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,
WWW home page: 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

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.

requiring oceanic crossings. We now consider how these latencies would relate to video applications.

For *purely-interactive* applications such as tele-conferencing or distributed gaming, users are highly sensitive to end-to-end delays of sub-second timescales, typically in the range of 150 to 200 milliseconds. This end-to-end delay requirement persists for the duration of these applications. Given the tight delay bounds, we think it is important to characterize various delay sources using the critical-path approach[12, 2]. The question is how much the retransmission-delay effects the mean and worst-case critical-paths for interactive applications. The critical path approach stresses the importance of interaction with other sources of delay. If congestion control is essential to the best-effort Internet, it may be that its delays dominate the critical path. For a deeper discussion of latency implications of congestion control, we refer to our separate work[10], which begins towards our goal understanding the critical path for latency.

Unlike purely-interactive applications, video on demand (VOD) has interactive requirements only for control events such as start, pause, fast-forward, etc., which are relatively infrequent compared to the normal streaming state. While streaming, the quality perceived by the user is not directly affected by end-to-end latency, as the interaction is strictly uni-directional. A VOD application may gradually increase buffering, hence end-to-end delay, by dividing its use of available bandwidth between servicing video play-out and buffer accumulation. After a time, the end-to-end delay will actually be quite large, but the user perceives it only indirectly, in the sense that quality during the buffer accumulation period might have been slightly decreased. In this way, we say that VOD does not have the inherent hard latency requirements of purely-interactive applications, and so TCP's packet-retransmissions are not a significant problem for VOD.

2.2 Congestion Control

The congestion control algorithms of TCP have been heavily studied and frequently discussed in the literature[4, 6, 14]. Briefly, the congestion algorithm is designed to probe available bandwidth, through deliberate manipulation of the transmission rate. In steady-state, TCP's congestion control converges on an average transmission rate close to a fair-share of available bandwidth[2]. When viewed over shorter time-scales, TCP's instantaneous transmission rate takes on a familiar *sawtooth* shape, where it cycles between periods of additive increase separated by multiplicative decrease (AIMD). This short-term rate sawtooth is the second major part of the common view that TCP is not a good selection for video applications.

Many TCP-friendly protocols with claims of better suitability for video have been proposed[3, 9, 17, 16, 18]. These protocols recognize the need for congestion control, but propose congestion control such that rate is smoother in the short-term than TCP's AIMD sawtooth. Discussion in the literature of the network

[2] Fairness under distributed control is necessarily somewhat subjective. TCP's control algorithm results in bias toward flows with shorter path RTTs.

implications in terms of efficiency, stability and scalability, continues. We now consider the implications from the perspective of a streaming video application, which are manifest in terms of relationship between rate variations and buffering.

An application's TCP flow experiences rate variations for two distinct reasons; the first being competing traffic in the network, and the second being the flow's own congestion control behavior[3]. Rate variations may be categorized by the application as either transient or persistent. The distinction between transient and persistent rate changes is whether the buffer capacity is large enough to smooth them out. The purpose of buffering is precisely to smooth out transient changes.

For any amount of buffering, competing traffic can have persistent effects on a stream's rate. Streaming video applications must deal with persistent rate changes, before the client-side buffers are overwhelmed. The usual way is to employ quality-adaptation, adjusting the basic quality-rate trade-off of the video[3, 9, 17]. The applications use a closed loop-feedback control between client and server, which monitors the transport's progress for persistent rate changes and actuates the stream's quality-rate trade-off in response. We call this the quality-adaptation control.

Conceptually, the cyclic nature of congestion control's increase and decrease phases, the TCP sawtooth, suggests it should be treated strictly as a source of transient rate changes. If the quality-adaptation control is intended only to adjust for persistent traffic changes, then it has the problem of masking out the TCP sawtooth by inference. Without direct information, the quality-adaptation control may be less than optimal in terms of responsiveness. However, from the perspective of the human viewer, frequent video-quality changes are annoying, so the quality-adaptation control should favor stability over responsiveness. Stable quality is a natural outcome of employing large client side buffers using methods like those described in section 2.1. On the other hand, for purely-interactive applications, it may not be possible to treat congestion-control adjustments as transient, since end-to-end latency and buffer capacity are constrained. In this case, the design of the quality-adaptation control will have to choose between having higher average quality, allowing quality to track the sawtooth, or smoother quality by imposing a rate-limit. New congestion controls may reduce the impact of these trade-offs, since they may spread the congestion control rate adjustments more evenly[16, 5].

3 Popularity and Momentum

Studies of traffic trends in the Internet suggest that applications based on TCP comprise most of the traffic[13]. Solutions that allow Infrastructure providers to improve network efficiency and application performance without changing the applications are naturally compelling, so there is a strong incentive to improve TCP. At the moment, video comprises a small minority of Internet usage, so

[3] TCP's flow control may also contribute, but for our discussion we assume the client is not the limit.

video-only oriented transports have limited immediate appeal. Also, video-only transport proposals must struggle to overcome resistance based on their potential to disrupt existing majority of TCP based traffic. Meanwhile, improvements for TCP will move the performance target. We give two examples: Early Congestion Notification (ECN) and ATCP.

TCP's congestion control was predicated on the assumption that router buffer overflows were by far the most common source of packet losses. Accordingly, TCP's congestion control mechanism relies on packet losses to support probing for bandwidth and congestion detection, which implies a certain amount of deliberate waste. ECN is a proposal for extending IP and TCP so that active queue management at network nodes can pro-actively inform TCP of congestion before packet losses occur[15]. While the retransmission mechanism is still necessary for TCP's reliable service model, ECN allows TCP to perform congestion avoidance without packet losses. Performance evaluation of ECN shows that ECN-enabled TCP connections usually proceed with little or no retransmissions over their lifetime[1]. While this has immediate implications for interactive video, it also leads to solutions of another deficiency in TCP, namely its performance over the expanding component of the Internet consisting of wireless ad hoc networks. It is well known that Wireless links often suffer high bit error rates, which standard TCP will mis-interpret as congestion. Invoking congestion control for such errors impacts throughput more than is necessary, and is basically the wrong response. Liu and Singh[11] describe ATCP and show that with ECN it is possible to distinguish physical link losses from buffer overflows (congestion), and preserve TCP's throughput. While ECN and ATCP face deployment issues, the scope of change they propose is relatively modest, and yet they deliver comparable benefits to new protocols with video-centric service models.

4 Discussion

In this paper, we make the case that TCP is a viable and attractive choice for quality-adaptive video streaming. We discuss the main challenges for video applications using TCP, which are due to TCP retransmissions and congestion-control. For VOD applications, we describe how client side buffering can mitigate the effects of both. Further investigation is needed to understand how much interactivity is possible using TCP, and how much extra interactivity TCP alternatives make possible. We present initial study of TCP's relationship to interactivity in a separate work[10]. We have developed a video system prototype that supports tailorable fine-grained quality adaptation, of MPEG derived video, through priority packet dropping[7]. Based on our video system, we have developed a streaming algorithm over TCP, which we describe in an extended version of this report[8]. In future work, we will present measurements to illustrate the efficacy of our streaming system in supporting VOD over TCP, and explore further the issues of iteractivity.

References

1. Uvaiz Ahmed and Jamai Hadi Salim. Performance evaluation of explicit congestion notification (ECN) in IP networks. IETF RFC 2884, July 2000.
2. Paul Barford and Mark Crovella. Critical Path Analysis of TCP Transactions. In *In Proceedings of the 2000 ACM SIGCOMM Conference*, September 2000.
3. Shanwei Cen and Jonathan Walpole. Flow and congestion control for internet streaming applications. In *Proceedings Multimedia Computing and Networking (MMCN98)*, 1998.
4. Dah-Ming Chiu and Raj Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, 17, 1989.
5. Nick Feamster, Deepak Bansal, and Hari Balakrishnan. On the Interactions Between Layered Quality Adaptation and Congestion Control for Streaming Video. In *11th International Packet Video Workshop (PV2001)*, Kyongiu, Korea, April 2001.
6. Van Jacobson and Michael J. Karels. Congestion Avoidance and Control. In *In Proceedings of ACM SIGCOMM'88*, pages pp. 79–88, August 1988.
7. Charles Kasic and Jonathon Walpole. Qos scalability for streamed media delivery. CSE Technical Report CSE-99-011, Oregon Graduate Institute, September 1999.
8. Charles Krasic, Jonathan Walpole, Kang Li, and Asvin Goel. The case for streaming multimedia with tcp. Technical report, Oregon Graduate Institute, CSE Technical Report 2001. CSE-01-003.
9. J.R. Li, D. Dwyer, and V. Bharghavan. A transport protocol for heterogeneous packet flows. In *IEEE Infocom'99*, 1999.
10. Kang Li, Charles Krasic, Jonathan Walpole, Molly H. Shor, and Calton Pu. The minimal buffering requirements of congestion controlled interactive multimedia applications. In *IDMS*, Lancaster, UK, September 2001.
11. J. Liu and S. Singh. ATCP: TCP for Mobile Ad Hoc Networks, 2001.
12. K. G. Lockyer. *Introduction to Critical Path Analysis*. Pitman Publishing Co., New York, N.Y., 1964.
13. S. McCreary and K. Claffy. Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange.
14. Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *In Proceedings of ACM SICOMM'98*, 1998.
15. K. K. Ramakrishnan, Sally Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. IETF Internet-Draft, January 2001.
16. R. Rejaie, M. Handley, and D. Estrin. RAP: An end-to-end rate-based congestiong control mechanism for realtime streams in the internet. In *Proceeedings of IEEE Infocomm*, March 1999.
17. Wai tan Tan and Avideh Zakhor. Internet video using error resilient scalable compression and cooperative transport prototocl. In *Proc. ICIP*, volume 1, pages 17–20, 1998.
18. The TCP-friendly website. http://www.psc.edu/networking/tcp_friendly.html.