Congestion Control for Multimedia Streaming with Self-Limiting Sources

Josh Helzer (student)
University of Nebraska–Lincoln
Lincoln, Nebraska 68588-0115
jhelzer@cse.unl.edu

Lisong Xu (faculty)
University of Nebraska–Lincoln
Lincoln, Nebraska 68588-0115
xu@cse.unl.edu

1. INTRODUCTION

In the past few years we have witnessed an explosive growth in the usage of media streaming applications. For instance, according to a recent study by AccuStream IMedia Research, 14.2 billion video streams were served in 2004, an 80% increase compared to 2003, and the number is forecast to reach 21 billion in 2005. The rapid growth in the usage of streaming media over the Internet has heightened [3] the need for a congestion control protocol suitable for streaming media. Among the proposed streaming-media congestion control protocols, TCP-Friendly Rate Control (TFRC) [2] is one of the promising solutions, and is currently being adopted in several Internet standards. TFRC maintains an equal or lesser average sending rate as competing TCP connections (referred to as *TCP Friendly*), while providing a relatively smooth sending rate to help packets to meet the real-time constraints required by streaming media.

Media sources can be divided into two classes: *greedy* or *self-limiting sources*. The former, including pre-recorded media sources, send data as fast as allowed by the congestion control protocol, whereas the latter, including live media sources, cannot transmit data faster than their encoding rate [5]. For a self-limiting source, the ideal sending rate should be the minimum of its encoding rate and the network fair share rate (i.e., the average sending rate of competing TCP connections). If a congestion control protocol can provide this ideal rate for a self-limiting source, it is *Self-Limiting Source Friendly*. However, it has been reported [5] that TFRC is not self-limiting source friendly, since the average sending rate achieved by TFRC may be much less than the ideal one.

We propose Equation-based Leaky-Bucket-regulated congEstion control (ELBE), a new congestion control protocol for the streaming of self-limiting media sources over the Internet. Our preliminary simulation results show that ELBE is both TCP friendly and self-limiting source friendly.

2. SELF-LIMITING SOURCE UNFRIENDLINESS OF TFRC

For a self-limiting media source TFRC maintains a transmit buffer [6] which is filled at the media encoding rate and drained at the average TCP sending rate if the buffer is not empty. When there is a dramatic change in the average TCP sending rate, the media source may switch its average encoding rate to a higher or lower one accordingly. To simplify the description, we assume that the average TCP sending rate is constant, and also that the average media encoding rate is constant. The instantaneous media encoding rate can still vary, for example, when silence or stillness suppression is employed.

Let S_{avg} denote the average sending rate that packets leave the buffer and enter the network, and E_{avg} denote the average media encoding rates. Let TCP_{avg} and TCP_{peak} denote the average and peak TCP sending rates, respectively. A self-limiting source friendly protocol should achieve $S_{avg} = \min(E_{avg},\ \text{TCP}_{avg})$. However, since TFRC limits the instantaneous sending rate to be no more than TCP_{avg}, we have $S_{avg} \leq \min(E_{avg},\ \text{TCP}_{avg})$. In particular, as long as an instantaneous encoding rate fluctuates around TCP_{avg}, we have $S_{avg} < \min(E_{avg},\ \text{TCP}_{avg})$. Furthermore, when the instantaneous encoding rate is significantly larger than TCP_{avg}, some packets will be delayed in the buffer, and dropped if the buffer has a small capacity in order to maintain a small packet latency. In this case, S_{avg} may be much less than $\min(E_{avg},\ \text{TCP}_{avg})$. Therefore, TFRC is not self-limiting source friendly.

3. DUAL LEAKY BUCKET ALGORITHM OF ELBE

ELBE regulates the traffic from a media source by the dual leaky bucket algorithm, as shown in Figure 1. The dual leaky bucket is usually used in networks with Quality of Service (QoS) support, such as ATM, Diffserv, and Intserv networks, here we use it for media streaming over the best-effort Internet. For simplicity, below we assume that all packets have the same size, and that one packet removes one token. The actual ELBE protocol will use a fluid dual leaky bucket algorithm to accommodate media sources with variable packet sizes.

ELBE sets $C_1 > 1$, $R_1 = \text{TCP}_{avg}$, $C_2 = 1$, and $R_2 = \text{TCP}_{peak}$. With this setting, leaky bucket 1 limits average sending rate S_{avg} to be no more than TCP_{avg}. Leaky bucket 2 allows an instantaneous sending rate to be larger

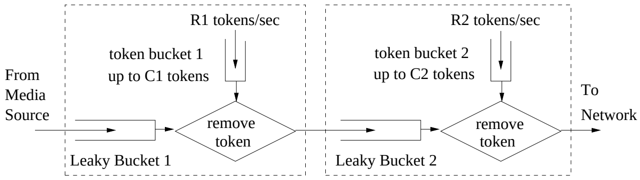

Figure 1: ELBE uses the dual leaky bucket algorithm to regulate the sending rate of a self-limiting media source.

than TCP_{avg} but up to TCP_{peak}, within a short interval whose duration is controlled by the value of C_1. Therefore, ELBE can achieve $S_{avg} \approx \min(E_{avg}, \text{TCP}_{avg})$, as long as the encoding rate of a media source has a relatively small fluctuation range compared to the sending rate of a competing TCP flow.

4. ELBE'S PARAMETERS

The average TCP sending rate TCP_{avg} can be estimated by using the TCP throughput function [2]. Based on a simple deterministic TCP model, we can approximate TCP_{peak} by $\frac{4}{3}\text{TCP}_{avg}$. In order to reduce the impact of sudden media rate increase on other competing traffic, we set R_2 to a value slightly less than TCP_{peak} (e.g. 1.2TCP_{avg}) in the simulation. C_1 is set to half of the congestion window of a competing TCP flow, so that a media source can send packets at R_2 for half a round-trip time.

5. SIMULATION RESULTS

The performance of TFRC and ELBE with self-limiting sources is evaluated by using NS-2. Our setup uses a typical dumbbell topology with drop tail routers. The bandwidth and one-way delay of the bottleneck link are set to 20Mbps and 50ms, respectively. We simulate five types of connections: UDP, TFRC, ELBE, long-lived TCP, and web traffic. The sources of UDP, TFRC, and ELBE connections are self-limiting media sources, which generate traffic by using an MPEG trace of the movie *Star Wars* [1]. Since every frame in the MPEG trace is divided into 200-byte packets, we set the size of all packets in the simulation to 200 bytes. For both TFRC and ELBE, the capacity of the transmit buffer is set to be five times of the bandwidth-delay product. The sources of long-lived TCP connections are sending data as fast as TCP allows. Web traffic is simulated to increase traffic dynamics and remove phase effect.

Figure 2 shows the average sending rate of UDP, TFRC, ELBE, and long-lived TCP connections, as we increase the total number of TCP connections (i.e., network load). As we expected, the sending rate of UDP is almost a constant around 400Kbps, which is the average media encoding rate. We can see that the average sending rate of ELBE is very close to the minimum of the average encoding rate and the average TCP sending rate. However, the average sending rate of TFRC is much less than the minimum. Note that, due to the limitations [7] of equation-based congestion control, there are always some gaps between the actual TFRC sending rate and its desired rate. The same holds for ELBE.

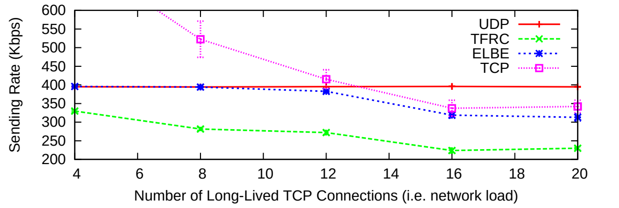

Figure 2: The average sending rate of ELBE is very close to the minimum of the average media encoding rate (i.e UDP rate) and the network fair share rate (i.e. TCP rate).

6. CONCLUSIONS

We have proposed a new congestion control protocol, ELBE, for the streaming of self-limiting media sources over the Internet, and have shown it is both TCP friendly and self-limiting source friendly. There are two other recently proposed protocols for self-limiting sources: MARC [8] and MFRC [4]. None of them uses the peak TCP sending rate as ELBE does. Currently, we are conducting more experiments to compare their performance with ELBE.

See `http://cse.unl.edu/~xu/streaming/index.html` for further results of our work.

7. ACKNOWLEDGMENTS

The work reported in this abstract is supported in part by Maude Hammond Fling Faculty Research Fellowship and Layman Fund Award.

8. REFERENCES

[1] MPEG Trace of the Movie Star Wars for NS-2. *http://www.research.att.com/~breslau/vint/trace.html.*

[2] S. Floyd, M. Handley, J. Padhye, and J. Widmer. TCP friendly rate control (TFRC): Protocol specification. *RFC 3448*, January 2003.

[3] S. Floyd and J. Kempf. IAB concerns regarding congestion control for voice traffic in the Internet. *RFC 3714*, March 2004.

[4] T. Phelan. Media friendly rate control (MFRC). *IETF Internet draft*, July 2004.

[5] T. Phelan. TFRC with self-limiting sources. *available at http://www.phelan-4.com/dccp/*, October 2004.

[6] T. Phelan. Strategies for streaming media applications using TCP-friendly rate control. *IETF Internet draft*, July 2005.

[7] I. Rhee and L. Xu. Limitations in equation-based congestion control. In *Proceedings of ACM SIGCOMM*, pages 49–60, Philadelphia, PA, August 2005.

[8] Z. Wang, S. Banerjee, and S. Jamin. Media-aware rate control. *Technical Report, Department of EECS, Univerity of Michigan*, November 2004.