

L-TFRC: AN END-TO-END CONGESTION CONTROL MECHANISM FOR VIDEO STREAMING OVER THE INTERNET

Zhen Li^{*}, Guobin Shen^{**}, Shipeng Li^{**}, and Edward J. Delp^{*}

Video and Image Processing Laboratory (VIPER)

^{*}School of Electrical and Computer Engineering, Purdue Univ., West Lafayette, IN 47907-1285, USA

^{**}Microsoft Research Asia, Beijing, 100080, China

ABSTRACT

Real-time multimedia applications over the Internet have posed a lot of challenges due to the lack of quality of service (QoS) guarantees, frequent fluctuations in channel bandwidth, and packet losses. To address these issues, a great deal of research has been done in both video coding and video transmission fields. In this paper we present a Logarithm-based TCP-Friendly Rate Control (L-TFRC) mechanism, which can estimate the available bandwidth more accurately and improve the smoothness of the multimedia streaming significantly. We also apply it to a Progressive Fine Granularity Scalable (PFGS)-based video streaming. Both simulations and experiments over the Internet confirm the performance of L-TFRC.

1. INTRODUCTION

It is well known that TCP have been one of the most important factors to the success of the Internet. The stability of the current Internet depends heavily on its end-to-end congestion control mechanisms, which use an Additive Increase Multiplicative Decrease (AIMD) algorithm [1]. However, the inherent conservative and abrupt change characteristics of AIMD often lead to the frequent rate fluctuations, which cause sharp reduction of visual quality for video streaming applications. Moreover, the current best-effort transmission over the Internet does not offer the necessary QoS guarantee normally required in conventional video streaming.

To address these challenges, extensive research work has been done in the past five years. In the field of video coding, much attention has been paid to rate scalability and error recovery. One of the examples is the Fine Granularity Scalability (FGS) profile [2] in MPEG-4, which enables a single bit stream to be decodable over a

wide rate range via bit-plane coding of the DCT residues. Despite its flexibility in bandwidth adaptation, FGS suffers a lot in coding efficiency. As a significant improvement, progressive FGS (PFGS) [3][4] presents a scheme to improve the coding efficiency by using higher quality references. In this paper we use PFGS as our basic video coding framework in the simulations and experiments.

Meanwhile, in the field of video transmission and networking, many TCP-like and TCP-friendly congestion control mechanisms have been developed, which aim to smooth the sending rate of video streaming while keeping the fairness to current TCP flows. Among all these proposals, TCP-Friendly Rate Control (TFRC) [5][6][7] is one of the most widely investigated due to its fairness and relative smoothness.

In this paper, we propose a Logarithm-based TCP-Friendly Rate Control (L-TFRC) mechanism as a nontrivial extension to TFRC. Based on L-TFRC, a family of congestion control mechanisms is built. Compared to other TCP-like or TCP-friendly mechanisms, the fairness level of L-TFRC over the TCP can be easily controlled and classified, which is a desirable feature for streaming applications. At the same time, the smoothness is much improved and the congestion avoidance property is still preserved. Besides the theoretical induction, extensive experiments both in the Network Simulator (NS2) and real Internet environment have been conducted to validate L-TFRC.

The rest of this paper is organized as follows. Section 2 presents the L-TFRC algorithm and compares it with TFRC. The congestion control mechanism family is also defined here. Extensive simulation results and performance comparisons on NS2 are presented in Section 3. In Section 4, we apply L-TFRC to PFGS on the streaming video applications. Section 5 concludes the whole paper with a brief remark.

[†] This work was done while Z. Li was a visiting student at Microsoft Research Asia.

2. LOGARITHM-BASED TFRC (L-TFRC)

As mentioned above, TCP is ill suited for real-time multimedia streaming application over the Internet due to its frequent halving the sending rate in response to a single congestion indication.

Such sharp reduction in sending rate can lead to obvious deteriorations in visual quality. Hence, many efforts have been made to smooth the rate control without changing the long-term overall throughput.

Most of current congestion control mechanisms generally fall into two categories: probe-based and equation-based approach. While the probe-based approach adjusts the sending rate in response to a single congestion indication, the equation-based approach uses a TCP throughput equation that explicitly estimates the sending rate as a function of recent loss event rate. Specifically, the TCP throughput model is given by the following formula [5]:

$$\lambda = \frac{1.22MTU}{RTT\sqrt{p}} \quad (\text{Bytes/s}), \quad (1)$$

where,

- λ Throughput of a TCP connection (in bytes/s);
- MTU Packet size used by the connection (in bytes);
- RTT Round-trip time of the connection (in second);
- p Packet loss ratio of the connection.

Among all the existing equation-based approaches, TFRC is one of the most deployable and successful. It tries to smooth the sending rate by giving a more accurate estimation of the TCP throughput [7] by considering the influence of timeout (RTO)

$$\lambda = \frac{MTU}{RTT\sqrt{\frac{2p}{3} + RTO(3\sqrt{\frac{3p}{8}})p(1 + 32p^2)}} \quad (\text{Bytes/s}) \quad (2)$$

Moreover, TFRC also uses the loss rate event, instead of single packet loss, to changes the control behavior of the sender and the receiver. These control approaches leads to a more smoother data stream compared to TCP and is hence also employed in the L-TFRC.

However, since generally $p \ll 1$ and $RTO = 4RTT$, the denominator in (2) is still at the order of $p^{1/2}$ asymptotically. Hence despite the fairness to the TCP throughput, what TFRC does not change is the inherent proportionality of throughput $\lambda \propto 1/\sqrt{p}$, which means that the overall rate change, i.e., the derivative of λ , is proportional to $1/(p\sqrt{p})$. Such proportionality makes rate change still over sensitive to packet losses, especially when packet loss ratio is small. Furthermore, according to previous research [8] and our simulations, TFRC also suffers from the following two problems:

- 1) Periodical Rate Fluctuation (PRF). Even if the channel bandwidth is a constant (denoted as BW), TFRC cannot stay at BW at its steady state. Instead, it still tries to increase the sending rate over BW , which unfortunately leads to a short-term congestion. As TFRC is still very sensitive in the low packet loss ratio environment, the sending rate is greatly reduced to avoid further congestion. Such abrupt reduction is quite similar to TCP's rate reduction, not as frequent though.
- 2) Share Starvation over Small narrow link (SSS). When transmitting through a narrow bottleneck link, TFRC's share is noticeably lower than the average competing TCP flows, and some times even drops to zero.

Hence, a natural improvement is to set the derivative of λ to $1/p$, instead of $1/(p\sqrt{p})$. This is the basic idea behind our new approach, referred as the Logarithm-based TFRC (L-TFRC). We can define a family of throughput equations as a polynomial in natural logarithm $\log(p)$, which describes L-TFRC.

$$\lambda = \frac{MTU}{RTT} \times \left[\sum_{n=0}^{\infty} a_n \log^n(p) \right] \quad (3)$$

For example, the 1st-order case is

$$\lambda = \frac{MTU}{RTT} \times [a_0 + a_1 \log(p)] \quad (4)$$

Although higher orders of the L-TFRC can model the throughput more accurately, we restrict our discussion to the 1st-order L-TFRC due to its simplicity.

For the practical video streaming transmission over the Internet, we can fix $MTU = 1K$ bytes, $RTT = 100ms$ and assume that there is no effective transmission, i.e. $\lambda = 0$, when $p > 60\%$. Next we set the desired throughput at $p = 0.1\%$ and solve the equation (4). Hence, the 1st-order throughput formula becomes

$$\lambda = \frac{k}{RTT} \times [-0.25 - 0.5 \times \log(p)] \quad (\text{K Bytes/s}) \quad (5)$$

where k is a control parameter..

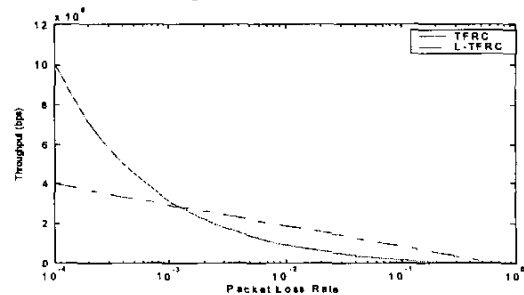


Figure 1: Comparison of TFRC and L-TFRC estimated throughputs.

Figure 1 shows the comparison between TFRC and 1st-order L-TFRC where $k = 11$. Compared with TFRC,

the sending rate of L-TFRC changes slower in low packet loss ratio case ($p \leq 0.05$) and faster in high packet loss ratio case ($p > 0.05$). This is obviously a desirable result for smooth congestion control.

As far as the fairness is concerned, different classes of fairness are defined by setting different k with respect to the requirements of the application. Currently there are three classes of fairness defined in the 1st-order L-TFRC:

- 1) TCP-friendly. When k is small (e.g. $k = 8$), the L-TFRC flows consistently get bandwidth no more than the TCP flow;
- 2) Bounded-friendly. Streaming applications may argue that it is actually unfair for a long-lived video stream to get only as much share as those short-lived TCP flows. In this case, we set k to be medium (e.g. $k = 11$), where an L-TFRC flow can get more shares than average TCP flows, but the overall L-TFRC shares is bounded by $c * \text{TCP share}$, where c is a weight set by the applications;
- 3) Streaming-oriented. In some streaming applications, what people care is to maximize the throughput of the streaming in order to exploit best visual quality out of current network resources. So k can be set to be large (e.g. $k = 15$) for this purpose. Then L-TFRC tries to seek the maximum available bandwidth, i.e., the bottleneck bandwidth. Hence, L-TFRC is also capable of estimating the bottleneck bandwidth.

3. SIMULATION RESULTS USING NS2

Extensive simulations have been performed on NS2 to evaluate the performance of L-TFRC.

The main objectives of our simulations here are to evaluate the networking property of L-TFRC, including the aggressiveness, responsiveness, smoothness and fairness. In all comparison simulations, we run TFRC and L-TFRC ($k = 8$) at the same time. And the classical dumb bell topology [6] is employed here.

5.1. Aggressiveness and Responsiveness

Aggressiveness and Responsiveness are defined to evaluate how fast the protocol can adjust to the change of available bandwidth (ABW). ON-OFF background traffic is used.

Figure 2 shows L-TFRC performs nearly the same at the very beginning and slightly better after that. This makes sense since L-TFRC still employs most of the control approaches in the slow start phase.

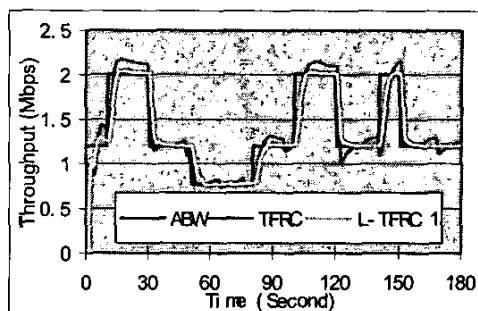


Figure 2 Aggressiveness and Responsiveness Test

5.2. Smoothness

Smoothness is defined in terms of the variance of the sending rate. While TFRC claims a significant improvement over TCP in smoothness, its inherent $1/\sqrt{p}$ proportionality still introduces frequent small variance, as shown in Fig. 2. Fig. 3 also demonstrates the Periodic Rate Fluctuation (PRF) effect in the steady state. L-TFRC demonstrates obvious improvement in smoothness.

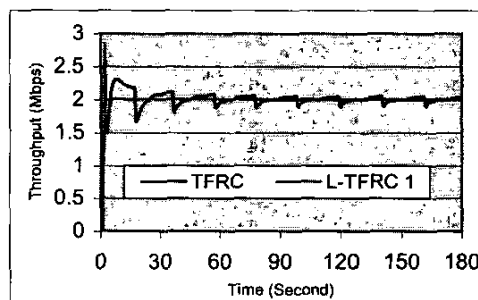


Figure 3 Smoothness and PRF effect

5.3. Fairness

Fairness is defined as the overall throughput ratio over a long time period.

One of the best properties of TFRC is its long-term convergence in throughput with respect to TCP flows. However it suffers from the share starvation (SSS) effect when competing with multiple TCP flows over a link with narrow bottleneck. In this simulation, a TFRC (or L-TFRC) flow is competing with 16 TCP flows with the bottleneck bandwidth = 5Mbps.

The results in Fig. 4 show that TFRC gets much less share than the average TCP flows and sometimes even drops to zero. On the other hand, L-TFRC still keeps relatively TCP-friendly. And it also overcomes SSS effect.

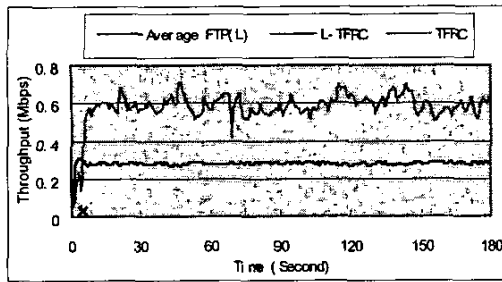


Figure 4 Fairness and SSS effect

All of these simulations are running with DropTail queue dynamics. Same observations are also found in simulations with Random Early Detection (RED).

4. EXPERIMENTAL RESULTS ON STREAMING VIDEO OVER THE INTERNET

We have also tested L-TFRC across the Internet with the help of The Cloud, a commercial WAN emulator, to shape the channel bandwidth to 1Mbps. The results further prove that L-TFRC outperforms TCP, TFRC and other congestion control mechanisms over the Internet. We use L-TFRC to control the sending rate of the video stream, which is coded with PFGS. A major advantage of the PFGS is its bandwidth adaptation capability that enables decoding at any bit rate and enhancing the visual quality with every extra bit received. Hence an accurate and estimation of the sending rate can greatly improve the perceived video quality.

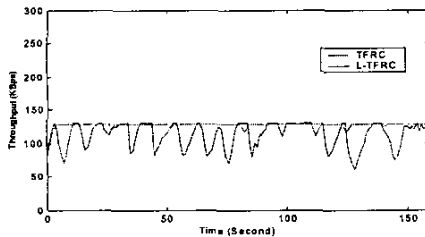


Figure 5 Throughput of the Video Streaming

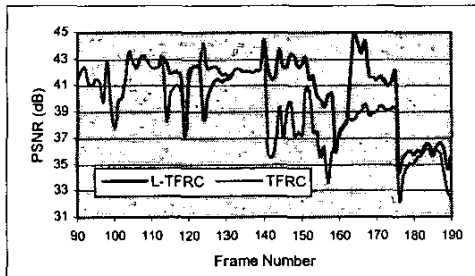


Figure 6 PSNR comparison (Y Component of Christina_sif)

Fig. 5 shows the throughput comparison between TFRC and L-TFRC. As we have mentioned above, there are periodic fluctuations and small variations in TFRC, which leads to a performance loss up to 4dB compared with L-TFRC, as shown in Fig. 6.

5. CONCLUSIONS

In this paper, a new end-to-end congestion control mechanism is proposed based on the analysis of various approaches to smooth the sending rate and avoid congestion collapse. The new Logarithm-based technique sets up a family of models, which have led to significant improvement from both networking and video streaming point of view.

ACKNOWLEDGEMENT

The authors would like to thank Dr. F. Wu, Dr. Q. Zhang, and other colleagues in the Internet Media Group and Wireless Networking Group at Microsoft Research China for many beneficial discussions and encouragements.

REFERENCES

- [1]. V. Jacobson, "Congestion Control Avoidance and Control," *Proc. Of ACM SIGCOMM 1988*, pp. 314-329, ACM Press, NY, USA, 1988.
- [2]. W. Li, "Overview of Fine Granularity Scalability for Internet Video," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 11, pp. 301-317, March 2001.
- [3]. F. Wu, S. Li and Y. Zhang, "A Framework for Efficient Progressive Fine Granularity Scalable Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 11, pp. 332-344, March 2001.
- [4]. Q. Zhang, W. Zhu and Y. Zhang, "Resource Allocation for Multimedia Streaming over the Internet," *IEEE Trans. Multimedia*, Vol. 3, pp. 339-355, Sept. 2001
- [5]. S. Floyd and M. Handley, "Equation-Based Congestion Control for Unicast Applications," *Proc. Of ACM SIGCOMM 2000*, pp. 43-56, ACM Press, NY, USA, 2000.
- [6]. J. Padhye and V. Firoiu, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," *Proc. Of ACM SIGCOMM 1998*, pp. 303-314, ACM Press, NY, USA, 1998.
- [7]. M. Handley, J. Padhye and S. Floyd, "TCP Friendly Rate Control (TFRC): Protocol Specification," *Internet Engineering Task Force (IETF) Internet Draft draft-ietf-tsvwg-tfrc-02*, <http://www.icir.org/tfrc/>, work in progress, May 2001.
- [8]. I. Rhee, V. Ozdemir and Y. Yi, "Tear: TCP Emulation at Receiver-Flow Control for Multimedia Streaming," http://www.csc.ncsu.edu/eos/users/irhee/WWW/export/tear_page/techreport/tearf.html, *NCSU Technical Report*, Apr. 2000.