Effect of Competing TCP Traffic on Interactive Real-Time Communication

Ilpo Järvinen*, Binoy Chemmagate*, Aaron Yi Ding*, Laila Daniel*, Markus Isomäki†, Jouni Korhonen‡, and Markku Kojo*

* University of Helsinki † Nokia ‡Nokia Siemens Networks

Abstract. Providing acceptable quality level for interactive media flows such as interactive video or audio is challenging in the presence of TCP traffic. 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 which increases the one-way delay for the media flow packets. 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-like traffic and long-lived TCP connections and highlight that current trend of using several parallel connections in Web browsers tends to have high cost on media flows. In addition, the recent proposal to increase the initial window of TCP to ten segments, if deployed, is going to make the jitter problem even worse.

1 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-end 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 the primary Internet access. Even in developed areas the link capacity for residential Internet access is quite often not more than a few megabits per second.

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 delay sensitive flows encounter as harmful jitter. Moreover, a browser of today is quite aggressive using many parallel TCP connections to speed up retrieval of the Web pages [2, 15]. At the same time, websites "optimize" the end user experience by taking advantage of the parallel TCP connections feature of the browser. The "optimized" Web pages contain objects that seem to reside in different domains but are instead coming from the same server. Such fake domains trick the browser to allow more parallel connections as browsers limit the number of parallel connections per domain. The use of 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. Moreover, in the recent years some efforts have been made to increase the initial window of TCP from three to ten segments [3, 5]. Such increase together with the large number of parallel TCP connections introduces rapidly changing environment for any traffic competing with the parallel TCP flows.

While solutions such as Low Extra Delay Background Transport (LEDBAT) [14] 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 [6] to shorten latency, it is unlikely that browser makers or website administrators would find LEDBAT or similar approach an 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 a threat to harm media flows exists, additional signalling between the end hosts would be required as LEDBAT is implemented at the sender. Such signalling again would face deployment challenges.

On the network side, phenomenon called bufferbloat [8, 11] has recently attracted some attention. Because of bufferbloat, devices in the network can end up buffering enormous amount of traffic such as the initial windows of all parallel web responses. Active queue management (AQM) and its most prominent representative Random Early Detection (RED) [7] 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" [10]. 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 if 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 jitter 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 media 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, resulting in unacceptable interactive media quality.

Some studies explored media flows and Web Traffic in 3G/3.5G network [9, 16]. In these studies, however, the different traffic types might not be competing with each other over the cellular data channel. In this study we focus on the effect of simultaneous TCP flows on interactive media, and also on the effect of the larger TCP initial window [3]. To our best knowledge, neither effect has been explored in a 3G/3.5G environment before. Although cellular access is used in the experiments, we believe that the results are representative for any access with similar moderate link capacity because deep buffers are a widespread phenomenon [8]. TCP performance and the interactions between parallel TCP connections are out of scope for this study.

In this paper we measure the effect of competing TCP traffic to interactive media flows in a real high-speed cellular network environment. The rest of this paper is organized as follows. In Section 2 we introduce the test setup and workloads for the experimentation. In Section 3 we analyze how TCP traffic affects the one-way delay and delay variation of a media flow. In Section 4 we analyze the transient effect of jitter-induced loss periods on a media flow and in Section 5 we conclude our findings.

2 Test Setup and Workloads

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. The test system comprises of a mobile host and fixed server, as presented in Figure 1.

In order to get the baseline for interactive media flow behavior without competing traffic in the test environment we first measure the performance of an

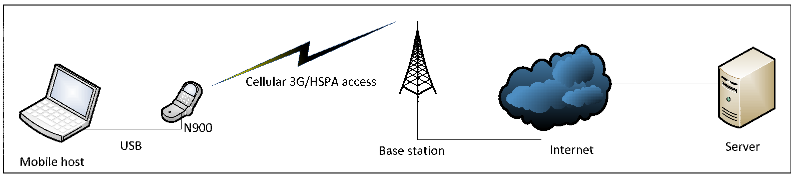


Fig. 1: Test environment

emulated audio only workload. We then focus on the two major workloads that roughly mimic two typical TCP traffic loads competing with an interactive media flow: (1) Software update during a voice call (*Audio+Bulk*) and (2) Web browsing when a voice call is ongoing (*Audio+n short TCP flows*). In the *Audio+Bulk* workload, an emulated audio flow 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 the *Audio+n short TCP flows* workload, an emulated audio flow starts first and then n short TCP flows start at the same time, the start time being 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. In both scenarios the 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 all test cases is from the fixed server to the mobile host. We also send enough warming up packets right before each test run to ensure that a dedicated channel (DCH state) is allocated for the actual test data, and thereby radio state changes are not affecting the results. The n short TCP flows are tested with initial window of three (IW3) and initial window of ten (IW10). The audio flow is a constant bit-rate (CBR) type with bit-rate of 16 kbps yielding 32 kbps total bit-rate with IP, UDP, and RTP headers, that is, an IP packet of 80 bytes is transmitted every 20 ms. We run 50 replications with each different combination of test parameter values. All the test traffic is captured using tcpdump [17] on both the mobile host and the fixed server. We carefully synchronized the end host clocks prior each test run using Network Time Protocol (NTP) [13] allowing initially enough time for the clocks to be slowly adjusted towards almost equal rates. This enabled us to measure one-way delay [1] for each media packet with reasonable accuracy by taking the difference in timestamps found in the tcpdump logs at each end.

3 Effects on One-way Delay and Delay Variation

In the conducted experiments, the HSPA network introduced hardly any losses during the observed period. Therefore, the effect of competing TCP traffic is mainly due to the delay and the changes in the delay. While analyzing the results, we noticed that on a few occasions the wireless link introduced very long delays to packet delivery ranging from 3 seconds to rare occurrences with more than 60 seconds of delay. Also a large number of consecutive losses, reordering, or packet duplication occurred during such events. We choose to filter out the cases where clear symptoms of such event occurred because we are interested in how TCP affects media flow rather than wireless link problems. As we do not have access to the cellular operator network to collect traces, we cannot confirm the exact cause for this "wireless phenomenon" but in most of the cases they are likely to be caused by the cellular access deciding to switch access technology.

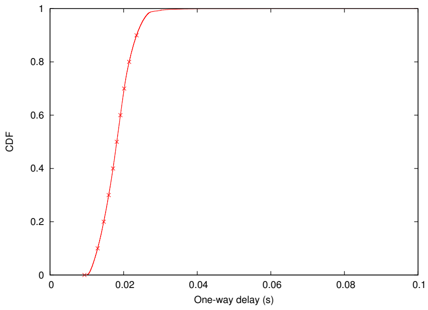


Fig. 2: CDF of one-way delay for 15 secs audio only workload, 50 replications

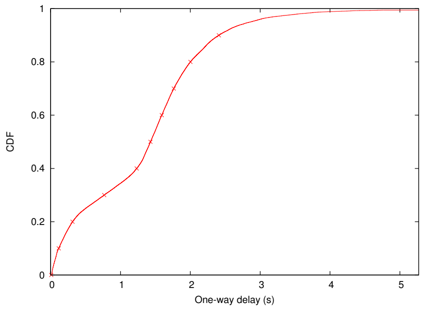


Fig. 3: CDF of one-way delay for an audio flow with a competing Bulk TCP connection, 50 replications

Figure 2 shows the cumulative distribution function (CDF) of end-to-end one-way delay [1] for 15 secs audio only workload. The one-way delay is good enough for inter-active audio conversation. The loss-rate is only 0.05 %. The delays remain below 40 ms except for a handful of packets, the median and maximum measured one-way delay being 18.0 ms and 70.4 ms, respectively.

Figure 3 shows the CDF of one-way delay for the media flow packets during a bulk TCP transfer. With the competing bulk TCP transfer interactive audio is impossible because the one-way delays during the TCP transfer are prohibitive. Already the 25th percentile of the one-way delay is 0.5 secs and the median is 1.42 secs. We confirmed from the traces that deep buffering is the main cause for the delay increase; soon after the bulk TCP transfer starts the delay increases and remains around 1.5-2.5 secs consistently for the duration of the TCP transfer. Such a delay increase was not present in audio only results. Few values especially in the highest end, however, might be due to wireless network phenomena on top of the deep buffering.

Figure 4 shows CDF of the one-way delay for the media flow with short TCP flows when different number of TCP connections and different TCP initial window sizes are in use. The one-way delay with one competing TCP flow using initial window of three segments is reasonably low and seems to allow smooth packet delivery for interactive media. Increasing the number of TCP connections from one to two causes only a moderate increase in the end-to-end delay. However, increasing the TCP connection count to six introduces larger one-way delays, and the sharp knee transition with one or two flows is transformed into an earlier increase in the one-way delay affecting roughly 40 % of the packets. However, in all cases with competing TCP traffic using IW3 the one-way delay remains below 150ms all the way up to 75th percentile.

The one-way delay with competing TCP traffic using initial window of ten segments is notably higher than when using initial window of three in all corresponding cases. In all cases with the initial window of ten the one-way delay is higher than with the case of six TCP connections using the initial window

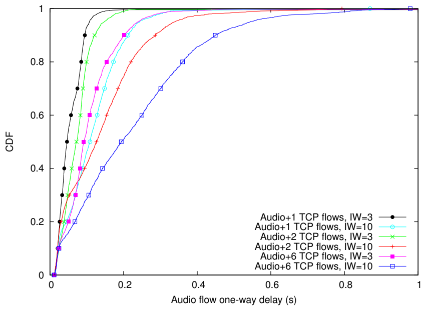


Fig. 4: CDF of one-way delay for an audio flow competing with n short TCP flows, 50 replications

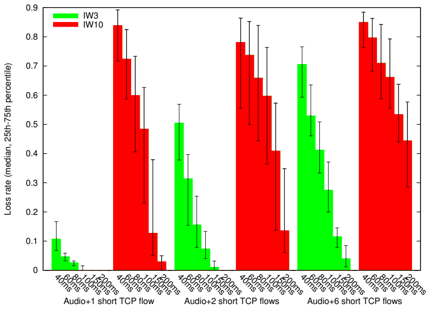


Fig. 5: Loss rate with different jitter buffer sizes for Audio+n short TCP flows workload

of three. The median one-way delay with six competing TCP flows using IW10 approaches 200ms but remains below 150ms even with one and two competing TCP flows.

IP Packet Delay Variation (IPDV) [4] for the media flow is shown in Table 1. As the high-end values seemed to correlate well with the increase in the size of the combined initial windows of parallel TCP flows, we extracted from the packet traces those TCP data packets that are received between two audio packets and confirmed that the large IPDV values typically occur when the TCP initial windows are among those TCP packets. In particular, with IW10 the large IPDV values are mostly introduced when the TCP flows inject the initial windows into the network.

4 Estimated Delay Induced Loss Period Effects

In order to explore the transient effect of the delay jitter on the media flow, we introduce a jitter filter to mimic receiving codec behavior in dropping late arriving media flow packets. 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 "delay-based loss" when a media flow packet delay exceeds the jitter buffer limit and thereby misses the deadline for codec to decode and play the transmitted content. Such a packet is unusable similar to the pure loss.

Table 1: CDF of IPDV for an audio flow competing with n short TCP flows, 50 replications

IW	n	Min	25%	Median	75%	90%	95%	96%	97%	98%	99%	Max
3	1	-0.020107	-0.011373	-0.000206	0.009194	0.020072	0.029445	0.031174	0.034697	0.043296	0.070158	0.111526
3	2	-0.020102	-0.011242	-0.000281	0.008824	0.018924	0.028301	0.029892	0.039526	0.050787	0.100523	0.182076
3	6	-0.020107	-0.011696	-0.000588	0.001666	0.012330	0.025413	0.031916	0.059762	0.081594	0.125042	0.282826
10	1	-0.020414	-0.012084	-0.000482	0.001835	0.016195	0.020696	0.029253	0.030297	0.050413	0.172464	0.242798
10	2	-0.020128	-0.019264	-0.000919	0.003032	0.019432	0.030032	0.031393	0.041291	0.070785	0.160448	0.322197
10	6	-0.020098	-0.019541	-0.009664	0.000454	0.018741	0.030004	0.040417	0.069099	0.121090	0.220447	0.589717

Delay-based 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.

Figure 5 shows the loss rate with different jitter buffer sizes, number of connections, and initial window settings. The loss rate is determined by combining pure losses and delay-based losses. IW10 increases the loss rate dramatically to nearly 100% with lower jitter buffer sizes. However, also IW3 with a large number of parallel connections produces significant number of losses. We want to reiterate that these losses occur almost solely due to excessive delay, not due to pure losses.

As codecs often are able to conceal isolated losses quite well, we specify a metric to estimate loss period effect on the interactive media from codec and end user perspective. The estimate is based on loss periods [12] that are encountered by the codec when several consecutive media flow packets are dropped due to jitter delay. We combine also pure losses into this metric though pure losses occur infrequently in our experiments. For a given jitter buffer size, each data packet carrying interactive media (Audio) is assigned a loss period level according to the definition in Table 2.

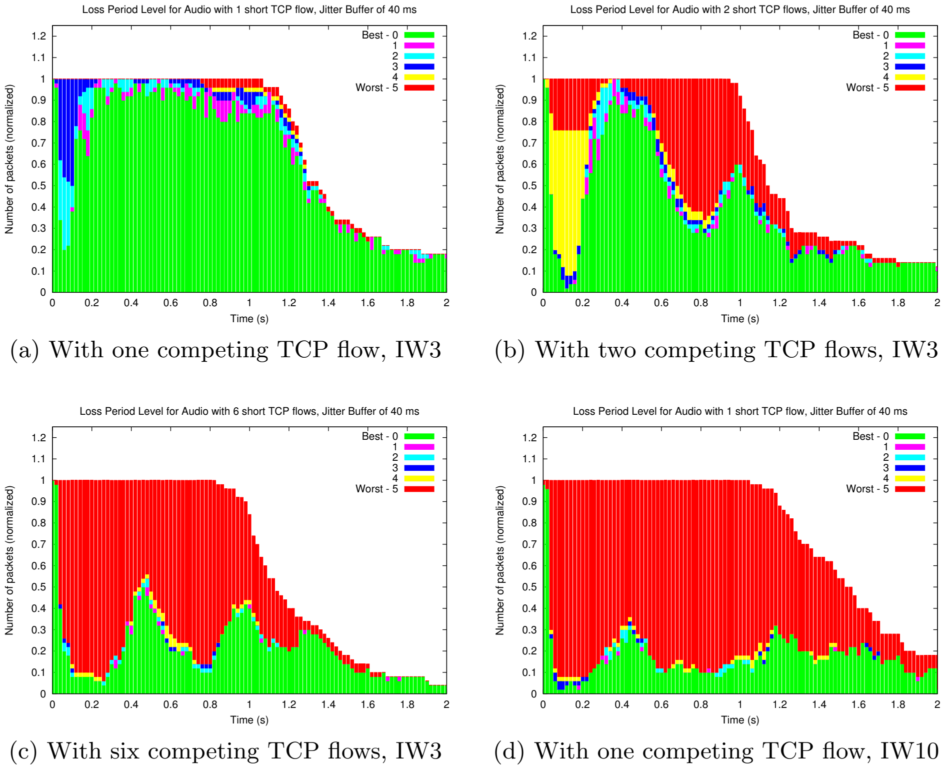
We intentionally chose to use minimum delay as base delay in order to report the worst-case behavior. As a real codec might choose higher value, it is reasonable to assume that the loss period effect is unlikely to be worse than that indicated by the loss period level.

In order to better understand transient effects that are hidden with CDF, Figures 6a, 6b, and 6c estimate the loss period effect in a function of time for a media flow using 40 ms jitter buffer size and competing with 1, 2, and 6 short TCP flows, respectively. 50 replications are included in each test case. The loss period level values are filtered to only include the media flow packets that overlap with the TCP transfers and therefore the number of samples starts to decline around 1 second when the TCP flows in individual test replications start to complete.

Almost immediately when the TCP flows start the TCP traffic generates significant loss period effect on the media flow packets, as the SYN handshakes complete and the TCP flows inject their initial windows into the network. We note that the arrival of the initial windows causes the worst effect during the

Table 2: Loss period level definition for estimating loss period effects

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



(a) With one competing TCP flow, IW3

(b) With two competing TCP flows, IW3

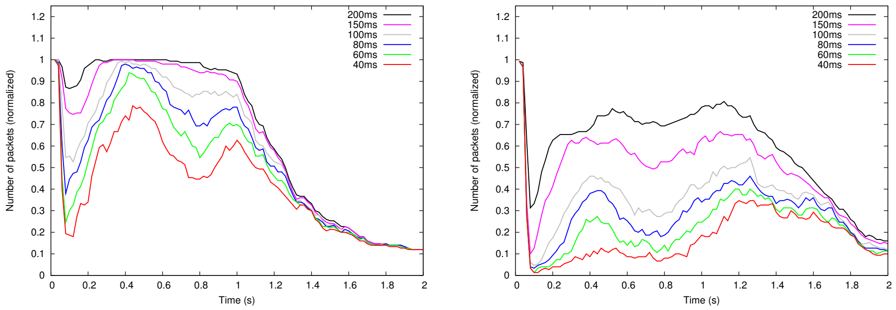
(c) With six competing TCP flows, IW3

(d) With one competing TCP flow, IW10

Fig. 6: Estimated loss period levels for audio packets when an audio flow using 40 ms jitter buffer competes with TCP transfers, 50 replications

whole transfer. When only a single TCP connection is competing with the media flow the loss period effect is not falling to the worst level and the level is rapidly restored after the initial window around 0.2 seconds. With two TCP flows the initial window injection causes much worse effect than with one concurrent flow but still the media flow is able to restore better level once the initial windows have been transmitted. However, as the two TCP flows start to open up their window resulting in more jitter the loss period effect again becomes notable. With six concurrent connections the loss period level is very bad right from the beginning and affects almost the whole duration of the TCP transfers. Figure 6d shows the loss period level with one competing TCP flow using IW10. The worst loss period level immediately becomes dominant like in case of six TCP flows with IW3 and remains dominant all the way until the completion of the TCP flows.

Figure 7 summarizes the estimated loss period effect on the media flow with n competing TCP flows when different IW sizes are used. The loss period levels 0 and 1 are combined to determine “acceptable” level (i.e., no lost packet has



(a) IW3, 150 replications (n=1,2,6)

(b) IW10, 150 replications (n=1,2,6)

Fig. 7: Overview of the acceptable loss period level for an audio flow with different jitter buffer sizes when 1, 2, and 6 TCP flows using (a) IW3 and (b) IW10 compete with the audio flow

an adjacent packet lost) and all the cases with one, two, or six short TCP flows are considered together. We observe that the number of acceptable media flow packets is clearly lower with IW10 than with IW3 for all corresponding jitter buffer sizes. The aggressive start with IW10 is also likely to make the later periods of transfer to trigger more delay-based packet discarding at the codecs.

5 Concluding Remarks

In this paper we present how interactive media flows are affected by concurrent TCP transmissions in a high-speed cellular network. Our measurements show that the packets of the media flow are heavily delayed when competing with TCP connections, which is likely to prevent a codec from using significant portion of the packets before the playback deadline. Even a moderate number of parallel TCP connections that are typically used for carrying Web page responses, for example, causes irreparable harm for a concurrent interactive media transfer. Startup dynamics for individual TCP connections may vary between the browsers and Web pages but we believe that our current measurements captured the major effect regardless of different mechanisms in browsers for launching parallel connections. Such variations are just likely to result in numerous variants of similar behavior.

Our experiments also indicate that during a short TCP transmission the worst effect on the media flow is measured during the burst of packets that occur because of the initial TCP window transmission, and that initial window of ten segments is worse for the competing media flow than initial window of three segments. With a competing bulk TCP transfer, the media stream becomes unusable for interactive purposes.

As the media flow performance degradation is caused by the behavior of Web traffic and deep buffers, we believe that the results are representative also for other than cellular access. The performance data is available at: http://www.cs.helsinki.fi/group/wibra/pam2013-data/.

References

1. Almes, G., Kalidindi, S., Zekauskas, M.: A One-way Delay Metric for IPPM. RFC 2679 (Sep 1999)
2. Browserscope: `http://www.browserscope.org/?category=network&v=1`
3. Chu, J., Dukkipati, N., Cheng, Y., Mathis, M.: Increasing TCP's Initial Window. Internet Draft (Nov 2012), Work in progress
4. Demichelis, C., Chimento, P.: IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). RFC 3393 (Nov 2002)
5. Dukkipati, N., et al.: An Argument for Increasing TCP's Initial Congestion Window. ACM SIGCOMM Computer Communications Review 40(3), 26–33 (2010)
6. Fielding, R., et al.: Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Jun 1999)
7. Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking 1(4), 397–413 (Aug 1993)
8. Gettys, J.: IW10 Considered Harmful. Internet Draft (Aug 2011), Work in progress
9. Huang, J., et al.: Anatomizing Application Performance Differences on Smartphones. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys). pp. 165–178 (Jun 2010)
10. Järvinen, I., Ding, Y., Nyrhinen, A., Kojo, M.: Harsh RED: Improving RED for Limited Aggregate Traffic. In: Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA) (Mar 2012)
11. Jiang, H., Liu, Z., Wang, Y., Lee, K., Rhee, I.: Understanding Bufferbloat in Cellular Networks. In: Proceedings of the Workshop on Cellular Networks: Operations, Challenges, and Future Design (CellNet) at SIGCOMM 2012 (Aug 2012)
12. Koodli, R., Ravikanth, R.: One-way Loss Pattern Sample Metrics. RFC 3357 (Aug 2002)
13. Mills, D., Martin, J., Burbank, J., Kasch, W.: Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905 (Jun 2010)
14. Shalunov, S., Hazel, G., Iyengar, J., Kuehlewind, M.: Low extra delay background transport (LEDBAT). RFC 6817 (Dec 2012)
15. Souders, S.: Roundup on Parallel Connections (Mar 2008), `http://www.stevesouders.com/blog/2008/03/20/roundup-on-parallel-connections/`
16. Tan, W., Lam, F., Lau, W.: An Empirical Study on the Capacity and Performance of 3G Networks. IEEE Transactions on Mobile Computing 7(6), 737–750 (Jun 2008)
17. TCPDUMP/LIBPCAP public repository, `http://www.tcpdump.org/`