A QoE Perspective on Sizing Network Buffers

Oliver Hohlfeld
RWTH Aachen University

Enric Pujol
TU Berlin

Florin Ciucu
University of Warwick

Anja Feldmann
TU Berlin

Paul Barford
UW Madison

ABSTRACT

Despite decades of operational experience and focused research efforts, standards for sizing and configuring buffers in network systems remain controversial. An extreme example of this is the recent claim that excessive buffering (*i.e.*, *bufferbloat*) can severely impact Internet services. In this paper, we systematically examine the implications of buffer sizing choices from the perspective of factors impacting *end user experience*. To assess user perception of application quality under various buffer sizing schemes we employ Quality of Experience (QoE) metrics. We evaluate these metrics over a wide range of end-user applications (*e.g.*, web browsing, VoIP, and RTP video streaming) and workloads in two realistic testbeds emulating access and backbone networks. The main finding of our extensive evaluations is that *network workload*, rather than buffer size, is the primary determinant of end user QoE. Our results also highlight the relatively narrow conditions under which bufferbloat seriously degrades QoE, *i.e.*, when buffers are oversized and sustainably filled.

Categories and Subject Descriptors

C.2.6 [**Internetworking**]: Routers

Keywords

Buffer size; bufferbloat; QoE

1. INTRODUCTION

Packet buffers are widely deployed in network devices to reduce packet loss caused by transient traffic bursts. Surprisingly, even after decades of research and operational experience, ‘proper’ buffer sizing remains challenging due to inherent trade-offs in performance metrics applied to the problem and different application requirements. While queueing theory suggests that large buffers improve transfer throughput at the expense of larger delays, there exists real-time applications requiring low and consistent delay, and thus preferring little to no buffering. Responding to the need to address such orthogonal objectives, the community has been involved in a decades-long struggle to identify general rules for both sizing and managing buffers that tries to match both ends of the spectrum (*i.e.*, low delay and high performance).

Traditionally, router buffer sizing is proportional to the bandwidth of the linecards *i.e.*, bandwidth-delay product (BDP). This rule-of-thumb emerged in the mid 1990s based on studies of the dynamics of TCP flows [25, 44]. A decade later, Appenzeller et al. reexamined buffer sizing and argued that throughput can be maintained using much smaller buffer sizes in core routers [9]. This reignited interest in the research community with regards to buffer dimensioning schemes, however the issue continues to remain far from resolved.

Recently, the buffer sizing debate has focused on the *existence* of large buffers in the network edge (bufferbloat [6]) and stimulated a debate on its potential negative effects. Excessive buffering *can* cause excessive queuing delays, *e.g.*, in the order of seconds), in phases of congestion when the buffer capacity is fully utilized. Resulting excessive delays *can* degrade the performance from a users' perspective [6], *e.g.*, by adversely effecting TCP due to increased round trip times or unnecessary timeouts. While the *existence* of large buffers has been observed, little is known on how often queues are *utilized* to degrade performance in practice. Also, the evaluation of the bufferbloat problem has so far focused on evaluating influence on QoS metrics. In the absence of a solid understanding, buffer sizes are currently used to drive engineering changes in Internet standards (see *e.g.*, [18]) and motivate new AQM approaches (*e.g.*, CoDeL [32]). We posit that a deeper understanding of buffering effects is needed before altering important engineering aspects.

This paper describes the first comprehensive study on the impact of buffer sizes on *end-user quality*. The goal of our work is to elucidate the sizing issues empirically and to pave the way for more informed sizing decisions. Unlike previous studies that consider Quality of Service (QoS) metrics (*e.g.*, packet loss or throughput) our study focuses on end-user *Quality of Experience (QoE)*. The use of standardized QoE metrics enables *estimation* of end-user perceived quality *without involving human subjects*. By using QoE metrics rather than conducting user studies, we are able to assess quality in an extensive sensitivity study involving a broad range of buffer size and workload configurations. Identified experimental scenarios pave the way for controlled user studies conducted in a scaled down fashion in the future.

Concretely, we evaluate QoE metrics for relevant user applications (*i.e.*, web browsing, VoIP, and RTP video streaming) in two realistic laboratory-based testbeds: access and backbone networks. Each application type is analyzed over Internet-like traffic scenarios—without isolation in separate QoS classes—and over a range of buffer sizes.

IMC'14, November 5–7, 2014, Vancouver, BC, Canada.

ACM 978-1-4503-3213-2/14/11 ...$15.00.
http://dx.doi.org/10.1145/2663716.2663730.

Our main observations are as follows:

1. We mainly find *network workload*, rather than buffer size, to be the primary determinant of end-user QoE. As intuitively expected, sustainable congestion impacts both QoS and QoE metrics by keeping the queue of the bottleneck buffer filled. This effect is amplified by large (bloated) buffers. In the absence of congestion, however, (even bloated) buffer sizes impact QoS metrics, as observed by previous studies, *e.g.*, [11], but impact QoE metrics only marginally. The good news for network operators is that limiting congestion, *e.g.*, via QoS or over-provisioning, can yield more immediate QoE improvements than efforts to optimize buffering.

2. We show the perceptual (QoE) perspective on buffering to differ from the known QoS perspective. This further emphasizes the use of application specific and perceptual metrics in Internet measurements. In this regard, this paper serves as an example on the use of QoE metrics for measurement studies.

2. RELATED WORK

The rule-of-thumb [25, 44] for dimensioning network buffers relies on the bandwidth-delay-product (BDP) $RTT * C$ formula, where RTT is the round-trip-time and C is the (bottleneck) link capacity. The reasoning is that, in the presence of *few* TCP flows, this ensures that the bottleneck link remains saturated even under packet loss. This is not necessary for links with a large number of concurrent TCP flows (*e.g.*, backbone links). It was suggested in [44] and convincingly shown in [9, 11] that much smaller buffers suffice to achieve high link utilizations. The proposal is to reduce buffer sizes by a factor of $\sqrt{n}$ as compared to the BDP, where n is the number of concurrent TCP flows [9]. Much smaller buffer sizes have been proposed, *e.g.*, drop-tail buffers with $\approx 20 - 50$ packets for core routers [17]. However, these come at the expense of reduced link utilization [11]. This problem has been addressed by a modified TCP congestion control control scheme that aims to maintain high link utilizations in small buffer regimes [20]. For an overview of existing buffer sizing schemes we refer the reader to [45].

While the above discussion focuses on backbone networks, more recent studies focus on access networks, *e.g.*, [13, 28, 30, 42], end-hosts [1], and 3G networks [22]. These studies find that excessive buffering in the access network exists and can cause excessive delays (*e.g.*, on the order of seconds). This has fueled the recent bufferbloat debate [6, 19] regarding a potential degradation in Quality of Service (QoS).

Indeed, prior work has shown that buffer sizing impact QoS metrics. Examples include *network-centric* aspects such as per-flow throughput [33], flow-completion times [29], link utilizations [11], packet loss rates [11], and fairness [46]. Sommers *et al.* studied buffer sizing from an operational perspective by addressing their impact on service level agreements [37]. However, QoS metrics and even SLAs do not necessarily reflect the actual implications for the end-user. A first step towards investigating the impact of buffering on gaming QoE has been made in simulations for Poisson traffic [36]. In this paper, we present the first *QoE centric* study that broadly investigates the impact of buffering and background traffic by using realistic testbed hardware and Internet like traffic scenarios.

3. BUFFERING IN THE WILD

Before investigating the *impact* of buffering on QoE, we first motivate our study by investigating the *occurrence* of buffering in the wild. Our analysis is based on snapshots of Linux kernel level TCP statistics for 430 million randomly selected TCP/HTTP connections captured at a major Content Distribution Network (CDN). The data was collected at different vantage points, located primarily in central Europe, over a period of five months in 2011. All flows were established by end-users to retrieve content from the respective CDN caches, thus they capture typical web browsing activity. This data corpus represents a significant sample of Internet users. It includes 81 million unique IP addresses originating from 22,490 autonomous systems (roughly 60% of the total advertised ASes when capturing the trace), located in more than 220 countries. Due to the vantage point locations, 56% of the IPs are located in central Europe.

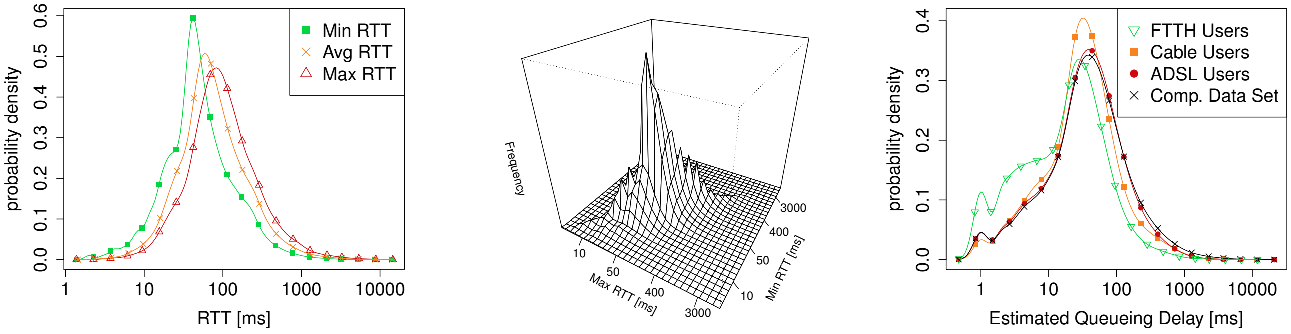
We build our evaluation on smoothed RTT (sRTT) information reported in the data set. Smoothed RTT values are estimated by the TCP stack using Karn's algorithm and are provided by the kernel level TCP statistics. For each TCP connection, the data set reports *(i)* the minimum sRTT, *(ii)* the average sRTT, *(iii)* the maximum sRTT, and *(iv)* the number of samples. To evaluate the variability due to queueing, we focus on flows that have at least 10 RTT samples. The distribution (PDF) of the logarithm of the minimum, average, and maximum RTT is shown in Figure 1a. The plot highlights that the average and maximum RTT deviate significantly from the minimum RTT, which is one indicator of possible queueing. Figure 1b underlines this intuition by showing the relationship of minimum and maximum RTT per flow in a 2D histogram. The figure shows that the maximum RTT significantly differs from the minimum RTT per flow, which further suggests the presence of queuing.

We estimate the queueing delay by evaluating the sRTT range (*i.e.*, max-min) for each connection with at least 10 RTT samples. The implicit assumption is that the minimum RTT accounts for an empty queue and that queueing is the only source of delay variations. In general, additional factors such as route changes and layer 2 delays—particularly prominent in wireless networks—also contribute to delay variations. Since we cannot distinguish these factors from queuing delays, our estimation overestimates queueing and thus yields an *upper bound* on the magnitude of queueing.

We show the PDF of the logarithm of the estimated queueing delay in Figure 1c. Based on whois and DNS information, we split the complete data set into ADSL, Cable, and FTTH users and show their respective queuing delay distribution. Using this scheme, we associate 70% the flows to ADSL users, 1.4% to Cable users, and 0.02% to FTTH users. Most of the user flows experience a modest amount of queueing; 80% of all the flows experience less than 100ms of delay variation. Only 2.8% (1%) experience excessive queueing delays of more than 500ms (1000ms). This corresponds to only 2.5% (2%) of the observed hosts. We also consider user proximity to the CDN caches. Specifically, we consider flows with minimum RTT $\leq$ 100ms. In this setting, even more flows experience modest amounts of queueing: 95% (99.9%) of all connections have a queuing delay of less than 100ms (1sec), respectively.

Recently, the issue of *buffer bloat* has attracted significant attention. The debate is based on observations (*e.g.*, [28]) showing that bufferbloat *can* happen, rather than it *does* happen. Despite this lack of empirical evidence, the bufferbloat argument has been used to motivate engineering changes in Internet standards (*e.g.*, see [18]) and to motivate new AQM approaches (*e.g.*, CoDeL [32]). Two very recent studies examined the magnitude of the problem based on data from 118K [8] and 25K hosts [12], respectively and concluded that the magnitude of bufferbloat is modest.

Our results, based on a much large data set of 80M hosts that is representative for a significant body of Internet users, further



(a) Min, Avg, and Max RTT Distribution

(b) Min vs. Max RTT per Flow

(c) Queueing Delays

Figure 1: Occurrence of queueing in the wild

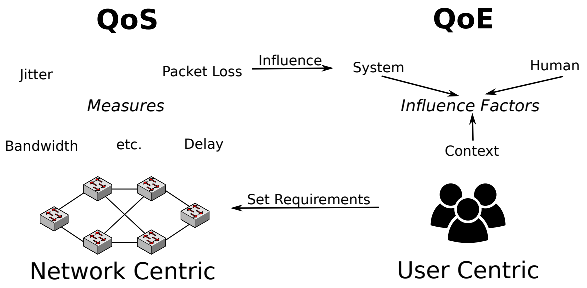


Figure 2: Conceptual difference of QoS and QoE

substantiate these findings. We empirically study whether Internet users at large experience excessive delays and we conclude that excessive delays do occur, but only for a small number of flows and hosts. Thus—despite what is often claimed by the bufferbloat community—our findings further confirm the modest magnitude of excessive queueing delays. One explanation is that uplink capacity in the access, where bufferbloat has been found, is seldom utilized.

Our study of buffering in the wild is the starting point for our evaluation of the impact of buffering on QoE, including the case of excessive buffering (bufferbloat). While we estimate the magnitude of bufferbloat to be modest, its implications on QoE are largely unknown. For instance, a single delayed flow can severely degrade the QoE of an entire HTTP transaction. To shed light on the QoE impact of buffering, we first briefly introduce QoE, and then conduct a multi-factorial testbed study covering a wide range of end-user applications, buffer configurations, and traffic scenarios.

4. QOS $\neq$ QOE

Assessing human quality perception is challenging due to its subjective nature. This challenge is addressed by research on Quality of Experience (QoE) which aims at capturing the "degree of delight of the user of a service. In the context of communication services, it is influenced by content, network, device, application, user expectations and goals, and context of use." [7].

While network performance is typically expressed by QoS metrics, QoS and QoE represent fundamentally different concepts that can influence each other; QoS represents a *network centric* view whereas QoE represents a *user centric* view (see Figure 2). QoE depends on a multidimensional perceptual space that includes *i)* system influence factors (*e.g.*, QoS measures, transport protocols, or device specific parameters), *ii)* human influence factors (*e.g.*, mood, personality traits, or expectations), and *iii)* contextual factors (*e.g.*, location, task, or costs). For an extensive discussion on factors influencing QoE, we refer to [7]. These features are not necessarily independent of each other and do not always have clear mappings, *e.g.*, users tend to give different opinion scores for the same stimulus, *e.g.*, depending on mood, expectation, and memory. While some QoE influence factors include QoS metrics (*e.g.*, packet loss), QoE depends on a larger set of influence factors that cannot be derived from QoS metrics and requires new measures.

To quantify the users' perception of the quality of (network) applications, QoE metrics have been defined and standardized for applications such as VoIP, Video, Web, etc. These metrics are rooted in psychological tests that involved human subjects in the metric *construction phase*. In the *application phase*, however, they allow automatic quality assessments without user involvement; *i.e.*, conclusions on user-experience can be drawn from testbed evaluations *without costly user involvement*. This is an appealing property as it enables the automatic exploration of a large state space that involves a significant number of different workload and buffer size configurations in *controlled experiments*. While desired, involving human subjects is time intensive and thus requires only a small set of conditions to be tested in order to be economically feasible. However, insights obtained in this paper allow sub-sampling of this large state space and therefore enable subjective tests to be conducted in future work.

We base our QoE evaluation on standardized and widely used QoE metrics for quality assessment. Since the metrics depend on the applications, we refer the reader to the corresponding section for a discussion of the used metrics.

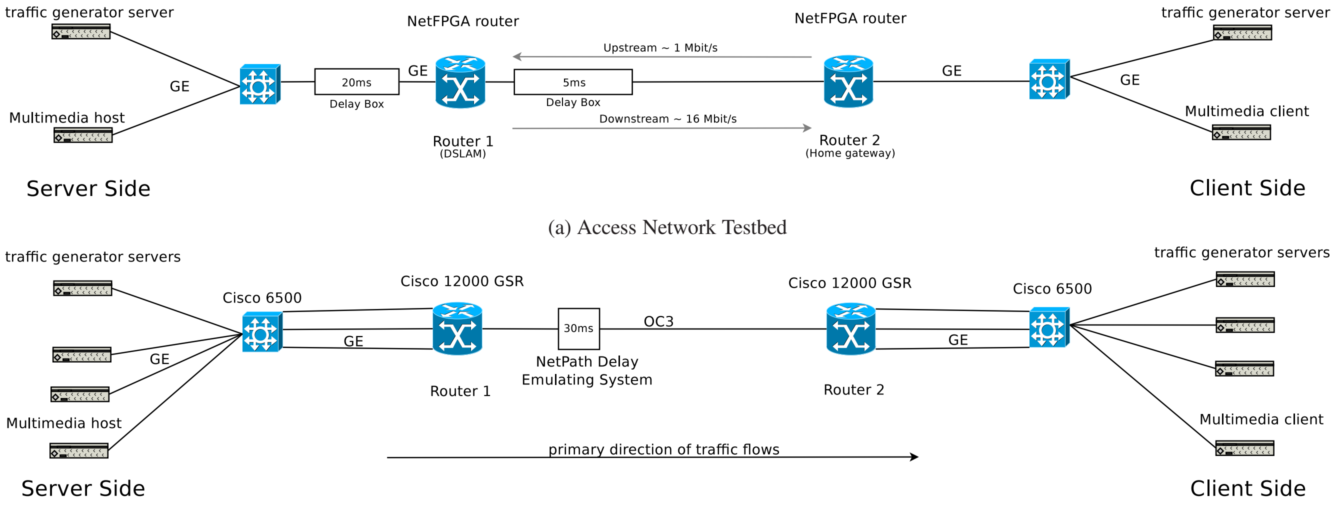
5. METHODOLOGY

We use a testbed driven approach to study the impact of buffer sizes on the user perception (QoE) of common types of Internet applications: *i)* Voice over IP, *ii)* RTP/UDP video streaming as used in IPTV networks, and *iii)* web browsing.

5.1 Testbed Setup

We consider two scenarios: *i)* an access network and *ii)* a backbone or core network. Each scenario is realized in a dedicated testbed as shown in Figure 3 (a) and (b). We use a testbed setup to have full control over all parameters including buffer sizes and generated workload.

As most flows typically experience only a single bottleneck link, both testbeds are organized as a dumbbell topology with a single bottleneck link, configurable buffer sizes, and a client and a server network. The hosts within the server (client) network on the left (right) side act as servers (clients), respectively. In the backbone



(a) Access Network Testbed

(b) Backbone Network Testbed

Figure 3: Access and backbone network testbeds used in the study

case we configured the bandwidth and the delays of all links symmetrically. For the access network we use an asymmetric bottleneck link. In the backbone case we only consider data transfers from the servers to the clients. For the access network we also include data uploads by the clients—as they mainly triggered the bufferbloat debate [19].

The access network testbed, see Figure 3a, consists of two Gigabit switches, four quadcore hosts equipped with 4 GB of RAM and multiple Gigabit Ethernet interfaces. Moreover, two hosts are equipped with a NetFPGA 1 Gb card each. The hosts are connected via their internal NICs to the switch to realize the client/server side network. The NetFPGA cards run the Stanford Reference Router software and are thus used to realize the bottleneck link. Thus the NetFPGA router and the multimedia hosts are located on the same physical host. As the NetFPGA card is able to operate independent of the host, it does not impose resource contention. The right NetFPGA router acts as the home router, aka DSL modem, whereas the left one acts as the DSLAM counterpart of the DSL access networks. To capture asymmetric bandwidth of DSL we use the hardware capabilities of the NetFPGA card to restrict the uplink and downlink capacities to approximately 1 respectively 16 Mbit/s. We use hardware to introduce a 5 ms respectively 20 ms delay between the client (server) network and the routers. The 5 ms delay corresponds to DSL with 16 frame interleaving or to the delays typical for cable access networks [10]. The 20 ms account for access and backbone delays. While we acknowledge that delays to different servers vary according to a network path, a detailed study of path delay variation is beyond the scope of this paper. This is also the reason we decided to omit WiFi connectivity which adds its own variable delay characteristics due to layer-2 retransmissions. Instead, we focus on delay variations induced by buffering.

To be able to scale up the background traffic to the backbone network, see Figure 3b, we include eight hosts, four clients and four servers. Each has again a quadcore CPU, 4 GB of RAM, and multiple Gigabit Ethernet network interfaces. The client/server networks are connected via separate Gigabit switches, Cisco 6500s, to backbone grade Cisco 12000GSR routers. Instead of using 10 Gbit/s and soon to be 100 Gbit/s interfaces for the bottleneck link, we use an OC3 (155 Mb/s nominal) link. The reason for this is that we wanted to keep the scale of the experiments reasonable, this includes, *e.g.*, the tcpdump files of traffic captures. Moreover, scaling down allows us to actually experience bufferbloat given the available memory within the router. We use multiple parallel links between the hosts, the switch, and the router so that it is possible for multiple packets to arrive within the same time instance at the router buffer. With regards to the delays we added a NetPath delay box with a constant one-way delay of 30 ms to the bottleneck link. 30 ms delay roughly corresponds to the one-way delay from the US east to the US west coast. We again note, that the path delays in the Internet are not constant. However, variable path delays are beyond the scope of this paper. Instead we focus on delay variability induced by buffering. Moreover, we eliminate most synchronization potential by our choice of workload (see § 5.2).

To gather statistics and to control the experiments we always use a separate Ethernet interface on the hosts as well as a separate physical network (not shown).

5.2 Traffic Scenarios

We use the Harpoon flow level network traffic generator [38] to create a number of congestion scenarios which range from no background traffic (noBG) to fully overloading (short-overload) the bottleneck link. Congestion causes packets from both the background traffic as well as the application under study to be queued or dropped just before the bottleneck link. Depending on the fill grade of the buffer, the size of the buffer, and the link speed, this will increase the RTT accordingly (see Table 2). Overall, we use 12 scenarios for the access testbed and 6 for the backbone. We consider more for the access to distinguish on which links (*i.e.*, upstream, downstream, or both) the congestion is subjected to.

In terms of traffic that imposes the congestion we distinguish two different kinds of scenarios (see Table 1): (*i*) long-lived TCP flows (long) and (*ii*) long-tailed file sizes to be able to resemble self-similar traffic as seen in today's networks (e.g., in core networks). For the latter, we choose Weibull distributed file sizes with a shape of 0.35 as their mean and standard deviation are finite as opposed to those of the often used Pareto distributions with a shape > 2. The generated traffic results in a mixture of bursty short-lived and long-lived flows with a mean of 50 KB. As the number of short flows dominates the number of long flows we refer to these scenarios as "short".

Testbed	Name	Flow Interarrival Distribution	File Size Distribution	# Sessions		Concurrent Flows	Link Utilization [%]				Packet Loss [%]		Description
				Up	Down		Mean		Sd				
							Up	Down	Up	Down	Up	Down	
Access	noBG	—	—	—	—	—	—	—	—	—	—	—	No bg. traffic
	short-few	exp-a	weibull	1	—		98.9	0.3	0.7	0.1	34.7	0	Upstream
				1	8		95	8.5	5.6	15.2	58.6	0.7	Bidirectional
				—	8		27.8	44.1	13.7	25.1	1.4	3	Downstream
	short-many	exp-a	weibull	1	—		98.9	0.3	0.7	0.1	33.1	0	Upstream
				1	16		93.3	10.7	4.3	20.1	60.9	1.3	Bidirectional
				—	16		53.8	78.7	12.8	23.5	4	4.5	Downstream
	long-few	—	infinite	1	—		99	0.2	0.7	0.0	1	0	Upstream
				1	8		71.9	83.1	8.9	12.6	41.7	0.6	Bidirectional
				—	8		39.5	99.9	1.9	0.6	0.1	0.5	Downstream
	long-many	—	infinite	8	—		98.9	0.3	0.7	0.0	14.4	0.0	Upstream
				8	64		83.8	61.8	11.2	26.4	60.7	0.2	Bidirectional
				—	64		68.5	99.6	3.9	4.9	0.03	9.3	Downstream
Backbone	noBG	—	—	—	—	—	—		—		—		No bg. traffic
	short-low	exp-b	weibull	—	$3 * 10$	18	16.5		11.6		0		
	short-medium	exp-b	weibull	—	$3 * 30$	49	49.5		18.8		0		
	short-high	exp-b	weibull	—	$3 * 60$	206	98		6.5		0.2		
	short-overload	exp-b	weibull	—	$3 * 256$	2170	99.7		2.2		5.2		
	long	—	infinite	—	$3 * 256$	675	99.7		0.1		3.8		

Table 1: Workload configuration for both testbeds, where the flow interarrival time distributions are specific to the access and backbone testbed; exp-a has a mean of 2 sec and exp-b a mean of 1 sec. The file size distribution is defined as weibull(shape=0.35, scale=10039), resulting in a mean flow size of 50 KB. The number of parallel flows at the bottleneck link is shown by their mean. Link utilization and loss measures are obtained for buffers configured according to the BDP.

For scenarios with long-lived flows (long) we use flows of infinite duration. In this case the link utilization is almost independent of the number of concurrent flows. For long-tailed file sizes the workload of each scenario is controlled via the number of concurrent sessions that Harpoon generates. A session in Harpoon is supposed to mimic the behaviour of a user [38] with a specific interarrival time, a file size distribution, and other parameters. We used the default parameters except for the file size distribution. In addition, we rescaled the mean of the interarrival time for the access network, as Harpoon's default parameters are geared towards core networks with a larger number of concurrent flows. To impose different levels of congestion we adjusted the number of sessions for the backbone scenario to result in low, medium, high, and overload scenarios which correspond to link utilizations as shown in Table 1. For the access network we distinguish between few and many concurrent flows which results in medium and high load for the downstream direction and high load for the upstream, see Table 1.

We checked that all hosts are using a TCP variant with window scaling. Due to the Linux version used the background traffic uses TCP-Reno in the backbone and TCP BIC/TCP CUBIC for the access. However, note that this does not substantially impact the QoE results as the applications VoIP and video rely on UDP and the Web page is relatively small. Moreover, since the results are consistent it suggests that using a TCP variant optimized for high latency does not change the overall behavior even when the buffers are large.

5.3 Buffer Configurations

One key element of our QoE study is the buffer size configurations. Buffers are everywhere along the network path including at the end-hosts, the routers, and the switches. The most critical one is at the bottleneck interface, the only location where packet loss occurs. Therefore we focus on these and rely on default parameters for the others. For the bottleneck we choose a range of different buffer sizes, some reflect existing sizing recommendations, some are chosen to be small other large in order to capture extremes. Table 2 summarizes the buffer size configuration in terms of number of packets and shows the corresponding queuing delays.

For the access network we choose buffer sizes of powers of two, ranging from 8 to 256 packets. 256 is the maximum supported buffer size by the Stanford Reference Router software. For our choice of an asymmetric link (recall 1 Mbps uplink/16 Mbps downlink) the bandwidth-delay product (BDP) corresponds to roughly 8 and 64 packets, respectively. Since this set of buffer sizes yields delays up to buffer bloat, we consider the buffer configurations to approximate home router behaviour.

Access					Backbone		
Buffer Size (Pkts)	Uplink		Downlink		Buffer Size (Pkts)	Delay (ms)	Scheme
	Delay (ms)	Scheme	Delay (ms)	Scheme			
8	98	≈ BDP	6	min	8	0.6	≈ TinyBuf
16	198		12		28	2.2	Stanford
32	395		24		749	58	BDP
64	788		49	≈ BDP	7490	580	10 × BDP
128	1,583		97				
256	3,167	max	195	max			

Table 2: Buffer size configurations and corresponding maximum queuing delays for both testbeds (full sized packets).

For the backbone network we use *i)* the same minimum buffer size of 8 packets, which resembles the TinyBuffer scheme [17], depending on the largest congestion window achieved by the workloads. In addition, we use *ii)* 749 full-sized packets which corresponds to the BDP formula given an RTT of 60 ms, *iii)* 28 packet which corresponds to the Stanford scheme [9], *i.e.*, BDP$/\sqrt{n}$, where $n = 3 * 256$ is the maximum number of concurrent for short-low, short-medium, short-high, and long (see Table 1), and *iv)* $10 \times$ BDP packets an excessive buffering scheme.

6. QOS: BACKGROUND TRAFFIC

To highlight the potential importance of the buffer configuration on latencies, network utilization, and packet loss—the typical QoS values—we start our study with a detailed look at the background traffic. While the story is relatively straight forward for the backbone scenario, and captured in Table 1, it is more complicated for the access network as the number of concurrent flows is smaller and there are subtle interactions between upstream and downstream.

To illustrate how the workloads and buffer sizes effect real-time applications, we conducted experiments to measure the latency introduced by the buffers. For this purpose we use the detailed buffer utilization statistics of the FPGA cards. Figure 4 shows the corresponding mean delays as heatmaps. We use three different heatmaps: one each for downstream/upstream workload only and one for combined up- and downstream workload. Each heatmap has two subareas—one for upstream at the top and one for downstream at the bottom. Each heatmap cell show the mean delay for a specific buffer size configuration and workload scenario, measured over two hours. The color of the heatmap cells correspond to categories of the ITU-T Recommendation G.114 which classifies delays based on their potential to degrade the QoE of interactive applications: green (light gray) is acceptable, orange (medium gray) problematic, and red (dark gray) causes problems.

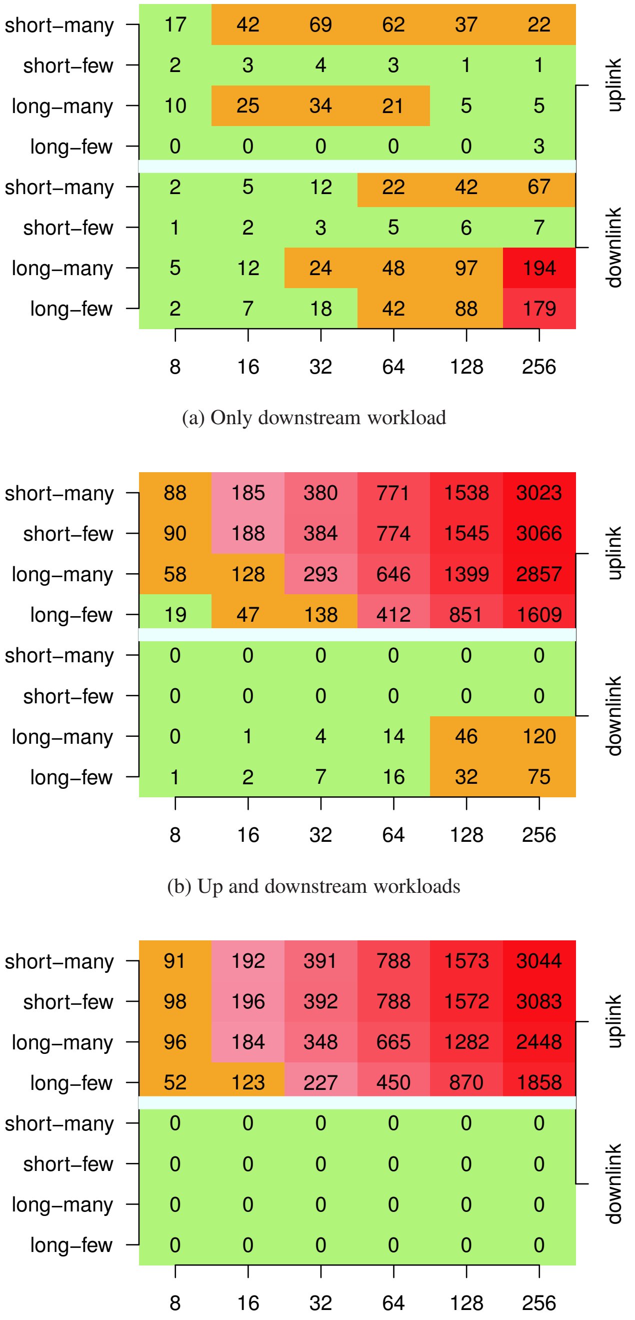
In principle, we see that larger buffers sizes can increase the delays significantly independent of the workload. For the downlink direction the maximum delay is less than 200 ms. However, this can differ for the uplink direction. In particular, we observe delays of up to three seconds for larger–over-sized–buffers when the upstream is used for the uplink direction. This is almost independent of the workload! Overall, these delays are consistent with observations by Gettys [6] which started the bufferbloat discussion.

Given these high latencies, we investigate the link utilization. Figure 5 shows a boxplot of the link utilization for the various buffer sizes in the scenario with simultaneously downloads and uploads (bidirectional workloads). The left/right half focuses on the downlink/uplink utilization. The uplink utilization is almost 100% while the downlink utilization ranges from 20% to 100%. Consistent with related work, we see that very small buffers can lead to underutilization while very large buffers can lead to large delays.

Comparing these link utilizations to those with no upstream workload (not shown) we find that, for bidirectional workloads, the buffer configurations below the BDP do not always fully utilize the downlink direction. Buffer sizes that correspond to the BDP yield full downlink utilization in the absence of upload workload, but not with concurrent download and upload activities. This phenomena can be explained by the queuing delay introduced by bloated uplink buffers that *virtually* increase the BDP thus rendering the downlink under-buffered. Related work coined the problem of bidirectional TCP flows that influence each other *data pendulum* [23]. In contrast to related work, our analysis highlights interdependencies between buffers and suggests that buffers should not be sized independent of each other.

The phenomena of low link utilization can be mitigated by counter-intuitively "bloating" the downlink buffer. Considering the delays observed in Figure 4b, the BDP increases beyond the initial buffer size of 64 to 835 full sized packets. Note, that we can get full link utilization for buffers of smaller than 835 packets as we have a sufficient number of concurrent flows active.

In summary, the latency introduced by the buffers in home routers, aka, the uplink, might not only i) harm real-time traffic applications (due to excessive buffering), but also ii) drastically reduce TCP performance (due to insufficient buffering) in case of bidirectional workloads in asymmetric links. In effect it invalidates the buffer dimensioning assumptions due to the increase in RTT.



(a) Only downstream workload

(b) Up and downstream workloads

(c) Only upstream workload

Figure 4: Mean queuing delay (in ms) for the access networks testbed with different buffer size (x-axis) and workload (y-axis) configurations. Delays that significantly degrade the QoE of interactive applications (ITU-T Rec. G.114) are colored in red.

7. VOICE OVER IP

We start our discussion of application QoE with Voice over IP (VoIP). In IP networks speech signals can be impaired by QoS parameters (*e.g.*, packet loss, jitter, and/or delay), talker echo, codec and audio hardware related parameters, etc. Regarding QoS parameters, packet losses directly degrade speech quality as long as forward error correction is not used as is typical today. Network jitter can result in losses at the application layer as the data arrives after its scheduled playout time. Moreover, excessive delays impairs any bidirectional conversation as it changes the conversational dynamics in turn taking behavior.

7.1 Approach

We use a set of 20 speech samples recommended by the ITU [4] for speech quality assessment. Each sample is an error-free

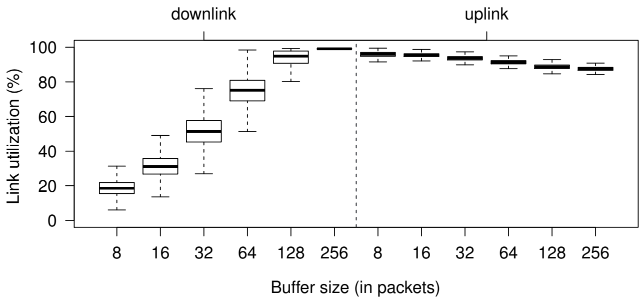


Figure 5: Link utilization for an asymmetric access link with various buffer sizes. The uplink and the downlink are simultaneously congested by 8 and 64 long-lived TCP flows, respectively.

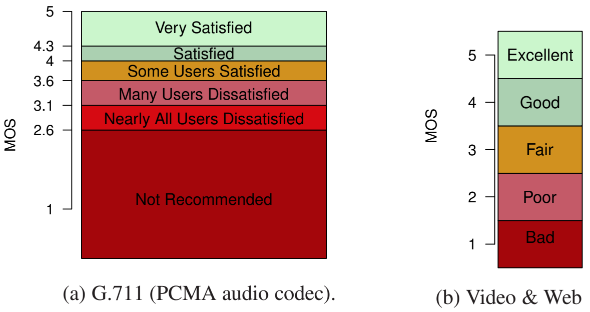
recording of a male or female Dutch speaker, encoded with G.711.a (PCMA) narrow-band audio codec, and lasts for eight seconds. Each of the 20 samples is automatically streamed, using the PjSIP library, over our two evaluation testbeds, see § 5 and subjected to the various workloads. PjSIP uses the typical protocol combination of SIP and RTP for VoIP. We remark that we do not consider other situational factors such as the users' expectation (*e.g.*, free vs. paid call) [31] which can also affect the perceived speech quality (see § 4). For the VoIP QoE assessment, we separately evaluate speech signal degradations and conversational dynamics, using two widely used and standardized QoE models: PESQ and E-Model. Individual scores are combined to the final QoE score.

Speech signal degradations. To assess the speech quality of each received output audio signal, relative to the error-free sample signal, we use the Perceptual Speech Quality Measure (PESQ) [2] as standardized model. PESQ takes as input both the error-free audio signal and the perturbed audio signal, and computes the QoE score z_1. Note that while z_1 is *influenced* by loss and jitter, the QoE estimation is *signal based* and *not a function of QoS parameters*. The influence of loss and jitter on z_1 can therefore not be quantified.

Conversational dynamics. The PESQ model only accounts for the perceived quality when listening to a remote speaker but does not account for conversational dynamics, *e.g.*, for humans taking turns and/or interrupting each other. This can be impaired by excessive delays and thus can degrade the quality of the conversation significantly [31, 26, 34, 35]. Thus, according to the ITU-T recommendation G.114 one-way delays should be below 150 ms (or at most 400 ms).

Therefore, we measure the packet delay during the VoIP calls. We now use the delay impairment factor of the ITU-T E-Model [3] to get a score z_2. We remark that even though z_2 is computed using a standardized and widely used model, it is subject to an intense debate within the QoE literature as there is a dispute about the impact of delay on speech perception [26, 34, 16]. Among the reasons is that the delay impact depends on the nature of the conversational task (e.g, reading random numbers vs. free conversation) as well as the level of interactivity required by the task [26]. Thus, there can be mismatches between the quality ratings of the E-Model and tests conducted with subjects.

Overall score. The range of the score z_1, which captures loss and jitter, is $[1, 5]$. We remap it to $[0, 100]$ according to [41]. The range of the score z_2, capturing the delay impairment, is $[0, 100]$. Note, the semantics of z_1 and z_2 are reversed: a large value for z_1 reflects an excellent quality; however, a large value for z_2 reflects a bad quality, and vice-versa. We combine the two scores to an overall one as follows: $z = \max\{0, z_1 - z_2\}$. Thus, if z_1 is good (*i.e.*, due to negligible loss and jitter), but the z_2 is bad (*i.e.*, due to large delays), then the overall score z is low, reflecting a poor quality and vice-versa. Finally, we map z to the MOS scale $[1, 5]$ according to the ITU-T recommendation P.862.2, see Figure 6a; in the end, low values correspond to bad quality and high values to excellent quality.



(a) G.711 (PCMA audio codec).

(b) Video & Web

Figure 6: MOS scales used in this paper

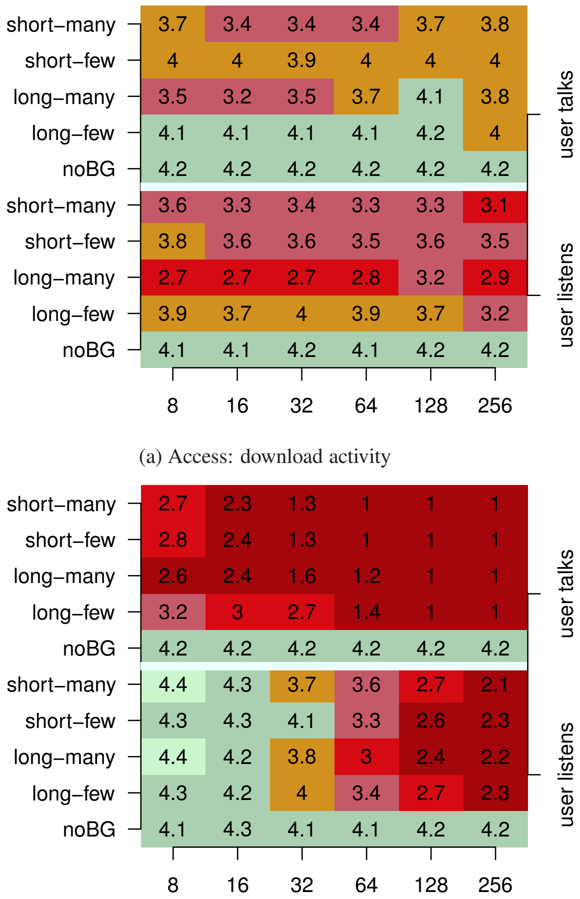
7.2 Access networks results

Figures 7a and 7b show heatmaps of the median call quality (MOS) for the access networks. Each cell in the heatmap shows the median MOS of 200 VoIP calls (each speech sample is send 10 times) per buffer size (x-axis) and workload scenario (y-axis) combination. The heatmap is colored according to the color scheme of Figure 6a. The heatmap is divided into two parts (i) when user talks (upper part) and (ii) when the user listens to the remote speaker (bottom part).

The baseline results, namely the ones without background traffic are shown in the bottom row of each heatmap part, labeled noBG. They reflect the achievable call quality of the scenarios. As all of them are green, we can conclude that in principle each scenario supports excellent speech quality and that any impairment is due to congestion and not due to the buffer size configuration per se.

Download activity. Figure 7a focuses on the scenarios when there is congestion in the downlink. As there is no explicit workload in the uplink, one may expect that only the "user listens" part is effected but not the "user talks" part. This is only partially true as the "user talks" part of the heatmap shows deviations of up to 0.8 MOS points from the baseline score. These degradations are explained by the substantial number of TCP ACK packets, reflected by higher link utilizations (not shown). Recall, the uplink capacity is 1/16th only of the downlink capacity.

The degradations in "user listens" part of the heatmap are, as expected, more pronounced then for the "user talks" part. However, there are also significant differences according to the workload and the buffer configurations. For instance, with buffers sizes of 64 packets the long-many workload yields a median MOS of 2.8, whereas the long-few workload yields a median MOS of 3.5. Interestingly, even though the short-few workload does not fully utilize the downlink, *i.e.*, less than 50% (not shown), it gets scores worse than a workload with higher link utilization, *e.g.*, long-few. This is due to the higher jitter that is imposed by the large changes in link utilization and thus in the buffer utilization. With regards to buffer sizes we in general observe the worst scores for the larger buffer configurations, *i.e.*, 256 packets due to the added delays. However, the best scores only deviate by 0.7 MOS points from this worst score (*e.g.*, for the 8 packets buffer), suggesting that smaller buffers do not significantly improve audio quality.



(a) Access: download activity

(b) Access: upload activity

Figure 7: VoIP Access: Median Mean Opinion Scores (MOS) for voice calls with different buffer size (x-axis) and workload (y-axis) configurations. The heatmaps for the access networks include inbound calls (user listens) and outbound calls (user talks).

We conclude that the level and kind of workload has a more significant effect than buffer size.

Upload activity. Figure 7b focuses on the scenarios when there is congestion on the uplink. According to the above reasoning one would therefore only expect degradations for the "user talks" part. This is not the case. The MOS in the "user listens" part of the heatmap decreases by 0.5 to 2 from the baseline results for all scenarios with buffer sizes ≥ 64. The reason for this is that the delays added by the excessive buffering in the uplink also degrade the overall score due to the delay impairment factor z_2. Since this factor expresses the conversational quality, it does not only effect the "user talks" but also the "user listen" part sent over the (non-congested) downlink.

Excessive delays added by the buffers also explain the significant degradation of MOS values from 4.2 to $1 - 1.4$ for the "user talks" part. Due to the congestion, packet loss is also significant for all scenarios. This is why the best MOS value is limited to 3.2 even for short buffer configurations.

In the context of the bufferbloat discussion, Figure 7b corroborates that excessive buffering in the uplink yields indeed bad quality scores. Reducing the buffer sizes results in better MOS and contributes to mitigate the negative effects of the large delays introduced by the uplink buffer, *e.g.*, the difference in the MOS for an inbound audio can be as high 2.5 points (long-many workload).

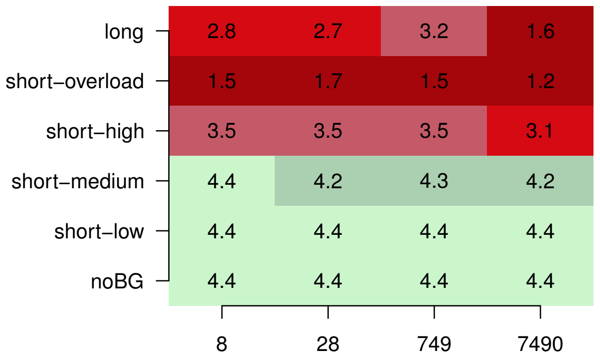


Figure 8: VoIP Backbone: Median Mean Opinion Scores (MOS) for voice calls with different buffer size (x-axis) and workload (y-axis) configurations.

Combined upload and download activity. Scenarios (plot not shown) with upload and download congestion show similar results to scenarios with only uploads. Here as well the delays introduced by the uplink buffer dominate in both "user talks" and "user listens" parts. However, with combined upload and download activity, the "user listens" is slightly more degraded than with only upload activity. The reason for this is additional background traffic in the downlink that interacts with the voice call. For instance, with buffers configured to 16 packets, the long-few shows an additional degradation of 0.8 MOS points (thus mapping to a different rating scale).

Limiting access congestion by isolating VoIP calls in a separate QoS class—as often implemented for ISP internal services but not Internet-wide—is therefore a good strategy.

7.3 Backbone networks results

Similar to the access network scenario, we show the voice quality in the backbone network scenario as a heatmap in Figure 8. The heatmap shows the median MOS for unidirectional audio from the left to the right side of the topology per buffer size (x-axis) and workload scenario (y-axis). Each cell in the heatmap is based on 2000 VoIP calls. Here, each speech sample is send 100 times which is possible as the total number of scenarios is smaller. As in the access network scenario, the bottom row label noBG shows the baseline results for an idle backbone without background traffic.

While the effects of the buffer size are less pronounced, the nature of the background traffic (long vs. short-*) and the link utilization (short-low to short-overload) are more significant. The type of workload can drastically degrade the quality score. Concretely, low to medium utilization levels as imposed by the scenarios short-low and short-medium, respectively, are close to the baseline results. In contrast, more demanding workloads such as the scenarios short-high and long, leading to higher link utilizations, and result in more than 1 point reductions in the MOS scale. Further, the aggressiveness of the workload further decrease the quality; the median MOS for the short-overload workload is 1.5 and thus significantly lower than for short-high and long that also lead to high link utilizations.

In general, the quality scores are, on a per workload basis, fairly stable across buffer-configurations below the BDP (749 packets). In these cases, we only observe small degradation of 0.4 points for the scenario long workload for the smallest buffer configuration. However, buffer configuration larger than the BDP, i.e, 7490 packets, lead to excessive queueing delays. As in the access network scenario, excessive delays lead to significant quality degradations of the z_2 delay impairment component. For example, the scores corresponding to the scenarios long and short-overload workloads have MOS values of almost half of their counterpart with the BDP configuration.

7.4 Key findings for VoIP QoE

We find that VoIP QoE is substantially degraded when VoIP flows have to compete for resources in congested links. This is particularly highlighted in the backbone network scenario, where low to medium link utilizations yields good QoE and high link utilization ($> 98\%$) degrade the QoE. In the case of the latter, the congestion leads to insufficient bandwidth on the bottleneck link that affects the VoIP QoE.

For access networks we show that, due to the asymmetric link capacities, the different audio directions can yield different QoE scores. For instance, in one direction (*e.g.*, user talks) the speech quality might be acceptable, while it is impaired for the other (*e.g.*, remote speaker talks) or vice-versa. Moreover, the speech quality is much more sensitive to congestion on the upstream direction than the downstream one. Due to the light queueing delays introduced by bloated buffers in the uplink, maintaining a conversation can be challenging in the presence of uplink congestion.

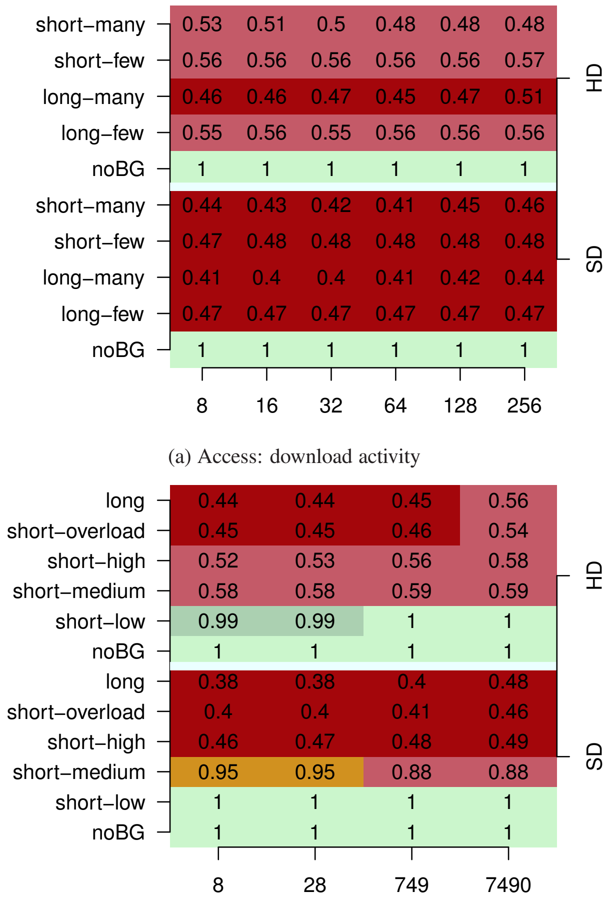
For both access and backbone networks, configuring small buffers can results in better QoE. However, our results highlight that this may not suffice to yield "excellent" quality ratings. Thus, we advocate to use QoS mechanisms to isolate VoIP traffic from the other traffic. This is already common for ISP internal services but not for ISP external services.

8. RTP VIDEO STREAMING

Next, we explore the quality of video streaming using the Realtime Transport Protocol (RTP) which is commonly used by IPTV service providers. RTP streaming can be impaired by packet loss, jitter, and/or delay. Again packet losses directly degrades the video as basic RTP-based video streaming *typically* does not involve any means of error recovery. Network jitter and delays result in similar impairments as with voice and include visual artifacts or jerky playback. However, they depend on the concrete error concealment strategy applied by the video decoder.

8.1 Approach

We chose three different video clips from various genres as reference. Each video has a length 16 seconds. They are chosen to be representative of various different kinds of TV content and vary in level of detail and movement complexity. Thus, they result in different frame-level properties and encoding efficiency; *A)* an interview scene, *B)* a soccer match, and *C)* a movie. Each video is encoded using H.264 in SD (4 Mbps) as well as HD (8 Mbps) resolution. Each frame is encoded using 32 slices to keep errors localized. This choice of our encoding settings is motivated by our experiences with an operational IPTV network of a Tier-1 ISP.



(a) Access: download activity

(b) Backbone

Figure 9: Median MOS (color) and SSIM (text) for HD and SD RTP video streams with different buffer size (x-axis) and workloads (y-axis).

We use VLC to stream each clip using UDP/RTP and MPEG-2 Transport Streams. Without any adjustment VLC tries to transmit all packets belonging a frame immediately. This leads to traffic spikes exceeding the access network capacity. In effect VLC and other streaming software propagate the information bursts directly to the network layer. As our network capacity, in particular for the access, is limited we configured VLC to smooth the transmission rate over a larger time window as is typical for commercial IPTV vendors. More specifically, we decided to use a smoothing interval (1 second) that ensures that the available capacity is not exceeded in the absence of background traffic. The importance of smoothing the sending rate is often ignored in available video assessment tools such as EvalVid, making them inapplicable for this study.

We note that Set-top-Boxes in IPTV networks often use proprietary retransmission schemes that request lost packets once [24]. Due to the unavailability of exact implementation details we do not account for such recovery. Our results thus present a baseline in the expected quality; however, systems deploying active (retransmission) or passive (FEC) error recovery can achieve higher quality.

We use two different full-reference metrics, PSNR and SSIM, to compute quality scores from the original and the perturbed video stream. While not considered as QoE metric, PSNR (Peak Signal Noise Ratio) enables a quality ranking of the same video content subject to different impairments [43, 27]. However, it does not necessarily correlate well with human-perception in general settings. SSIM (Structural SIMilarity) [47] has been shown to correlate better with human perception [48]. We map PSNR and SSIM scores to quality scores according to [49].

8.2 Access network results

We show our results as heatmap in Figure 9a. The heatmap shows the quality score for video C sent 50 times per buffer size (x-axis) and workload (y-axis) combination. Each cell shows the median SSIM score and is colored according to the corresponding MOS score (see Figure 6b); a SSIM score of 1 expresses excellent video quality, whereas 0 expresses bad quality. The upper and the bottom parts of the heatmap correspond to the results of HD and SD video streams, respectively. We omit quality scores obtained for the PSNR metric as they yield predicted scores similar to those obtained by SSIM. Also, as we focus on IPTV networks where the user consumes TV streams, no video traffic is present in the upstream. For this reason, we only show results for workloads congesting the downlink.

To show the achievable quality for all buffer size configurations in the absence of background traffic, we show baseline results in rows labeled noBG. In these cases, the video quality is not degraded due to the absence of congestion.

In the presence of congestion, however, the SD video quality is severely degraded, expressed by a "bad" MOS score. This holds regardless of the workloads and the buffer configuration; the link utilization by all of the workloads cause video degradation due to packet loss in the video stream. We observe that even a low packet loss rate can yield low MOS estimates. Moreover, much higher loss rates (one order of magnitude bigger) can yield the same estimates. For instance, although both scenarios, long-few and long-many, have a similar SSIM and MOS score for buffers sized to 256 and 8 packets respectively, they show different packet loss rates of 0.5% and 12.5%.

In comparison to the SD video, degradations in HD videos are less pronounced although, in some cases, the packet loss rate is higher. For instance, the packet loss rate for HD and SD video streaming is, with the long-few workload and buffers sized to 256 packets, 2.6% and 1.3% respectively. However, the HD video stream obtains a better MOS score. This interesting phenomena can be explained by the higher resolution and bit-rate of HD video streams, which reduce the visual impact of artifacts resulting from packet losses during video streams.

In the case of UDP video streaming in access networks, what matters is the available bandwidth, not the buffer size. Moreover, even though buffers regulate the trade-off between packet losses and delay, they have limited influence on the quality from the perspective of an IPTV viewer.

8.3 Backbone network results

Similar to the previous access network scenario, we show the video quality scores obtained for the same video C as a heatmap in Figure 9b, both for SD and HD resolution. Each cell of the heatmap shows the median SSIM score and is colored according to the corresponding perceptive MOS score (see Figure 6b). As in the previous scenario, the video was sent 50 times per buffer size (x-axis) and workload (y-axis) configuration. We omit PSNR quality scores as they are similar to the SSIM quality scores.

As in the access network scenario, the bottom row labeled noBG shows the baseline results for an idle backbone without background traffic. Similarly, workloads that do not fully utilize the bottleneck link, *i.e.*, short-low, lead to optimal video quality, as expressed by an SSIM score of 1. The reason is that the available capacity in the bottleneck link allows streaming the video without suffering from packet losses.

First quality degradations are observable in the short-medium scenario, where the quality decreases with increasing link utilization. In this scenario, workloads achieve full link utilization for 749/7490 buffers more often than for the 8/28 buffer configurations. It results in higher loss rates for the video flows lowering the quality and is more pronounced for the HD videos which have higher bandwidth requirements.

Workloads that sustainably utilize the bottleneck link, *i.e.*, short-high, short-overload, and long, yield bad quality scores due to high loss rates. These scenarios provide insufficient available bandwidth to stream the video without losses. Increasing the buffer size helps to decrease the loss rate, leading to slight improvements in the SSIM score.

Comparing the obtained quality scores among the three different videos leads to minor differences in quality scores. These differences result from different encoding efficiencies that cause different levels of burstiness in the streamed video. However, the quality scores of all video clips lead to the same primary observation: quality mainly depends on the workload configuration and decreases with link utilization. Increasing the buffer size helps to lower the loss rate and therefore to marginally improve the video quality.

8.4 Key findings for RTP video Quality

Our results indicate a roughly binary behavior of video quality: *i)* when the bottleneck link has sufficient available capacity to stream the video, the video quality is good, and *ii)* otherwise the quality is bad. In between, if the background traffic utilizes the link only temporarily, the video quality is sometimes degraded. This results in an overall degradation that increases with link utilization. Using HD videos yields marginally better quality scores even though they use higher bandwidth. We find that the influence of the buffer size is marginal as delay does not play a major role for IPTV. What mainly matters is the available bandwidth. We did not include quality metrics relevant for interactive TV or video-calls. We further note that our results represent a baseline quality achievable without error recovery. Error recovery (e.g., retransmissions) will increase the overall quality.

9. WEB BROWSING

We next move to web browsing, our last application under study. The web browsing experience (WebQoE) can be quantified by two main indicators [14]. One is the *page loading time* (PLT), which is defined as the difference between a Web page request time and the completion time of rendering the Web page in a browser. Another is the time for the first visual sign of progress. In this paper we consider PLT of information retrieval tasks, for which there exists an ITU QoE model (*i.e.*, G.1030 [5]) to map page loading times to user scores.

We note that WebQoE does not directly depend on packet loss artifacts, but rather on the completion time of underlying TCP flows. Thus, factoring in various workloads and buffer sizing configurations—which influence the TCP performance—is particularly relevant for understanding WebQoE from a network only perspective. Given that the PLT as measured in a browser can be approximated from flow completion times as parameter, is sometimes considered as a QoS parameter. Since the applied G.1030 model logarithmically maps PLT to QoE, it can be misbelieved QoS parameters can (always) be mapped to QoE. We therefore note that other QoE models are of higher complexity as different input parameter are used that cannot be directly derived from a QoS parameters, e.g., speech signals as used in Section 7.

9.1 Approach

To evaluate the WebQoE, we map the PLT to a user score z by using the ITU Recommendation G.1030 [5] specified for web information retrieval tasks. We consider the one-page version of the ITU model, which logarithmically maps *single* PLT's to scores in the range $z \in [1, 5]$ (*i.e.*, 5:excellent, 4:good, 3:fair, 2:poor, 1:bad, as shown in Figure 6b). This mapping uses six seconds as the maximum PLT, *i.e.*, mapping to a "bad" QoE score. The minimum PLT—mapping to "excellent"—is set to 0.56 (0.85) seconds for access (backbone) scenario, due to different RTTs.

We remark that WebQoE research has advanced beyond factors captured in the applied ITU G.1030 [5] model. This concerns the impact of distraction factors such as noise or traffic on quality perception [21], task and content dependent factors [39], or task completion times and loading pattern [40]. These advances have, however, not yet converged to an revised model that is applicable in this study. Beyond additional factors, WebQoE research addresses the need for interactive web use that goes beyond information retrieval tasks which follow request response pattern [15]. We remark that no QoE models fully addresses interactive web usage (such as AJAX requests), which is why we must leave this aspect for future work. For the information retrieval scenario considered in this section, however, recent findings suggest the logarithmic dependency of waiting time and QoE [15, 40]—as used in applied G.1030 model—to remain valid. We thus stick to using ITU Recommendation G.1030 [5] as the current standard for assessing WebQoE.

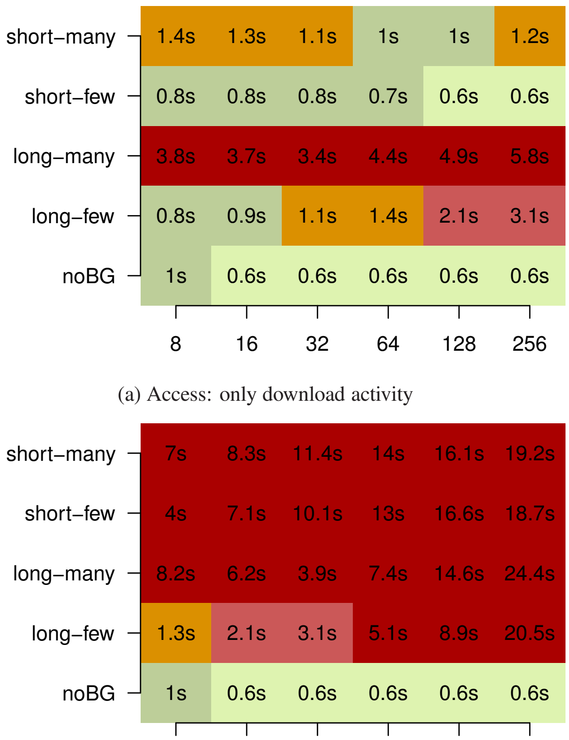
To measure the PLT's, we consider a single static web page, located in one of the testbed servers, and consisting of: one html file, one CSS file, and two medium JPEG images (sized to 15, 5.8, 30, and 30 KB, respectively). The web page is loaded within 14 RTTs, including the TCP connection setup and teardown. Choosing a relatively small web page size was inspired by the frequently accessed Google front page designed to quickly load. To retrieve this web page we use the *wget* tool which measures the transfer time. *wget* is configured to sequentially fetch the web page and all of its objects in a single persistent HTTP/1.0 TCP connection without pipelining. We point out that, as static web pages have constant rendering times, it suffices to rely on *wget* rather than on a specific web browser.

To further analyze the page retrieval performance, we rely on full packet traces capturing the HTTP transactions. We analyze the loss process of the captured TCP flows using the *tcpcsm* tool estimating retransmission events. We further measure the RTT during each experiment. We denote PLTs as RTT dominated if a significant portion of the PLT consists of the RTT component expressed by $14 * RTT$. Similarly, we denote PLTs as loss dominated if the increase in PLT can be mainly attributed to TCP retransmissions.

9.2 Access network results

Figures 10a and 10b show heatmaps of the median web browsing quality (MOS) for the access network. Each cell in the heatmap shows the median PLT of 300 web page retrievals per buffer size (x-axis) and workload scenario (y-axis) combination. The heatmap is colored according to Figure 6b.

The baseline results, namely the ones without background traffic, are shown in the bottom row of each heatmap part, labeled noBG. The fastest PLT that can be achieved in this testbed is $\approx$ 0.56s. As all of the cells are green (light gray), we can conclude that in principle each scenario almost supports excellent browsing quality and that any impairment is due to congestion. In this respect, it turns out that, even without background traffic, the WebQoE can be degraded by (too) small buffers, *e.g.*, 8 packets. Due to packet losses causing retransmissions, the PLT is increased to 1 second thereby changing the user perceived quality.



(a) Access: only download activity

(b) Access: only upload activity

Figure 10: WebQoE Access: Median MOS (color) and page loading times (text) with different buffer size (x-axis) and workloads (y-axis).

Download activity. Figure 10a focuses on the scenarios when there is congestion on the downlink. For the short-few scenario the downlink is not fully utilized, thus most scores do not deviate much from the baseline results. With this type of moderate workload browsing can benefit from the capacity of large buffers to absorb transient bursts and reduce packet losses. For instance, configuring the buffers size to 256 packets reduces the PLTs to the baseline results (as opposed to PLTs of 0.8s for the smallest buffer configuration). Likewise, for the short-many scenario, which involves more competing flows and imposes a higher link utilization, big buffers generally reduce PLTs. As the queueing delays for these scenarios are not excessive, *i.e.*, they are bounded by 192 ms, see Table 2, large buffers do in fact improve the QoE by limiting the loss rate.

Bufferbloat is visible for the long-few scenario, where the median PLT increases with the buffer size, as the PLT is dominated by RTTs caused by large queueing delays. As for the previous scenario, the effects of various buffer sizes are clearly perceived by the end-user (yet in a different manner).

In contrast, the buffer size does not change the WebQoE in the long-many scenario. The larger number of competing flows reduces the per-flow capacity and lets the PLT increases beyond the users' acceptance threshold. Therefore, the perceived QoE, in contrast to the previous configuration, can not be improved by adjusting the buffer size. Nevertheless, from a QoS perspective, configuring an appropiate buffer size can let web pages to load 2 seconds faster. This is not as straightforward since it involves considering the tradeoff between small buffers (packet losses) and large buffers (combined effect of packet losses and large RTTs).

Upload activity. Figure 10b focuses on the scenarios when there is congestion on the uplink. As expected, congesting the uplink seriously degrades the link overall performance and thereby also the WebQoE. The perceived quality is degraded to the min-

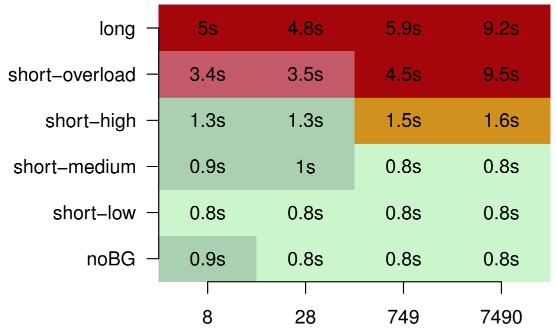


Figure 11: WebQoE Backbone: Median MOS (color) and page loading times (text) with different buffer size (x-axis) and workloads (y-axis).

imum for every buffer size configuration of the scenarios short-many, short-few, and the long-many. The only scenario where the browsing experience is slightly more acceptable is the long-few scenario if buffers are small. Such configuration reduces the median PLT from 20 to 1.3 seconds, which maps to a *fair* quality rating.

From a QoS perspective, the figure shows that the PLT and the buffer size are strongly correlated to the QoE. A wise decision on the dimensioning of the buffers can reduce the PLT from 24.4 to 3.8 seconds (long-many). However, and in line with the previous observations, such reductions do not generally suffice to change the user perceived (*bad*) quality.

Combined upload and download activity. In the case of workloads in both, the uplink and downlink direction (not shown), the QoE is dominated by the upload activity. However, due to lower *overall* link utilization and shorter queueing delays (see § 6), the median PLT are less than for the scenarios involving only uploads. The resulting scores generally map to *bad* quality scores; only the long-few workload shows better QoE for buffers ≤ 128 packets.

9.3 Backbone networks results

The median PLT and the corresponding QoE scores in the backbone setup are shown as a heatmap in Figure 11. As in the access scenario, the heatmap shows buffer sizes on the x-axis and the workload configuration on the y-axis. Each cell is colored according to the MOS scale from Figure 6b and displays the median PLT of 500 web page retrievals.

The baseline results (noBG) show median page loading times of ≈ 0.8 seconds. These loading times are mainly modulated by $14 \times$ RTT (RTT $= 60$ ms (see § 5.1)) needed to fully load the page (RTT component), making them higher than in the access network scenarios that has lower RTTs. In this scenario, the distribution of page loading times generally yields a slightly better performance for buffer sizes greater than or equal to the BDP; for these buffer configurations web pages load up to 200 ms faster (80th percentile not shown in the figure). The short-low scenario yields similar results despite the existence of background traffic.

We observe the first PLT degradations in the short-medium scenario for the 8 and 28 packets buffer configurations. In these cases, PLTs are affected by packet losses causing TCP retransmissions, while the 749 (BDP) and 7490 packet buffers absorb bursts and prevent retransmissions. As in the previous case, web pages load up to 200 ms faster (80th percentile not shown in the figure). The degradations in PLT are, however, small and only marginally affect the QoE score.

Degradations in the short-high scenario are twofold; while packet losses mainly affect the QoE for the 8 and 28 packets buffers, queuing delays degrade the QoE for the larger buffers. This effect is more pronounced in the short-overload and long scenarios that impose a higher link load. In these scenarios, the degradations for the 8 and 28 buffers are mainly caused by packet losses. The 749 and especially the large 7490 buffer affected flow by introducing significant queueing delays; while the RTT doubles for the 749 buffer configuration, it increases by a factor of 10 for the 7490 buffer. Comparing short-overload to long for the 8, 28 and 749 buffer size yields a higher number of retransmissions in the long scenario, degrading the PLT. Concerning the PLT, short buffers of 8 and 28 packets show faster PLT for the short-high, short-overload, and long scenarios. However, improvements in the PLT do not help to generally improve the QoE as the PLTs are already high, causing bad QoE scores.

Our findings highlight the trade-off between packet loss and queueing delays. While larger buffers prevent packet losses and therefore improve the PLT in cases of less utilized queues/links, the introduced queuing delays degrade the performance in scenarios of high buffer/link utilization. In the latter, shorter buffers improve the PLT by avoiding large queueing delays, despite the introduced packet losses. The “right” choice in buffer size therefore depends on the utilization of the link and the buffer.

9.4 Key findings for WebQoE

Our observations fall into two categories: *i)* When the link is low to moderately loaded, larger buffers (*e.g.*, BDP or higher) help minimizing the number of retransmissions that prolong the page transfer time and thus degrade WebQoE. *ii)* When the link utilization is high, however, this increases RTT and thus the page transfers become RTT dominated. Also, loss recovery times increase. Therefore, smaller buffers yield better WebQoE despite a larger number of losses.

However, the impact of the buffer size on the QoE metric page loading time is ultimately marginal, although the QoS metric page loading time sees significant improvements. While this may seem weird at first, let us consider a twofold improvement of the page loading time from 9 seconds to 5 seconds. This improvement is large for the QoS metric, but it is insignificant for the QoE metric, as both 9 and 5 seconds map to “bad” QoE scores regardless the QoS performance.

10. SUMMARY & DISCUSSION

The goal of our work is to elucidate the open problem of proper buffer sizing and to pave the way for more informed sizing decisions. In this respect, this paper presents the first comprehensive study of the impact of buffer sizes on *Quality of Experience*. By this we complement a large body of related work on buffering with a first look at factors relevant to end-user experience. This is a relevant view since it has implications for network operators and service providers, and by extension, device manufacturers.

To tackle this problem, we first evaluate the impact of buffering in the wild using a large data set from a major CDN that serves for a large number of Internet users (80M IPs from 235 countries). Our analysis shows that buffering is likely to be prevalent on a large scale. We also observe a rather modest amount of potential buffer bloat. This motivates our further evaluation of buffer sizing including the impact of very large buffers, *i.e.*, buffer bloat.

The main contribution of this paper is an extensive sensitivity study on the impact of buffer sizing on Quality of Experience. This is based on a testbed-driven approach to study three standard application classes (voice, video, and web) in two realistic testbeds emulating access and backbone networks. Our evaluation considers a wide range of traffic scenarios and buffer size configurations, including buffer bloat.

Our main finding is that the *level of competing network workload* is the primary determinant of user QoE. It is generally known and understood that buffer sizing impacts QoS metrics. In particular, it is not surprising that sustainable congestion degrades network performance. Surprisingly, our results show that in the absence of congestion, buffer sizing has a significant impact on QoS metrics, whereas it only marginally impacts QoE metrics. The good news of this novel observation for network operators is that limiting congestion, *e.g.*, via QoS mechanisms or over-provisioning, may actually yield more immediate improvements in QoE than efforts to reduce buffering. There are, however, several subtle issues that complicate buffer sizing.

Concretely, application characteristics and the level of congestion determine the potential impact of buffer sizing choices. In the case of Web browsing, large buffers yield better QoE for moderate network loads, while smaller buffers improve QoE for high network loads. This suggests load-dependent buffer sizing schemes. Despite the potential for optimization, the impact of reasonable buffer sizes on QoE metrics is marginal, while the impact on QoS metrics can be significant. This is relevant for network operators, as it indicates that as long as buffers are kept to a reasonable size their impact is of marginal relevance. Concerning the ongoing bufferbloat debate, our main claim is that only relatively narrow conditions seriously degrade QoE, *i.e.*, when buffers are over-sized and sustainably filled. Such conditions indeed occur in practice, as our empirical evaluation and other recent studies confirm, but their occurrence is relatively rare.

We remark that emulations are by definition an abstraction of live networks and that predictive QoE models are abstractions of end-users. Thus our results should not be interpreted as representative of any specific network deployment or specific end-user quality ratings. We do, however, argue that our results accurately reflect the key interactions between buffer sizes and network traffic, which is the objective of our study.

We envision future work to extend our first step towards the QoE-driven buffer size evaluation in the following directions: 1) including more applications, 2) going beyond testbed studies by verifying the results of our testbed-driven evaluations in operational (wireless) networks, and 3) verifying selected scenarios in user studies. While our UDP video quality assessment did not involve retransmissions for error recovery, initial work on HTTP video streaming is consistent with our results.

Observed discrepancies among network-centric QoS metrics and application/user centric QoE metrics advocate a stronger use of application centric metrics in measurement and performance evaluation studies. This is challenging since QoE is an application-specific measure and thus needs to be evaluated individually for every application. To reduce the complexity of QoE assessment in network design and network measurement, it appears appealing to aim for a general mapping of network performance (e.g., QoS metrics) to QoE. We believe this is possible for some QoE indicators, e.g., page-loading time in specific scenarios. However, since QoS and QoE represent fundamentally different concepts that depend on different parameters despite of common misconceptions QoE cannot be generally derived from QoS metrics. Examples used in this paper include speech QoE assessment based on audio signals or video quality assessment based on decoded video frames. To simplify future QoE evaluations, this paper exemplifies the use of QoE metrics for measurement studies.

11. ACKNOWLEDGEMENTS

Oliver Hohlfeld was supported in part by DFG SFB 1053 and Telekom Innovation Laboratories project OMAR. We thank Mark Allman and Dennis Guse for their insightful comments and Michael Blodgett for maintaining the WAIL backbone testbed. We also thank the anonymous IMC reviewers and our shepherd Rodrigo Fonseca for their valuable comments and suggestions to improve this manuscript.

12. REFERENCES

[1] Bufferbloat. http://www.bufferbloat.net/.

[2] ITU-T Recommendation P.862: Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-To-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs, 2001.

[3] ITU-T Recommendation G.107: The E-Model, a computational model for use in transmission planning, 2003.

[4] ITU-T Rec. P.862 annex a: Reference implementations and conformance testing for ITU-T Recs P.862, P.862.1 a. P.862.2, 2005.

[5] ITU-T Recommendation G.1030: estimating end-to-end performance in IP networks for data applications, 2005.

[6] Bufferbloat: What's wrong with the internet? *Queue*, 9(12):10:10–10:20, Dec. 2011.

[7] Qualinet white paper on definitions of Quality of Experience. European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Patrick Le Callet, Sebastian Möller and Andrew Perkis, eds., Version 1.1, June 2012.

[8] M. Allman. Comments on bufferbloat. *ACM CCR*, 43(1):31–37, Jan. 2013.

[9] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *ACM SIGCOMM*, 2004.

[10] N. Barakat and T. E. Darcie. Delay characterization of cable access networks. *IEEE Communications Letters*, 11(4):357–359, 2007.

[11] N. Beheshti, Y. Ganjali, M. Ghobadi, N. McKeown, and G. Salmon. Experimental study of router buffer sizing. In *ACM IMC*, 2008.

[12] C. Chirichella and D. Rossi. To the moon and back: are Internet bufferbloat delays really that large. In *IEEE INFOCOM TMA Workshop*, 2013.

[13] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *ACM IMC*, 2007.

[14] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler. Tutorial: Waiting Times in Quality of Experience for Web based Services. In *IEEE QoMEX*, 2012.

[15] S. Egger, P. Reichl, T. Hosfeld, and R. Schatz. "time is bandwidth"? narrowing the gap between subjective time perception and Quality of Experience. In *IEEE ICC*, 2012.

[16] S. Egger, R. Schatz, K. Schoenenberg, A. Raake, and G. Kubin. Same but different? - using speech signal features for comparing conversational voip quality studies. In *IEEE ICC*, 2012.

[17] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden. Routers with very small buffers. In *IEEE INFOCOM*, 2006.

[18] J. Gettys. IW10 considered harmful. IETF Internet-Draft, 2011.

[19] J. Gettys and K. Nichols. Bufferbloat: Dark buffers in the internet. *ACM Queue*, 9:40:40–40:54, Nov. 2011.

[20] Y. Gu, D. F. Towsley, C. V. Hollot, and H. Zhang. Congestion control for small buffer high speed networks. In *IEEE INFOCOM*, 2007.

[21] D. Guse, S. Egger, A. Raake, and S. Möller. Web-QoE under real-world distractions: Two test cases. In *IEEE QoMEX*, 2014.

[22] K. L. Haiqing Jiang, Yaogong Wang and I. Rhee. Tackling bufferbloat in 3G/4G networks. In *ACM IMC*, 2012.

[23] M. Heusse, S. A. Merritt, T. X. Brown, and A. Duda. Two-way TCP connections: old problem, new insight. *ACM CCR*, 41(2):6–15, 2011.

[24] O. Hohlfeld, B. Balarajah, S. Benner, A. Raake, and F. Ciucu. On revealing the ARQ mechanism of MSTV. In *IEEE ICC*, 2011.

[25] V. Jacobson. Modified TCP congestion control algorithm. End2end-interest mailing list, Apr. 1990.

[26] N. Kitawaki and K. Itoh. Pure delay effects on speech quality in telecommunications. *IEEE Journal on Selected Areas in Communications*, 9(4):586–593, 1991.

[27] J. Korhonen and J. You. Peak signal-to-noise radio revised: Is simple beautiful? In *IEEE QoMEX*, 2012.

[28] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating the edge network. In *ACM IMC*, 2010.

[29] A. Lakshmikantha, C. Beck, and R. Srikant. Impact of file arrivals and departures on buffer sizing in core routers. *IEEE/ACM ToN*, 19(2):347–358, Apr. 2011.

[30] J. Martin, J. Westall, T. Shaw, G. White, R. Woundy, J. Finkelstein, and G. Hart. Cable modem buffer management in docsis networks. In *IEEE conference on Sarnoff*, 2010.

[31] S. Möller, W.-Y. Chan, N. Côté, T. H. Falk, A. Raake, and M. Wältermann. Speech quality estimation: Models and trends. *IEEE Signal Process. Mag.*, 28(6):18–28, 2011.

[32] K. Nichols and V. Jacobson. Controlling queue delay. *ACM Queue*, 10(5), May 2012.

[33] R. S. Prasad, C. Dovrolis, and M. Thottan. Router buffer sizing for tcp traffic and the role of the output/input capacity ratio. *IEEE/ACM ToN*, 17(5):1645–1658, Oct. 2009.

[34] A. Raake. Predicting speech quality under random packet loss: Individual impairment and additivity with other network impairments. *Acta Acustia*, 90:1061–1083, 2004.

[35] B. Sat and B. W. Wah. Analyzing voice quality in popular voip applications. *IEEE MultiMedia*, 16(1):46–59, 2009.

[36] L. Sequeira et al. The influence of the buffer size in packet loss for competing multimedia and bursty traffic. In *SPECTS*, 2013.

[37] J. Sommers, P. Barford, A. Greenberg, and W. Willinger. An SLA perspective on the router buffer sizing problem. *SIGMETRICS Perf. Eval. Review*, 35:40–51, March 2008.

[38] J. Sommers, H. Kim, and P. Barford. Harpoon: a flow-level traffic generator for router and network tests. *SIGMETRICS Perf. Eval. Review*, 32(1):392–392, June 2004.

[39] D. Strohmeier, S. Jumisko-Pyykko, and A. Raake. Toward task-dependent evaluation of web-QoE: Free exploration vs. "who ate what?". In *IEEE Globecom Workshops*, 2012.

[40] D. Strohmeier, M. Mikkola, and A. Raake. The importance of task completion times for modeling web-QoE of consecutive web page requests. In *IEEE QoMEX*, 2013.

[41] L. Sun. *Speech Quality Prediction for Voice over Internet Protocol Networks*. PhD thesis, Univ. of Plymouth, 2004.

[42] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Measuring home broadband performance. *Commun. ACM*, 55(11):100–109, Nov. 2012.

[43] Q. H. Thu and M. Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, June 2008.

[44] C. Villamizar and C. Song. High performance tcp in ansnet. *ACM CCR*, 24(5):45–60, Oct. 1994.

[45] A. Vishwanath, V. Sivaraman, and M. Thottan. Perspectives on router buffer sizing: recent results and open problems. *ACM CCR*, 39(2):34–39, Mar. 2009.

[46] M. Wang and Y. Ganjali. The effects of fairness in buffer sizing. In *IFIP-TC6 conference on Ad Hoc and sensor networks, wireless networks, next generation internet*, 2007.

[47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. on Image Processing*, 13:600–612, 2004.

[48] Z. Wang, L. Lu, and A. Bovik. Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication*, 19(1), Jan. 2004.

[49] T. Zinner, O. Abboud, O. Hohlfeld, T. Hossfeld, and P. Tran-Gia. Towards QoE management for scalable video streaming. In *21st ITC-SS*, 2010.