflow.

We model the network as a fluid queue as shown in Fig. 1. Network routers can be seen as a "store and forwarding" buffer. N flows compete for bandwidth by emitting packets into the buffer. Denote that the ith flow occupies b_i packets in the buffer, its enqueue rate is x_i, and the total buffer bandwidth is C. From the perspective of the buffer, we have:

$$b'(t) = \sum_{i=1}^{N} x_i(t) - C \tag{1}$$

where $b'(t) = db/dt$ is the derivative function of queue length with respect to time t.

The transmission delays q denotes the time required for a packet to travel from the source client to destination, which is also referred to as RTT. Generally, this transmission delay consists of two parts: propagation delay and queuing delay. The propagation delay is the transmitting time on the link, which is almost fixed for a given circumstance while queuing delay varies with the buffer size all the time. q is formulated as,

$$q(t) = \frac{b(t)}{C} + t_p \tag{2}$$

Since flows have no information about the action of others, we design a distributed congestion algorithm according to the shadow price method[5]. Each flow has a weight ω_i which is related to video rate distortion curves. It tries to grasp its share of bandwidth to achieve video distortion fairness. The shadow price $p_i(t)$ denotes the cost of network congestion. Flows pay the cost for its video rate with the price. The rate control by distortion weight and shadow price is formulated as:

$$x'_i(t) = \omega_i - x_i(t)p_i(t) \tag{3}$$

where $x'_i(t) = dx_i/dt$ is the derivative function of video rate with respect to time t.

Each flow changes its rate according to Eq.(3). In case of congestion, a fast flow would drop rate quickly since all flows have close shadow price and the fast one will pay more cost. In the end, all flow obtain a distortion-fair share in a distributed way.

Unlike the queueing-delay based control law in FAST TCP[6], here we decouple queuing delay and rate control in order to avoid accumulative queueing delay.

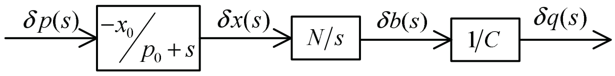


Fig. 2: Block diagram of the feedback control system.

B. Closed-loop Feedback Control

For simplicity, we assume that all clients are homogeneous, namely they have the same ω_i and target sending rate x, then Eqs.(1-3) can be simplified as

$$x' = \omega - xp \tag{4}$$

$$b' = Nx - C \tag{5}$$

$$q = \frac{b}{C} + t_p \tag{6}$$

Moreover, the control system illustrated above is a nonlinear system, which is hard to be analyzed. Here we take the differential form around the operation point to transform it into a linear system.

Denote the shadow price at operation point to be p_0 and the corresponding rate as x_0. From Eqs.(4-6), we have:

$$\omega = x_0 p_0 \tag{7}$$

$$x_0 = \frac{C}{N} \tag{8}$$

With difference being $\delta p(t) = p_0 - p(t)$, $\delta x(t) = x_0 - x(t)$ and $\delta q(t) = q_0 - q(t)$, we have:

$$\delta b'(t) = \sum_{i=1}^{N} \delta x(t) \tag{9}$$

$$\delta q(t) = \frac{\delta b(t)}{C} \tag{10}$$

$$\delta x'(t) = -p_0 \delta x(t) - x_0 \delta p(t) \tag{11}$$

Applying the Laplace transform to Eqs.(9-11), we derive a linear control block diagram in the complex-s domain as shown in Fig. 2.

The main idea of our algorithm is to control p, the shadow price of flows, by feedback delay. According to Fig. 2, we consider a differential form controller as:

$$\delta p(s) = G_c(s)\delta q(t) \tag{12}$$

which makes it a closed-loop feedback control system as in Fig.3.

C. Proportional Controller Design

To decouple the binding relation between p and q, here we employ a proportional controller to control the shadow price,

$$G_c(s) = K_P \tag{13}$$

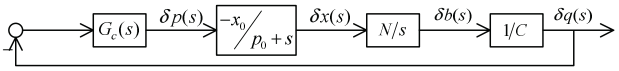


Fig. 3: Block diagram of the feedback control system.

Combining Fig. 3 and the controller in (13), the closed-loop transfer function of the rate control system can be written as

$$H(s) = \frac{K_P}{s^2 + sp_0 + K_P} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \tag{14}$$

where ζ is the damping ratio, and ω is the natural frequency of the control system. Thus we have $2\zeta\omega = p_0$ and $\omega^2 = K_P$, which yields that $\zeta = \frac{p_0}{2\sqrt{K_P}}$. To obtain good dynamic performance of the system, ζ is set to $\frac{\sqrt{2}}{2}$[10], which is known as the optimal damping ratio. We have

$$K_P = \frac{p_0^2}{2} \tag{15}$$

and the 5% settling time of the system is $t_s = \frac{6}{p_0}$.

III. IMPLEMENTATION DETAILS

In order to focus our work on video rate control, we build our test-bed on Mediastreamer2, an open source streaming engine. We implement an independent rate control module based on our algorithm, dealing with packet transmitting, collecting data required by the rate control algorithm and calculating the recommended video rate. Meanwhile, we inherit most features of the original system, and the rate-adaptive video encoding and rate shaping[4] that force the source to send the video stream at the rate dictated by the rate control algorithm are done by Mediastreamer2. In this section, we discuss some implementation details about our proposed rate control approach.

A. Update of shadow price

From the proportional controller, we know that

$$\delta p = K_P \delta q \tag{16}$$

When it comes to implementation, it's hard to know p_0 beforehand. To avoid the initialization value of p, we take the differential form of Eq.(16),

$$\delta p(t) - \delta p(t-1) = K_p(\delta q(t) - \delta q(t-1)) \tag{17}$$

thus, we have

$$\delta p(t) = \delta p(t-1) + K_p(\delta q(t) - \delta q(t-1)) \tag{18}$$

In this way, the value of p approaches to p_0 piecemeal. Meanwhile, we dynamically update K_P accordingly.

B. Rate controller

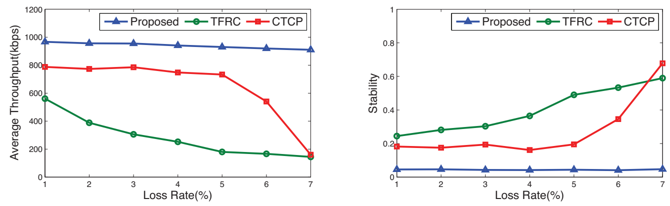
The change rate of x, x', is calculated according to Eq.(3), with rate adaptation interval ΔT, we derive the optimal sending rate as

$$x(t) = x(t-1) + x'(t)\Delta T \tag{19}$$

We adapt the video bitrate at the beginning of every time period. In the beginning, the sending rate is set to be the minimum acceptable bitrate of the codec, e.g. 64Kbps.

IV. EXPERIMENT

In this section, we evaluate our rate control algorithm under controlled environment on our test-bed and compare it with current widely used methods.



(a) Average throughput (b) Stability

Fig. 4: Performance under different loss rate with bandwidth=1Mbps and latency=50ms.

A. Experiment Setup

Our test-bed consists of three nodes, one performs as a router, on which the Network Emulator for Windows Toolkit is implemented to simulate wireless network by injecting bandwidth limitation, packet loss, and propagation delay. The other two nodes are performed as the sender and receiver respectively.

With theoretical analysis and experiments, we choose a series of value that yields relatively good performance. We set ΔT to 1s. The targeting queuing delay is set to $q_0 = 200$ms, balancing between bandwidth usage and latency limitation.

The performance is compared with TFRC, a typical model-based rate control algorithm and an AIMD based algorithm CTCP. TFRC directly uses the TCP throughput modeling equation to set target video sending rate which is a function of packet loss event rate and round-trip time. While CTCP algorithm represents a category of rate control algorithms, which emulate the behavior of TCP that adjusts video rate by increased-by-one and decreased-to-half.

B. Evaluation Metric

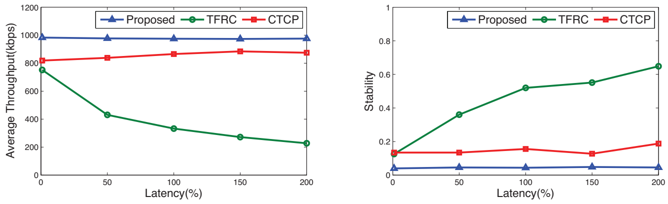
The matrices we concern about in this paper are average throughput, stability as transmitting performance, and PSNR as QoS. The average throughput is recorded at the receiver for 60 seconds after the flow becomes stable. And the stability is the sample standard deviation normalized by the average throughput:

$$Stability = \frac{1}{\bar{x}}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

PSNR is calculated between the source and recorded CIF sequence.

C. Average throughput and stability evaluation

We first evaluate the average throughput and stability performance of our proposed rate control method. The results in Fig. 4 show the performance under different packet loss rate. From the results, we can see that our proposed method always performs the best. This is mainly because our proposed algorithm uses queuing delay as the control signal rather than packet loss event, which makes it tolerant to error-prone environments. Thus, the performance remains at a high level for both throughput and stability while loss rate increases greatly.



(a) Average throughput (b) Stability

Fig. 5: Performance under different latency with bandwidth=1Mbps and loss rate =1%.

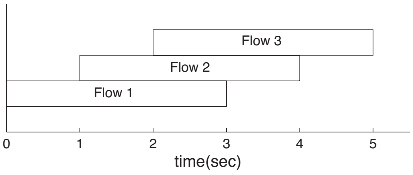


Fig. 6: Multiple flows: active periods.

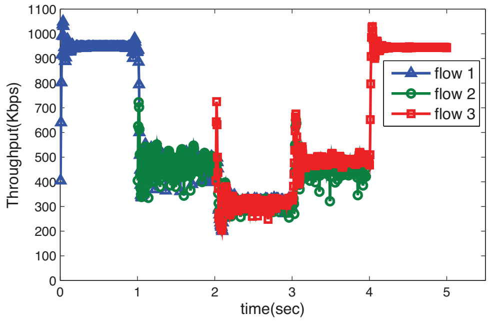


Fig. 7: Throughput trajectory of different flows.

Also, we have compared the performance under different latency and the results are shown in Fig. 5. From the results, we can see that all these algorithms acquire large throughput when latency is low. With latency increasing, the throughput and stability performance of TFRC decrease dramatically, as latency is included in its equation while the other two algorithms remains the same.

D. Intra-protocol fairness

In this case, we test the fairness between different flows of our algorithm. There are three flows that start and terminate at different times, as illustrated in Fig.6. From the sender, we obtain the trajectory of individual connection throughput over time. The result is shown in Fig.7. The result shows that our algorithm reacts fast to network fluctuation, and connections sharing the link achieve very similar rates. There's a reasonably stable sending rate, with high bandwidth utilization.

E. Video quality measurement

At last, we have compared the PSNR performance of our proposed algorithm with TFRC and CTCP, and the result is plotted in Fig. 8. The result has demonstrated that much higher PSNR is obtained in our proposed method. The significant performance improvement comes from that we have designed a PI controller for rate control by using the queuing delay as the control signal, which obtains much higher bandwidth usage and stability. There's less frame loss, latency and rate fluctuation so that better PSNR performance is obtained.

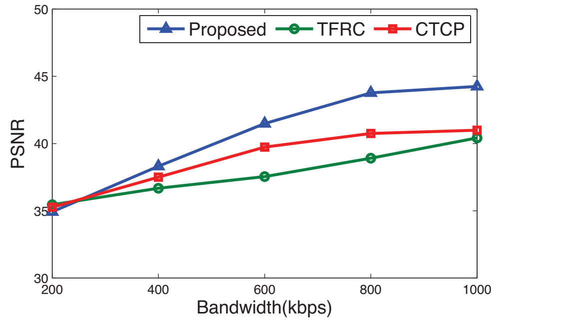


Fig. 8: Video quality performance under different bandwidth with loss rate=1% and latency=50ms.

V. CONCLUSION

In this paper, we propose a novel distributed delay-constrained rate control algorithm to meet the requirements of low latency, rate stability and high bandwidth utilization for real-time video streaming over unstable and error-prone wireless networks. We first introduce a fluid-queuing model and illustrate the relationship between video rate, queuing delay and network congestion. With this model, we perform congestion control according to shadow price and distortion weight, in order to control video rate in a distributed way and achieve distortion-fairness. By taking feedback queuing delay as input, we propose a control-theoretic approach to make the control more stable and agile, specifically a proportional controller is adopted to improve the performance. Additionally, we develop a fully-functional real-time wireless video streaming test-bed and conduct extensive experiments over it. We demonstrate that, compared with other algorithms, our proposal has better performance in bandwidth utilization, latency, and stability and achieves higher PSNR. And the multi-flow scenario of our scheme shows high fairness performance.

REFERENCES

[1] V. Jacobson, “Congestion avoidance and control,” in *ACM SIGCOMM Computer Communication Review*, vol. 18. ACM, 1988, pp. 314–329.
[2] S. O. L. Brakmo and L. Peterson, “Tcp vegas: New techniques for congestion detection and avoidance,” in *SIGCOMM, 1994. Proceedings. 1994 International Conference on*. ACM, 1994.
[3] L. X. I. R. Sangtae Ha, “Cubic: A new tcp-friendly high-speed tcp variant,” in *SIGOPS-OSR, July 2008*. ACM, 2008.
[4] D. Wu, Y. T. Hou, and Y.-Q. Zhang, “Transporting real-time video over the internet: Challenges and approaches,” *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1855–1877, 2000.
[5] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the internet,” *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 4, pp. 458–472, 1999.
[6] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, “Fast tcp: motivation, architecture, algorithms, performance,” *IEEE/ACM Transactions on Networking (ToN)*, vol. 14, no. 6, pp. 1246–1259, 2006.
[7] K. Kotla and A. N. Reddy, “Making a delay-based protocol adaptive to heterogeneous environments,” in *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*. IEEE, 2008, pp. 100–109.
[8] M. Podlesny and C. Williamson, “Providing fairness between tcp newreno and tcp vegas with rd network services,” in *Quality of Service (IWQoS), 2010 18th International Workshop on*. IEEE, 2010, pp. 1–9.
[9] K. T. J. Song, Q. Zhang, and M. Sridharan, “Compound tcp: A scalable and tcp-friendly congestion control for high-speed networks,” *Proceedings of PFLDnet 2006*, 2006.
[10] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, “Feedback control of dynamics systems,” *Pretince Hall Inc*, 2006.