Multimedia Streaming via TCP: An Analytic Performance Study

Bing Wang, Jim Kurose, Prashant Shenoy, Don Towsley

Department of Computer Science
University of Massachusetts, Amherst, MA 01003

Categories and Subject Descriptors: C.2.2 [Network protocols]: Application (video streaming)

General Terms: Performance.

Keywords: Performance modeling, Multimedia streaming.

1. INTRODUCTION

TCP is widely used in commercial streaming systems. For instance, Real Media and Windows Media, the two dominant streaming media products, both support TCP streaming. Furthermore, a recent measurement study has shown that a significant fraction of commercial streaming traffic uses TCP [1]. This study analyzed 4.5 million session-level logs for two commercial streaming servers over a four month period and found that 72% and 75% of the on-demand and live streaming traffic, respectively, used TCP.

Motivated by the wide use of TCP streaming in commercial systems, we seek to answer the following question: *Under what circumstances can TCP streaming provide satisfactory performance?* To answer this question, we study a baseline streaming scheme which uses TCP directly for streaming, henceforth referred to as *direct TCP streaming*. We study the performance of direct TCP streaming using analytical models. Our models enable us to systematically investigate the performance of TCP streaming under various conditions, a task that is difficult when using empirical measurements or simulation alone.

We build upon the TCP model in [2, 3] to develop discrete-time Markov models for *live* and *stored* video streaming. The models are validated using *ns* simulation and Internet experiments. Using the models, we explore the parameter space (i.e., loss rate, round trip time and timeout value in TCP as well as video playback rate) to provide guidelines as to when direct TCP streaming leads to satisfactory performance. Our results show that direct TCP streaming generally provides good performance when the achievable TCP throughput is roughly twice the the video bitrate, with only a few seconds of startup delay.

Our study has the following implication. Studies have shown that a large fraction of streaming video clips on the Internet today are encoded at bit rates below 300 Kbps, e.g., [4]. Moreover, most DSL and cable modem connections support download rates of 750 Kbps - 1 Mbps. In the situations where the available end-end bandwidth is only constrained by the last-mile access link, our performance study indicates that direct TCP streaming may be adequate for many broadband users.

SIGMETRICS/Performance'04, June 12–16, 2004, New York, NY, USA.
ACM 1-58113-664-1/04/0006.

2. MODELS FOR STREAMING USING TCP

In this work, we assume that the average TCP throughput is no less than the video bitrate. This guarantees that, on average, the throughput provided by TCP satisfies the requirement for streaming the video. However, fluctuations in the instantaneous TCP throughput can still lead to significant late packet arrivals. The client allows a startup delay on the order of seconds, which is a common practice in commercial streaming products. All the packets arriving earlier than their playback times are stored at the client's local buffer. We assume this local buffer is sufficiently large so that no packet loss is caused by buffer overflow at the client side.

Measurement studies show that most of the videos in the Internet are CBR (constant bit rate) videos [4]. We therefore consider a CBR video. The playback rate of the video is μ packets per second. For simplicity, all packets are assumed to be of the same size. For analytical tractability, we assume continuous playback at the client. A packet arriving later than its playback time is referred to as a *late packet*. We assume a late packet leads to a glitch during the playback and use the *fraction of late packets*, i.e., the probability that a packet is late, to measure the performance.

We study two forms of streaming that correspond respectively to live and stored video streaming in practice. In live streaming, the server generates video content in real time and is only able to transmit the content that has already been generated. The transmission is therefore constrained by the generation rate of the video at the application level. Hence we refer to this form of streaming as *constrained streaming*. For a stored video, we assume the server transmits the video as fast as allowed by the achievable TCP throughput in order to fully utilize the TCP throughput. We refer to this form of streaming as *unconstrained streaming* since the application does not impose any constraint on the transmission.

We now present discrete-time Markov models for constrained and unconstrained streaming. Each time unit is denoted as a "round", corresponding to a RTT of length as R time units. We consider a video whose length is L rounds. The playback rate of the video is μR packets per round. Let f denote the fraction of late packets during the playback of the video. Our goal is to derive models for determining f as a function of various system parameters (including the loss rate, RTT, timeout value in the TCP flow and the video playback rate).

Constrained streaming can be modeled as a producer-consumer problem. The producer produces packets according to the mechanisms of TCP and stores the packets in a buffer. For a startup

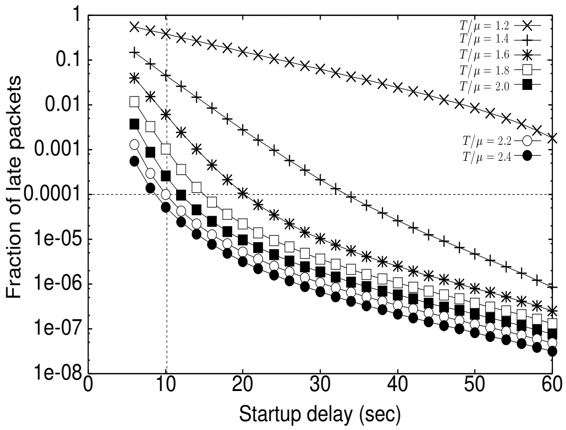

Figure 1: The fraction of late packets versus the startup delay for $p = 0.02$, $T_O = 4$ and $\mu = 25$ packets per second.

delay of τ seconds, the consumer starts to consume the packets in the buffer from time τ at a constant rate of μR packets per round. At any time, the number of packets in the buffer is no more than N_{max}, $N_{max} = \mu\tau$, because of the constraint of video generation rate [5]. To satisfy this constraint, the producer stops producing packets when there are N_{max} packets in the buffer.

In unconstrained streaming, the fraction of late packets depends heavily on the position of the round. This is because, under the assumption that the average TCP throughput is higher than the video bitrate, as the length of the video goes to infinity, the number of early packets approaches infinity and, hence, the fraction of late packets approaches 0. The fraction of late packets in the steady state (when the video is regarded as infinitely long) is thus trivial (equal to 0). To obtain the fraction of late packets over a finite video, we therefore resort to transient analysis [5].

We solve the models for constrained and unconstrained streaming numerically. The models are validated using extensive *ns* simulation and experiments over the Internet. Our validation showed that the performance predicted by the models are accurate over a wide range of settings [5].

3. EXPLORING PARAMETER SPACE

We vary the model parameters in constrained and unconstrained streaming to study the impacts of these parameters on performance. In doing so, we provide guidelines as to when TCP streaming leads to satisfactory performance.

We set the values of the TCP parameters to represent a wide range of scenarios. The loss rate, p, is varied in the range of 0.004 to 0.04. The RTT, R, is varied from 40 ms to 300 ms. Previous work shows that the median RTT between two sites on the same coast in the US is 50 ms, while the median RTT between west-coast and east-coast sites is 100 ms [6]. The timeout value is represented as a multiple of the RTT. We denote this multiple as T_O and vary T_O from 1 to 4. We denote the achievable TCP throughput as T packets per second. Then T/μ represents how much the achievable TCP throughput is higher than the video playback rate.

Studies show that the video quality drops sharply when the packet loss ratio exceeds 10^{-4} (e.g., [7]). Consequently, we assume that the performance of direct TCP streaming is satisfactory when the faction of late packets is below 10^{-4} for a startup delay of around 10 seconds. For constrained streaming, the video is assumed sufficiently long (hundreds of seconds) so that stationary analysis can be used to obtain the fraction of late packets. For unconstrained streaming, the video is assumed to be 80 seconds.

We first fix the playback rate of the video to be 25 packets per second. For $p = 0.004, 0.02, 0.04$ and T_O from 1 to 4, we vary the value of RTT such that T/μ ranges from 1.2 to 2.4. In constrained streaming, the performance improves dramatically as T/μ increases from 1.2 to 1.6. One example when $p = 0.02$ and $T_O = 4$ is shown in Fig. 1. Under various settings, the performance becomes satisfactory when the achievable TCP throughput is roughly twice the video bitrate. In unconstrained streaming, the performance becomes satisfactory when the achievable TCP throughput is roughly 1.8 times of the video bitrate.

We next set the value of RTT to 50, 100, 200 or 300 ms. For $p = 0.004, 0.02, 0.04$ and T_O from 1 to 4, we vary the playback rate of the video such that T/μ ranges from 1.2 to 2.4. In constrained streaming, under relatively short RTT, i.e., $R = 50$ ms (corresponding roughly to same coast in US) or 100 ms (roughly between two coasts in US), for various settings, the performance is generally good when the achievable TCP throughput is twice the video bitrate. However, for large RTT ($R = 300$ ms), high loss rate ($p = 0.04$) and timeout value ($T_O = 4$), a startup delay of 60 seconds is required in order for the fraction of late packets to be below 10^{-4} for $T/\mu = 2$. This indicates that, for large RTT and high loss rate, either a large T/μ or a long startup delay is required for a good performance when using TCP for streaming. We observe similar results in unconstrained streaming.

In summary, for both constrained and unconstrained streaming, the performance improves as the value of T/μ increases. However, the performance is not solely determined by T/μ but is sensitive to the values of the various parameters in the models. For both constrained and unconstrained streaming, the performance is generally good when the achievable TCP throughput is roughly twice the video bitrate, with only a few seconds of startup delay.

4. ACKNOWLEDGMENT

We thank E. de Souza e Silva and D. R. Figueiredo for many helps with numerically solving Markov models. This research was supported in part by the National Science Foundation under NSF grants ANI-9977635.

5. REFERENCES

[1] J. van der Merwe, S. Sen, and C. Kalmanek, "Streaming video traffic: Characterization and network impact," in *Proceedings of the Seventh International Web Content Caching and Distribution Workshop*, August 2002.

[2] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Tech. Rep. 99-02, Department of Computer Science, University of Massachusetts, Amherst, 1999.

[3] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley, "On the autocorrelation structure of TCP traffic," *Computer Networks Journal Special Issue on Advances in Modeling and Engineering of Long-Range Dependent Traffic*, 2002.

[4] M. Li, M. Claypool, R. Kinicki, and J. Nichols, "Characteristics of streaming media stored on the Internet," Tech. Rep. WPI-CS-TR-03-18, CS Department, Worcester Polytechnic Institute, May 2003.

[5] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia Streaming via TCP: An Analytic Performance Study," Tech. Rep., Department of Computer Science, University of Massachusetts, Amherst, 2004.

[6] B. Huffaker, M. Fomenkov, D. Moore, and K. Claffy, "Macroscopic analyses of the infrastructure: Measurement and visualization of Internet connectivity and performance," in *A Workshop on passive and active measurements*, (Amsterdam), April 2001.

[7] O. Verscheure, P. Frossard, and M. Hamdi, "MPEG-2 video services over packet networks: Joint effect of encoding rate and data loss on user-oriented QoS," in *Proc. of NOSSDAV*, July 1998.