The Case for Streaming Multimedia with TCP

Charles Krasic, Kang Li, and Jonathan Walpole*

Oregon Graduate Institute, Beaverton OR 97206, USA,
{krasic,kangli,walpole}@cse.ogi.edu,
http://www.cse.ogi.edu/sysl/

Abstract. In this paper, we revisit and challenge the dogma that TCP is an undesirable choice for streaming multimedia, video in particular. For some time, the common view held that neither TCP nor UDP, the Internet's main transport protocols, are adequate for video applications. UDP's service model doesn't provide enough support to the application while TCP's provides too much. Consequently, numerous research works proposed new transport protocols with alternate service-models as more suitable for video. For example, such service models might provide higher reliability than UDP but not the full-reliability of TCP. More recently, study of Internet dynamics has shown that TCP's stature as the predominant protocol persists. Through some combination of accident and design, TCP's congestion avoidance mechanism seems essential to the Internet's scalability and stability. Research on modeling TCP dynamics in order to effectively define the notion of TCP-friendly congestion avoidance is very active. Meanwhile, proposals for video-oriented transport protocols continue to appear, but they now generally include TCP-friendly congestion avoidance. Our concern is over the marginal benefit of changing TCP's service model, given the presence of congestion avoidance. As a position paper, our contribution will not be in the form of final answers, but our hope is to convince the reader of the merit in re-examining the question: do applications need a replacement for TCP in order to do streaming video?

1 Introduction

The Internet's ubiquity has long made it an attractive platform for distributed multimedia applications. A particularly elusive goal has been effective streaming solutions. To prevent confusion, we clarify the distinction between streaming and other forms of distribution, namely download. We assume download is defined so that the transfer of the video must complete before the video is viewed. Transfer and viewing are temporally sequential. With this definition, it is a simple matter to employ quality-adaptive video. One algorithm would be to deliver the entire video in the order from low to high quality components. The user may terminate

* This work was partially supported by DARPA/ITO under the Information Technology Expeditions, Ubiquitous Computing, Quorum, and PCES programs and by Intel

D. Shepherd et al. (Eds.): IDMS 2001, LNCS 2158, pp. 213–218, 2001.

the download early, and the incomplete video will automatically have as high quality as was possible. Thus, quality-adaptive download can be implemented in an entirely best-effort, time-insensitive, fashion. On the other hand, we assume streaming means that the user views the video at the same time that the transfer occurs. Transfer and viewing are concurrent. There are timeliness requirements inherent in this definition, which can only be reconciled with best-effort delivery through a time-sensitive adaptive approach.

In considering TCP's viability for streaming video, our position has much in common with the recent proliferation of work on TCP-friendly streaming. For us, the important issue is whether TCP's service model need to change. Much of the TCP-friendly research does not involve changes to the programming interface, our position is concerned with proposals that do entail new service models.

2 Anti-TCP Dogma

Numerous works on streaming video have asserted that TCP is undesirable for multimedia streaming, yet propose alternate solutions compatible with the same best-effort IP infrastructure[3, 9, 17, 16]. In this section, we identify common objections to two of TCP's basic mechanisms, packet retransmissions and congestion control, that are at the root of this anti-TCP dogma.

2.1 Reliability through Retransmissions

One objection states that TCP's use of packet retransmissions introduces unacceptable end-to-end latency. The claim is that re-sending lost data is not appropriate because, given the real-time nature of video, the resent data would arrive at the receiver too late for display. Retransmissions can also be the result of packet re-ordering rather than loss, however the latency penalty for re-ordered packets will be small, since TCP will still accept an out of order packet when it arrives. We now consider the latency penalty for retransmission of lost packets. A TCP sender's earliest detection of lost packets occurs in response to duplicate ACKs from the receiver. TCP also uses timeouts, these should be rare for streams behaving as an infinite-source. An adaptive video streaming application will behave as such an infinite source, since it will attempt to use all the throughput TCP will provide. Therefore the typical time the re-transmission will arrive at the receiver is one full round-trip (RTT) after the lost data was originally sent, resulting in an end-to-end latency of 1.5 times RTT at the minimum[1]. Thus, the latency penalty for retransmission of lost packets will be on the order of one RTT. RTTs vary for numerous reasons on the wide-area internet, but the following is a rough taxonomy of RTT scales, and consequently the latency penalties resulting from TCP retransmission: 20ms between sites in the same region, 100ms for sites on the same continent, and about 200ms between sites

[1] There is no bound on TCP's contribution to end-to-end latency, since the underlying IP model implies that acknowledgments or packet retransmissions may be lost. However, retransmission-delay on the order of a single RTT is the normal case.