

A New TCP-Friendly Rate Control Algorithm for Scalable Video Streams

Jinyao Yan^{1,2}, Martin May¹, Kostas Katrinis¹, and Bernhard Plattner¹

¹ Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, ETH Zurich, 8092, Switzerland

² Communication University of China, Beijing, 100024, China
{jinyao,maym, katrinis,plattner}@tik.ee.ethz.ch

Abstract. This paper presents Media- and TCP-Friendly Rate Control Algorithm (i.e. MTFRC) for scalable video streams. We develop a two-timescale approach to satisfy media friendliness in the short-term, while maintaining TCP friendliness in the long-term. We evaluate MTFRC with five criteria, namely TCP-fairness, responsiveness, aggressiveness, video quality and smoothness of sending rate. Our simulation results manifest that MTFRC outperforms the TCP-Friendly Rate Control Protocol (TFRC) and improves the overall video quality.

1 Introduction

TCP uses an additive increase multiplicative decrease mechanism (AIMD) to detect additional available bandwidth and to react to congestion in the steady state. However, it is not well suited for the growing number of media streaming applications. As a result, new TCP-friendly congestion control protocols such as TFRC [1] have been proposed for streaming applications while competing TCP flows friendly. Still, there is a fundamental shortcoming of the above protocols: they target at optimizing the utilization of network resources *without taking the resulting application quality into account*. Even though TFRC does smooth the rate variability, this does not guarantee maximum media quality at the users.

Given a system with scalable video streaming sources/receivers and links, the challenge is to allocate a rate value to each source that is both media- and TCP-friendly. A rate control algorithm is termed media-friendly, if it optimizes the overall quality of the various video streams in the network.

The main contribution of this paper is the MTFRC algorithm that optimizes the overall video quality for scalable streams, while being TCP-friendly in the long term.

2 Transforming MTFRC into a TCP-friendly Algorithm

In our previous work [2], we introduced the video quality function for scalable video as the utility function in a utility-based network model and used it to maximize video

quality. Let $I_s=[m_s, M_s]$ denote the feasible range of rate x_s for a scalable video source s . We derived the optimal throughput response function $x_s(p)$ that maximizes the overall video quality of scalable video streams (a , b and c are parameters to be evaluated empirically by the video stream quality):

$$x_s(p)=[(\sqrt{\log_2^{(-p/a \ln 2)}/a-c/a+b^2/4a^2-b/2a})^2]_{m_s}^{M_s} \quad (1)$$

To behave in a TCP-friendly manner, the TFRC protocol adjusts its sending rate based on the TCP throughput function in [3].

Let the throughput response function of MTFRC Eq. (1) be denoted by $X_{MTFRC}(p)$; the throughput response function of TFRC be denoted by $X_{TFRC}(p)$. When two traffic flows compete for resources at the same network bottleneck, both flows will obtain a fair share of the bandwidth, if and only if both flows apply similar long-term response functions. In order to meet TCP friendliness, we modify the MTFRC algorithm in a two-timescale approach as follows:

1. The sender (or receiver) estimates two congestion control parameters: the long-term packet loss rate p_L , and the short-term packet loss rate p_s .
2. The sender calculates its long-term rate using the long-term packet loss rate p_L according to the TFRC response function. However, the sender reacts to short-term congestion given by p_s according to the MTFRC response function (1).

In particular, the MTFRC algorithm adjusts the parameters a , b , c in Eq. (1) to achieve the long-term rate $X_{TFRC}(p_L)=X_{MTFRC}(p_L)$, the sending rate for next period is $X_{MTFRC}(p_s)$ using updated values for a , b , or c .

3 Results and Analysis

In this section, we present the results obtained from simulations we conducted to evaluate the behavior of the MTFRC algorithm against TFRC. We used the ns2 network simulator [5] for our experiments. The scenario implemented throughout our simulations is illustrated in Figure 1. Physical links are marked with propagation delay and capacity bandwidth values (default is 10Mbps/10ms). We considered the following evaluation criteria: 1) TCP-friendliness of competing flows 2) Responsiveness: reaction time of the protocol, when decreasing the sending rate in case of severe network congestion 3) Aggressiveness: acceleration of protocol sending rate after a congestion incidence 4) Smoothness: rate variations over time for a particular flow 5) Overall video quality.

We compare the characteristics of our MTFRC algorithm to those of TFRC(8) (history size 8) and TFRC(128) (history size 128). For MTFRC, we used a long-term history size of $N_L=256$ and a short-term history size of $N_s=8$ (denoted as MTFRC(256, 8)).

Figure 2, Fig. 3, and Fig. 4 present the bit rate evolution of the two protocols in the event of severe congestion and sudden increase of the available bandwidth. We use a CBR source (sending rate: 1Mbps) on Conn.4 to decrease the available bandwidth at the 60th second (severe network congestion) and to increase the available bandwidth at the 120th second.

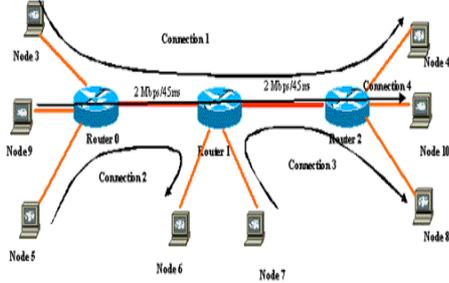


Fig. 1. Network Topology

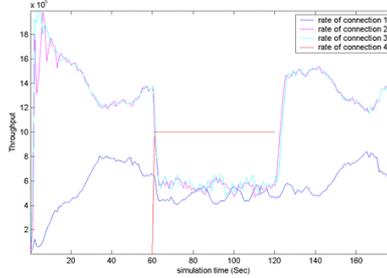


Fig. 2. Bit rate traces with TFRC(8)

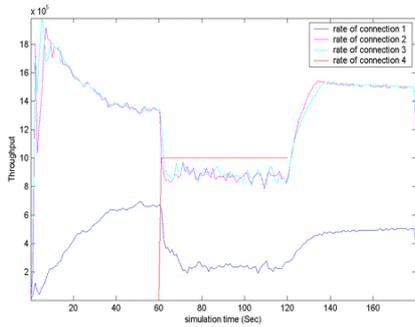


Fig. 3. Bit rate traces with MTFRC(256,9)

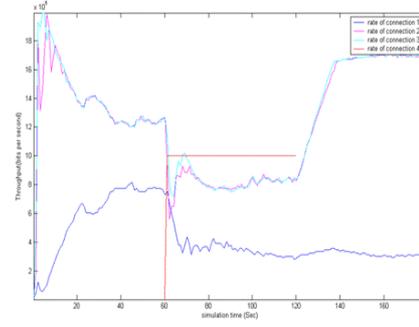


Fig. 4. Bit rate traces with TFRC(128)

Overall video quality We apply the obtained sending rate traces of the simulated network connections to FGS video scaling [4] and obtain the video quality (luminance component) that the MTFRC flow would exhibit. The PSNR (Peak Signal-to-Noise Ratio) values of MTFRC and those achieved by TFRC for the “Highway” test stream are depicted in Table 1.

Due to space limitation, we omit the detailed evaluation of TCP-friendliness, responsiveness, aggressiveness, and smoothness for the MTFRC algorithm. In table 2, we summarize the findings of our evaluation. Overall, MTFRC algorithm offers a better tradeoff between network utilization and media quality.

4 Conclusion and future work

In this paper, we studied a both Media- and TCP-friendly rate control algorithm for scalable video streams. We introduced a two-timescale approach of rate averages (long-term and short-term) to the MTFRC algorithm to satisfy both media and TCP friendliness. Using simulations, we showed that our algorithm is TCP-friendly and

Jinyao Yan^{1,2}, Martin May¹, Kostas Katrinis¹, and Bernhard Plattner¹

achieves an improved network-friendly behavior (evaluated in terms of responsiveness, aggressiveness and fairness). Additionally, MTFRC exhibited superior media friendliness than that of TFRC in terms of smoothness and video quality. Our ongoing work will focus on investigating the behavior of MTFRC algorithm in the open Internet.

Table 1. Comparison of resulting quality

	Stream on conn.1	Stream on conn.2	Stream on conn.3
PSNR and its variance with MTFRC	27.9 $\sigma = 0.020$	32.35 $\sigma = 0.0099$	32.36 $\sigma = 0.010$
PSNR and its variance with TFRC	29.39 $\sigma = 0.018$	30.70 $\sigma = 0.020$	30.69 $\sigma = 0.020$
Total quality gain	Increase of PSNR over MTFRC=1.8, Decrease of its variance=0.018		

Table 2. Summary of the simulation results

	MTFRC(256,8)	TFRC(8)	TFRC(128)
TCP Fairness	All three algorithms are TCP-friendly		
Smoothness of bit rate	Smother than TFRC(8), roughly same as TFRC (128)	Smooth bit rate	Smother than TFRC(8) similar to MTFRC
Responsiveness	Medium	High	Low
Aggressiveness	Faster than TFRC (128), slower than TFRC(8)	Faster than TFRC (128) and MTFRC	Slower than TFRC(8) and MTFRC
Video quality	Better quality than TFRC (8) and TFRC(128)	Similar quality to TFRC(128), worse than MTFRC	Similar quality to TFRC(8), worse than MTFRC

References

1. S. Floyd, M. Handley, and J. Padhye, "Equation-Based Congestion Control for Unicast Applications", ACM SIGCOMM, September 2000.
2. Jinyao Yan, Kostas Katrinis, Martin May, Bernhard Plattner, "Optimizing Rate Control for Multiple Fine-granular Scalable Video Streams"(short paper), IEEE ICNP 2004, Berlin, Germany Oct. 2004.
3. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. "Modeling TCP Throughput: A Simple Model and its Empirical Validation". SIGCOMM Symposium on Communications Architectures and Protocols, Aug. 1998.
4. W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard", IEEE Trans.on Circuits and Systems for Video Technology, Vol.11, No.3, March 2001, pp301-317
5. ns-2 network simulator, <http://www.isi.edu/nsnam/ns/>