TCP Enhancements for Interactive Thin-Stream Applications

Andreas Petlund, Kristian Evensen, Carsten Griwodz, Pål Halvorsen
Simula Research Laboratory, Norway
Department of Informatics, University of Oslo, Norway
{apetlund, kristrev, griff, paalh}@simula.no

1. INTRODUCTION

TCP is frequently used for interactive multimedia applications like online games and voice-over-IP (VoIP) because it avoids firewall issues. However, traffic analysis shows that these streams usually have small packets and a low packet rate, and that in case of loss, severe latency penalties occur for all existing TCP variations in Linux [5]. In this demonstration, we show how small TCP enhancements greatly improve the perceived quality of such low latency, interactive applications.

2. INTERACTIVE THIN STREAMS

Many interactive applications have *thin stream* characteristics. In this context, a stream is considered *thin* if the application generates data in such a way that: a) The packet interarrival times are so high that the transport protocol's fast retransmission mechanisms are ineffective, and b) the size of most packets is well below the Maximum Segment Size (MSS).

Audio conferencing with real-time delivery of voice data across the network is an example of a class of applications that uses thin data streams and has a strict timeliness requirement due to its interactive nature. Nowadays, audio chat is typically included in virtual environments, and IP telephony is increasingly common. Many VoIP telephone systems use the G.7xx audio compression formats recommended by ITU-T. G.711 and G.729 have a bandwidth requirement of 64 and 8 Kbps, respectively. The packet size is determined by the packet transmission cycle (typically in the area of a few tens of ms, resulting in packet sizes of around 80 to 320 bytes for G.711). A similar example is Skype which boasts millions of registered users. We have, as an example, analyzed Skype sessions and seen that this application shares the characteristics of IP telephony. The packets are small (payload of approximately 110 bytes in average) and the bandwidth low (about 40 Kbps). To enable satisfactory interaction in such applications, ITU-T defines guidelines for the one-way transmission time. These guidelines state that users begin to get dissatisfied when the delay exceeds 150-200 ms and that the maximum delay should not exceed 400 ms [6].

Distributed online games are examples of thin-stream applications as well. We have analyzed game traces of several titles including Anarchy Online, World of Warcraft, Counter Strike, Halo 3 and Gears of War with respect to their network traffic characteristics. These games have packets sizes far below 250 bytes and rates below 20 packets per second. Occasionally, gamers experienced extreme worst-case delays. When considering user satisfaction, this class of applications requires tight timeliness, with latency thresholds at approximately 100 ms for first-person shooter games, 500 ms for role-playing games and 1000 ms for real-time strategy games [3].

With these characteristics and strict latency requirements in mind, supporting interactive thin-stream applications is challenging. The data streams in the described scenarios are poorly supported by the existing TCP variations in Linux. Their shortcomings are 1) that they rarely trigger fast retransmissions, thus making timeouts the main cause of retransmissions, and 2) that TCP-style congestion control does not apply because the stream cannot back off. Since improvements for TCP have mainly focused on traditional thick stream applications like web and ftp download, new mechanisms are needed for the interactive thin stream scenario.

3. TCP ENHANCEMENTS

The results of our earlier investigations of TCP [4, 5, 7] show that it is important to distinguish between thick and thin streams with respect to latency. They also show that there is potential for a large performance gain by introducing new mechanisms in the thin-stream cases. In short, if the kernel detects a thin stream, we trade a small amount of bandwidth for latency reduction and apply:

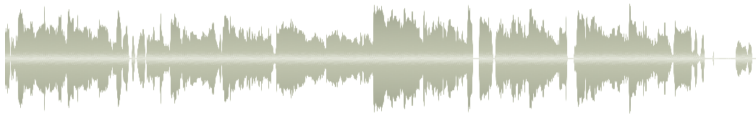
Removal of exponential backoff: To prevent an exponential increase in retransmission delay for a repeatedly lost packet, we remove the exponential factor [5].

Faster Fast Retransmit: Instead of waiting for 3 duplicate acknowledgments before sending a fast retransmission, we retransmit after receiving only one [7].

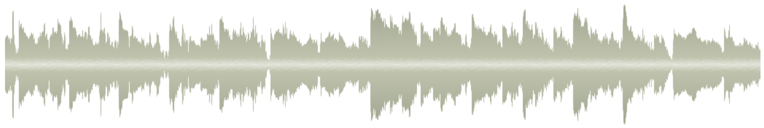
Redundant Data Bundling: We copy (bundle) data from the unacknowledged packets in the send buffer into the next packet if space is available [4].

As mentioned above, these enhancements are applied only if the stream is detected as thin. This is accomplished by defin-

NOSSDAV '08 Braunschweig, Germany

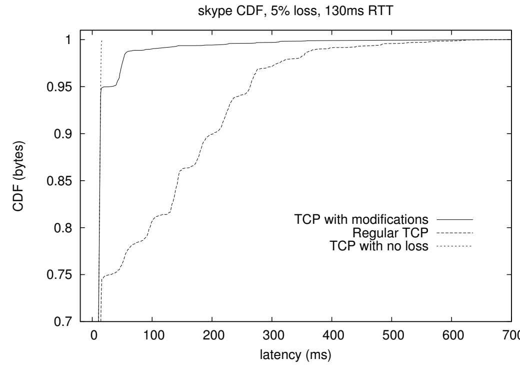


1) Unmodified TCP



2) Modified TCP

(a) A 20 second "audio spectrum" of a representative clip



(b) CDF of data arrival latency

Figure 1: Results from a Skype session.

ing thresholds for packet size and packets in flight. Also, we consider the redundancy introduced by our mechanisms acceptable because the streams are so thin that normal congestion mechanisms do not come into effect. Tests run so far indicate that fairness is indeed preserved.

4. RESULTS

We have performed several experiments with thin-stream applications like games and audio conferencing systems. All tests show improvements in user-perceived quality due to the reduced application layer latency when using our TCP enhancements. In this demo, we demonstrate the effects using the *Skype* audio conferencing system [2] and the *BZFlag* distributed game [1] as examples of interactive thin-stream applications.

We first demonstrate the performance gain of our enhancements in an audio conference. We have used Skype [2] which provides VoIP functionality, and which falls back to TCP when UDP is blocked (e.g., by ISP firewalls). The human ear is very sensitive to audio delays, and lost or delayed packets will quickly reduce the user-perceived quality. Figure 1(a) shows the received audio waves, and figure 1(b) shows the arrival latency of the received audio stream in a 2 % loss and 130 ms RTT scenario (UiO - UMASS) with and without our enhancements[1]. The figures show that the faster recovery of lost packets using our enhancements reduces the size and the number of gaps in the audio stream.

In the game demonstration, we show that, using our enhancements, we can achieve a better perceived gameplay. BZFlag [1] is a first person shooter, multi-user tank game. The game predicts movement, which has to cover a longer time period when packet loss delays delivery. In case of an erroneous prediction, tanks will jump to the correct position when the (retransmitted) position update arrives. Figure 2 shows the results from an example where the game is played for 5 minutes in a 5 percent loss and 100 ms RTT scenario with and without our enhancements. As we can see, the payload of lost packets is recovered faster when the server applies our enhancements.

Both examples will be set up for the demo session where we dynamically turn on and off our enhancements. Thus, the participants can interact with the applications and see and hear the increased quality of the service when our thin stream modifications for TCP are enabled.

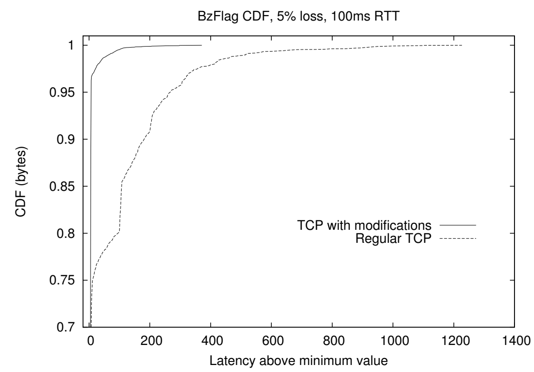


Figure 2: CDF of data arrival latency in BZFlag.

5. REFERENCES

[1] BZFlag, March 2008. http://bzflag.org.

[2] Skype, March 2008. http://www.skype.com.

[3] Claypool, M., and Claypool, K. Latency and player actions in online games. *Communications of the ACM 49*, 11 (Nov. 2005), 40–45.

[4] Evensen, K. R., Petlund, A., Griwodz, C., and Halvorsen, P. Redundant bundling in tcp to reduce perceived latency for time-dependent thin streams. *to appear in IEEE Communication Letters* (2008).

[5] Griwodz, C., and Halvorsen, P. The fun of using TCP for an MMORPG. In *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)* (May 2006), ACM Press, pp. 1–7.

[6] International Telecommunication Union (ITU-T). One-way Transmission Time, ITU-T Recommendation G.114, 2003.

[7] Pedersen, J., Griwodz, C., and Halvorsen, P. Considerations of SCTP retransmission delays for thin streams. In *IEEE Conference on Local Computer Networks (LCN)* (Nov. 2006), pp. 1–12.

[1] To be able to reproduce and compare the results, one of the audio conference participants played back different audio clips. The respective (received) audio clips can be downloaded from http://home.ifi.uio.no/apetlund/nossdavdemo