Impact of TCP on Interactive Real-Time Communication

Ilpo Järvinen[*], Binoy Chemmagate[*], Laila Daniel[*], Aaron Yi Ding[*], Markku Kojo[*], and Markus Isomäki[†]

[*] University of Helsinki

[†] Nokia

Abstract—**Providing acceptable quality level for media flows such as interactive video or audio is challenging in the presence of TCP. Volatile TCP traffic such as Web traffic causes transient queues to appear and vanish rapidly introducing jitter to the packets of the media flow. Meanwhile long-lived TCP connections cause standing queues to form. To get insights into this problem space we conducted experiments in a real high-speed cellular network. Our results confirm the existence of issues with both Web traffic and long-lived TCP connections and highlight that the use of parallel connections in Web browsers has high cost on media flows. In addition, the recent proposal to increase initial window of TCP to ten segments, if deployed, is going to make the jitter problem even worse.**

I. Introduction

Introducing delay sensitive end-to-end media flows such as interactive video and audio between Internet users introduces a number of challenges with congestion control. These challenges involve two interrelated problems. First, how to ensure that real-time communications behave fairly with other competing Internet traffic. Second, how to ensure good quality to the interactive media, in particular with the other competing traffic that the users potentially generate to share the bottleneck(s) on the end-to-to path. In this paper we focus on the latter challenge. In a common case the bottleneck resides in the access network of the end user, where most of the traffic, if not all, is that generated by the user. When we consider the link speed in developing or underdeveloped areas, we can see that, most of the users are still using residential access such as DSL or Mobile Broadband as primary Internet resource. Even in developed areas the link capacity for residential Internet access is quite often not more than a few Mbits/s.

Web traffic in general is very bursty and easily creates transient queues at bottlenecks in front of slow and moderate speed access links. These queues interfere with any competing traffic by introducing delay spikes that the delay sensitive flows encounter as harmful jitter. A large number of parallel TCP connections with typical Web traffic tends to intensify queuing effect and may dramatically increase the effect of the delay spikes, which is likely to be particularly harmful to delay sensitive traffic such as interactive audio and video.

A browser of today is quite aggressive. It uses many parallel TCP connections to speed up retrieval of the Web pages [Bro], [Sou08]. At the same time, websites "optimize" the end user experience by taking advantage of the parallel TCP connections feature of the browser. Typically, a higher limit is set by the browser for the number of parallel TCP connections with different domains. The "optimized" Web pages contain objects that seem to reside in a different domain but are instead coming from the same server. Those fake domains trick the browser to allow more parallel connections. Moreover, in the recent years some efforts have been made to increase initial window from three to ten segments [CDCM12], [D^{+}10]. Such increase put together with the large number of parallel TCP connections introduces rapidly changing environment for any traffic competing with those parallel TCP flows.

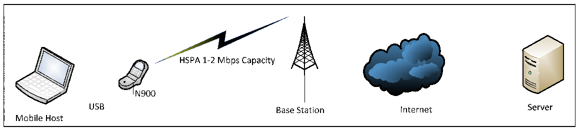


Fig. 1: Test environment

While solutions such as Low Extra Delay Background Transport (LEDBAT) [SHIK11] that attempt to keep queuing delay low exist. Their use for Web traffic would be controversial as the Web traffic is certainly not less than best effort type. Quite contrary, the browsers and websites aim to minimize the latency in Web page transmission which is in direct conflict with the carefulness that approaches such as LEDBAT need. Considering that current browsers and websites disregard advice on number of concurrent connections [F^{+}99] to shorten latency, it is unlikely that browser makers or website administrators would find LEDBAT or similar approach acceptable solution. Besides, deployment of a new TCP variant in large scale would be a challenge in itself. On the other hand, if such TCP variant would be used only on-demand when threat to harm media flows exists, additional signalling between the end hosts would be required as LEDBAT is implemented in the sender. Such signalling again would face deployment challenges.

On the network side, phenomenon called bufferbloat [Get11] has recently attracted some attention. Because of bufferbloat, devices in the network can hold enormous amount of traffic such as the initial windows of all parallel web requests. Active queue management (AQM) and its most prominent representative Random Early Detection (RED) [FJ93] is often proposed as a solution to the bufferbloat but that is challenging to realize in practice. The access network devices that are typically bottlenecks lack support for AQM/RED, and even if available, RED does not work with the default settings as it is "too gentle to handle fast changes due to TCP slow start when the aggregate traffic is limited" [JDNK12]. As tuning of the RED parameters requires modifications on the intermediate network nodes, it is not deployable in the short run on large scale even when RED itself is supported by the devices.

Media flows are typically reduced in size for transmission by a codec which tries to retain human observable properties of the original content while removing information where human senses cannot detect the changes. Usually codecs can conceal sporadic losses quite well, but when more losses occur consecutively, quality deteriorates and distortions become noticeable. A buffer between the receiving codec and the network absorbs jitter that occurs in the packet transmission over the network. The codec needs the data on time because the media playback is time bound. If a sudden delay increase occurs in the network, the packet might not arrive in time for the playback and needs to be discarded unused. Selecting a larger jitter buffer size is a tradeoff as it would allow larger jitter to occur but at the same time it increases the total end-to-end delay, potentially resulting in unacceptable interactive media quality.

Another problem for media flows are long-lived TCP connections such as software updates and file downloads. A long-lived TCP connection tends to create long queues that occupy the bottleneck buffers for a long period of time. The long term queues often cause high end-to-end one-way delay for interactive media and thereby result in unacceptable interactive media quality.

II. Experiments with Competing Real-time Media and TCP Traffic

We have conducted an experimental study to measure the effect of competing TCP traffic to interactive media. The experiments have been carried out over a real cellular Internet access using emulated traffic flows to allow full control over the workloads and more accurate analysis of the results. Although cellular access is used in the experiments, we believe that the results are representative for any access with similar moderate link capacity.

A. Test Setup and Workloads

The test system comprises of a mobile host and fixed server, as presented in Figure 1. We consider two workloads that are applicable for smart phones, tablets or any device with limited bandwidth.

1) Web browsing when a voice call is ongoing (**Audio+n**): Audio starts first and then the n short TCP flows start at the same time. The start time is distributed uniformly between 10 to 12 seconds after the start of the audio flow. The n short TCP flows can be one TCP flow, two TCP flows or six TCP flows. The total size of the short TCP flows is 372 kB.
2) Software update during a voice call (**Audio+Bulk**): Audio starts first and then a Bulk TCP transfer of 28 MB starts. Bulk TCP's start time is distributed uniformly between 10 to 12 seconds after the start of the audio flow.

In both scenarios an audio flow is ongoing while TCP traffic is starting in the middle of the audio flow. The audio flow lasts long enough to cover the whole duration of the TCP transfer. The direction of traffic in the both test cases are from a fixed host to mobile host. The audio flow is a constant bit-rate type with bit-rate of 16kbps yielding 32kbps total bit-rate with IP, UDP, and RTP headers. We run 50 replications with each different combination of test parameter values.

B. Initial results

We define a jitter filter to mimic codec behavior. First there are "pure losses" when a packet is dropped in the network, either due to congestion or link errors. With interactive media, there is also "delayed loss" when an audio packet misses the deadline for codec to decode and play the transmitted content. Such a packet is unusable similar to the pure loss. Delayed losses are flagged when one-way delay of the packet exceeds "base delay" plus jitter buffer size. The "base delay" is calculated as the minimum delay over the period of two seconds prior to the arrival of the TCP flows.

We specify quality metric to present the quality of the interactive media from codec and end user perspective. The quality metric is based on loss periods that are encountered by the codec when combining pure and delayed losses. Each data packet carrying interactive media (Audio) is assigned a quality metric value according to the definition in Table I.

Value	Description
0	no loss
1	20 ms gap in the stream, no adjacent packet lost
2	40-60 ms of the stream was lost
3	80-100 ms of the stream was lost
4	120-180 ms of the stream was lost
5	200+ ms of the stream was lost

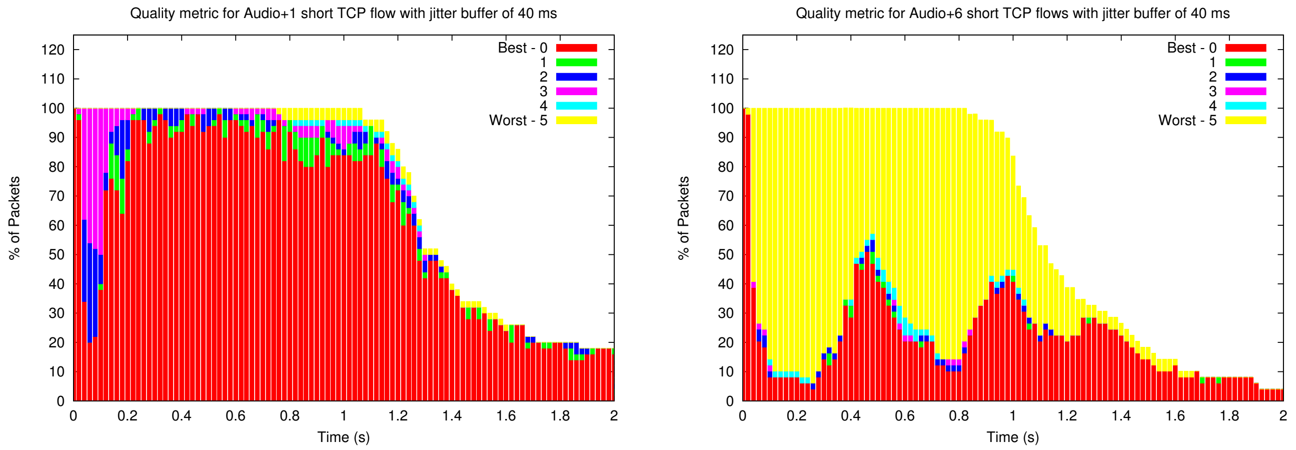
TABLE I: Quality Metric Definition

We intentionally chose to use minimum delay as base delay in order to report the worst-case behavior. As a real codec might choose higher value it is reasonable to assume that the quality cannot be worse than that indicated by the quality metric.

In the conducted experiments, the HSPA network introduced hardly any losses during the observed period. Therefore audio quality is determined mainly by the delay and the changes in the delay.

With Audio+Bulk interactive streaming is impossible because the one-way delays during the TCP transfer are prohibitive. Already the 25th percentile of the one-way delay is 0.5sec and the median is 1.42sec.

In Figure 2 is the audio quality with Audio+n short TCP flows as a function of time with one and six short concurrent TCP flows. The quality metric values are taken over 50 replications and filtered to only include the values which overlap with the TCP transfers and therefore the number of



(a) With one concurrent TCP flow, 50 replications

(b) With six concurrent TCP flows, 50 replications

Fig. 2: Audio quality during TCP transmission

samples starts to decline when the TCP flows in individual test replications were completed. Once the TCP flows start at zero seconds, almost immediately the quality deteriorates as the SYN handshake complete and the flows inject their initial window into the network. We note that the initial window causes the worst quality moments during the whole transfer. When only single connection is in use the quality is not falling to the worst quality level and quality level is rapidly restored after the initial window around 0.2 seconds. However, with six concurrent connections the quality level is very bad right from the beginning and affects almost the whole duration of the TCP transmissions.

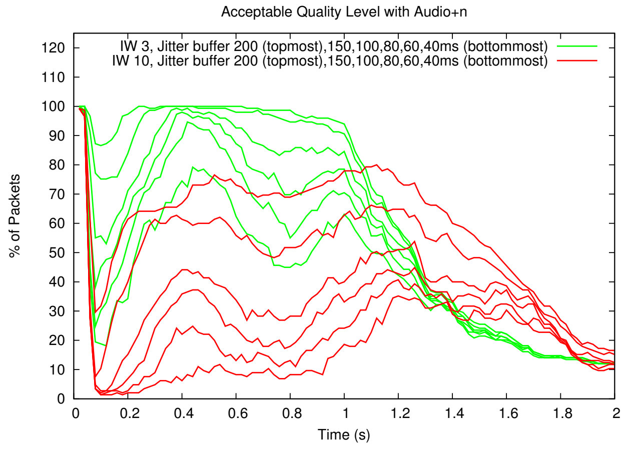


Fig. 3: Overview of acceptable audio quality with different jitter buffer sizes and initial window settings

Figure 3 summarizes the audio quality with Audio+n workload when different initial window is used. The quality metric levels 0 and 1 are combined to determine "acceptable" quality level (i.e., any loss does not have an adjacent packet lost) and all the cases with one, two or six short TCP flows are considered together. We observe that IW10 is clearly worse than the IW3 as the quality drops to the lowest level for prolonged time. The aggressive start with IW10 makes also the later part of transfer to trigger more discarding at the codec.

III. Conclusion

In this paper we present how audio quality is affected by concurrent TCP transmissions in a high-speed cellular network. Even a moderate number of parallel Web page responses causes irreparable harm for audio transfer. We measured that audio data is delayed too much which prevents codec from using it before the playback deadline. We also show that the worst quality during a short transmission is measured during the burst of packets that occur because of the initial window transmission, and that initial window of ten segments is worse for the competing audio flow than initial window of three segments. With bulk TCP transfer, audio stream becomes unusable for interactive purposes.

References

[Bro] Browserscope. http://www.browserscope.org/?category=network&v=1.

[CDCM12] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis. Increasing TCP's Initial Window. Internet Draft, Internet Society, February 2012.

[D+10] N. Dukkipati et al. An argument for increasing tcp's initial congestion window. *ACM SIGCOMM Computer Communications Review*, 40(3):26–33, 2010.

[F+99] R. Fielding et al. Hypertext transfer protocol – HTTP/1.1. RFC 2616, June 1999.

[FJ93] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[Get11] J. Gettys. IW10 Considered Harmful. Internet Draft, Internet Society, August 2011.

[JDNK12] I. Järvinen, A. Y. Ding, A. Nyrhinen, and M. Kojo. Harsh RED: Improving RED for Limited Aggregate Traffic. In *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA 2012)*, March 2012.

[SHIK11] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind. Low Extra Delay Background Transport (LEDBAT). Internet Draft, Internet Society, October 2011.

[Sou08] S. Souders. Roundup on parallel connections, March 2008. http://www.stevesouders.com/blog/2008/03/20/roundup-on-parallel-connections/.