Forschungsberichte der Fakultät IV — Elektrotechnik und Informatik

BufferBloat: How Relevant? A QoE Perspective on Buffer Sizing

Oliver Hohlfeld
Enric Pujol
Florin Ciucu
Anja Feldmann
Paul Barford

Technical Report
Technische Universität Berlin
Telekom Innovation Laboratories

Bericht-Nummer: 2012-11
ISSN: 1436-9915

November 2012

BufferBloat: How Relevant? A QoE Perspective on Buffer Sizing

Oliver Hohlfeld
TU Berlin / T-Labs

Enric Pujol
TU Berlin / T-Labs

Florin Ciucu
TU Berlin / T-Labs

Anja Feldmann
TU Berlin / T-Labs

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. Recently, the debate has focused on the claim that excessive buffering (*i.e., bufferbloat*) can “lead to network-crippling latency spikes” [1] which severely impact Internet services. In this paper, we systematically examine the implications of buffer sizing choices from the perspective of *end users*. To assess user perception of application quality under various buffer sizing schemes we employ Quality of Experience (QoE) metrics contextualized in the familiar framework of systems theory. 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 buffer bloat seriously degrades QoE *i.e.,* when buffers are oversized and sustainably filled.

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 dimensioning remains challenging due to trade-offs in network performance metrics. Large buffers can absorb large traffic bursts and increase TCP throughput, but they can also induce significant delays and thus degrade the performance of network applications; small buffers, in turn, can have inverse effects. Indeed, the implications of ‘preset’ buffer sizes on application performance are largely unknown from technical, operational, economic, and even perceptual perspectives.

Traditionally, router manufactures preset buffer sizes proportionally to the bandwidth of the linecards *i.e.,* bandwidth-delay product (BDP). This rule-of-thumb emerged in the mid 1990s based on a study of the dynamics of TCP flows [21, 39]. A decade later, Appenzeller *et al.* reexamined buffer sizing and argued that much smaller buffer sizes suffice [10]. This reignited interest in the research community with regards to ‘proper’ dimensioning schemes (see Related Work § 2 for a detailed discussion). However, the issue remains far from resolved.

Most recently, the buffer sizing debate has focused on the negative effects of large buffers. The essential argument is that excessive buffering in devices commonly deployed in the Internet today (aka *bufferbloat*) leads to excessive queuing delays (*e.g.,* in the order of seconds), which negatively influences the Internet performance at the users’ home’ [8]. Indeed, bufferbloat can adversely effect TCP by increasing round trip times or even triggering unnecessary TCP timeouts. This can lead to unpredictable TCP performance. Moreover, it can also adversely effect UDP by increasing RTTs or packet losses. While such effects have been observed, the correlation between buffer sizes and the end-user experience has not yet been rigorously demonstrated or quantified.

The objective of our work is to broadly characterize the impact of buffer sizes on end-user experience. Unlike previous studies that address Quality of Service (QoS) metrics (*e.g.,* packet loss events or throughput decay) our study focuses on the (subjective) perception of end-users, namely the *Quality of Experience (QoE)*. We argue that end-user QoE is the metric that is relevant for network operators and service providers, and by extension, device manufacturers.

Quality of Experience is an active research area whose core objective is to quantify the users’ perception of the applications. This is challenging since the users’ perception is subjective. Currently, a solid, theoretical, and practical framework of QoE is still missing [9]. While closing this gap is outside of the scope of this paper, we provide an intuitive illustration of common QoE metrics used for VoIP, video, and Web to assess user perception. For this purpose, we rely on an analogy to the framework of systems theory. In this way, we hope to make the QoE metrics more accessible to the performance evaluation and networking community.

To quantitatively evaluate the impact of buffer sizing on users’ perception we conduct an extensive sensitivity study. Specifically, we evaluate QoE for a wide range of user applications (*e.g.,* web browsing, VoIP, and RTP video streaming) in two realistic laboratory-based testbeds: access and backbone networks. Each application type is analyzed over a range of workloads—without isolation in separate QoS classes—and over a range of buffer sizes.

In the following we summarize the main observations:

1. Exacerbated (bloated) buffers (*e.g.,* ten times rule-of-thumb) have a significant effect on QoE metrics. Reasonable buffer sizes (e.g., $\leq$ BDP) have a significant effect on QoS, as observed by previous studies, e.g., [12], but impact the QoE metrics only marginally.
2. The dominant factor for the QoE metrics in our experiments is the *level of competing network workload*. That is, workloads in which the competing flows keep the queue of the bottleneck link filled (*e.g.,* via many short-lived and therefore not congestion controlled flows) have a much larger impact on QoE than buffer size.
3. In the access network, competing flows in the upload direction, degrade QoE metrics irrespective of the buffer size configuration. This is mainly due to the lower capacity of the uplink. On the one hand, this means that even a small number of competing flows decreases the fair share bandwidth below the necessary bandwidth. On the other hand, buffers drain more slowly thus increasing RTTs. Once the RTT exceeds

the BDP the bidirectional throughput can decrease below the necessary bandwidth.

2. RELATED WORK

The rule-of-thumb [21, 39] 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 [39] and convincingly shown in [10, 12] 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 [10]. Much smaller buffer sizes have been proposed, *e.g.,* drop-tail buffers with $\approx 20 - 50$ packets for core routers [16]. However, these come at the expense of reduced link utilization [12]. For an overview of existing buffer sizing schemes we refer the reader to [40].

While the above discussion focuses on backbone networks, more recent studies focus on access networks, *e.g.,* [13, 24, 27, 37], end-hosts [1], and 3G networks [18]. 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 [8, 17] 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 [29], flow-completion times [25], link utilizations [12], packet loss rates [12], and fairness [41]. Sommers *et al.* studied buffer sizing from an operational perspective by addressing their impact on service level agreements [34]. However, QoS metrics and even SLAs do not necessarily reflect the actual implications for the end-user. Thus, in this paper, we present the first *user-centric* study of the impact of bufferbloat and background traffic by focusing on user perception as captured by QoE metrics.

3. QUALITY OF EXPERIENCE (QOE)

QoE tries to capture 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." [9] To quantify the users' perception of the quality of (network) applications, *QoE Metrics* have been defined for applications such as VoIP, Video, Web, etc. The *value* range of QoE metrics are typically mapped to a 5-point scale—corresponding to absolute categories ranging from 'bad' to 'excellent' [2]. Since QoE depends on the network conditions (*e.g.,* packet loss or delay), the application, and the subjective user experience, QoE metrics are a function of these:

$$\text{QoE} = \mathcal{T}\left(\text{Network, Application, User Perception}\right) , \qquad (1)$$

for *some* mapping $\mathcal{T}$. Here, 'Network' reflects widely used QoS metrics which capture network conditions (*e.g.,* packet loss, delay and jitter), and which implicitly include transport protocols, bandwidth limitations, or buffer sizes. 'Application' captures application level properties such as video encoding artifacts, error concealment strategies, video resolution, and application layer buffering. Lastly, 'User Perception' captures subjective factors including content (*e.g.,* scoring bias for popular videos), user past-experiences, personality traits, gender, age, mood, and expectations (*e.g.,* premium calls should perform better than free calls). Thus, QoE corresponds to a multidimensional perceptual space whereby its features are not necessarily independent of each other. For an extensive discussion on factors influencing QoE, we refer to [9].

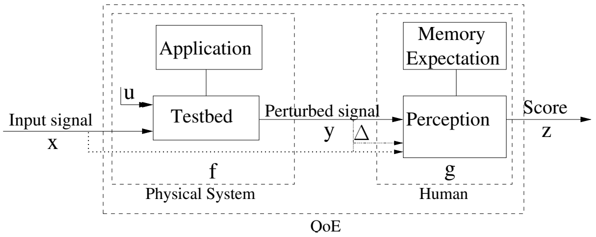


Figure 1: A dual-system view of QoE; u is a perturbation, in this paper defined as workload and buffer size configurations.

The subjective nature of the *user perception* renders it hard to formalize $\mathcal{T}$ in a rigorous manner. As such, a solid, theoretical, and practical framework of QoE is still missing [9]. The scope of our paper is not on closing this gap. Rather, we intend to use an analogy between QoE and system theory to elucidate the concepts of QoE in a familiar framework. The analogy also enables us to point out similarities and differences in key concepts. In this way, we hope to make the QoE metrics more attractive and accessible to the performance evaluation and networking communities, and to pave the way for future contributions in this area.

3.1 QoE: A System Theoretic View

To illustrate $\mathcal{T}$ we use an *analogy* with linear and time invariant (LTI) system theory [26]. In the LTI parlance, a *system* is a mapping $\mathcal{T}$ that transforms *signals*, *i.e.,*

$$\mathcal{T} : \mathcal{S}_1 \to \mathcal{S}_2 \ , \qquad (2)$$

where $\mathcal{S}_1$ and $\mathcal{S}_2$ are sets of signals (*e.g.,* audio streams or number sequences). (Note, we slightly abuse the notation for $\mathcal{T}$.)

The fundamental problem in system theory is to *completely characterize* the system, in the sense that *for any* input signal the corresponding output signal is computable. In the context of LTI, the solution is the so-called *impulse-response* signal, denoted by h, such that for any input signal x, the output signal is computable as

$$\mathcal{T}(u) = x * h \ ,$$

where '$*$' is a *convolution* operator.

In the context of QoE, the definition of an QoE model often involves an analogous concept to the impulse-response signal. Figure 1 illustrates the analogy by introducing a dual-system representation of QoE in terms of Eq. (1). In the first system, the input signal x (*i.e.,* a network application such as an encoded VoIP call) is subject to the perturbations imposed by the testbed and its applications. Here, the testbed is controlled by u, *i.e.,* the workload and buffer size configuration. The output (perturbed) signal y from the first system is then subject to the user perception. This is the responsibility of the second system. The output signal z then corresponds to the perceptive QoE score on the 5-point scale.

One important difference between LTI and QoE is that the systems depicted in Figure 1 are not necessarily time-invariant and may not be linear. In particular, the human component of user perception involves factors such as memory and expectations. Thus, the system has to be considered as time-variant. Moreover, if the performance drops below some threshold, the information carried by y may no longer be useful to the user and therefore the system may have to be considered as non-linear. To capture the time-variant case we include Δ, *e.g.,* to capture that the signal y arrives delayed for example due to delays in phone conversations. To capture non-linearity the score computation has to be adopted.

3.2 QoE Model Construction

According to our above discussion, constructing a QoE model roughly corresponds to identifying the appropriate impulse-responses f and g such that, for any application (input signal) x and perturbation u, the corresponding score z is *computable* according to

$$z \approx ((x \oplus u) *_1 f) *_2 g \;, \qquad (3)$$

where '$\oplus$' is 'some' addition operation, whereas '$*_1$' and '$*_2$' are 'some' convolution operations in 'some' algebra. In particular, g can be seen as the stimulus-response which has been used in modern psychology and is related to QoE in the context of pricing [32].

However, this is where our analogy ends. In practice, QoE models are constructed holistically due to a number of non-linearities caused mainly by the subjective nature of QoE. In particular, g cannot capture the fact that often times users give different scores z_j's for the same perturbed signal y (*e.g.,* depending on mood, expectation, and memory).

To circumvent this lack of linearity, we propose to use *approximate* QoE models. Based on standard testing methodologies, an application x is typically perturbed with n signals u_i's, and the output (perturbed) signals y_i's are further subject to the perception of k people (*i.e.,* the subjective users) (see, *e.g.,* ITU-T recommendations P.800 and P.910 [2, 7]):

$$(x, u_i) \to y_i \to (z_{i,1}, \dots, z_{i,k}) \;\forall i = 1, \dots, n \,.$$

The individual scores z_i's are then summarized in the so called Mean Opinion Score, $MOS_i = \sum_j z_{i,j}/k$. This reduction to the first moment can, however, lead some inaccuracies in the resulting QoE models [20].

Various relationships between u_i's and z's are considered in the QoE literature, including some that are consistent with results in psychology [31]. For instance, the score z often depends logarithmically on x, *e.g.,* if x corresponds to the data rate, on u_i, *e.g.,* if the u_i's correspond to packet loss [31], or on Δ, *e.g.,* if Δ's corresponds to the web page loading times [5]. This captures the Weber-Fechner law[1]. Thus, based on the input (x and u_i's) and output ($z_{i,j}$'s), the construction of an approximate QoE model reduces to fitting Eq. (3).

3.3 QoE Model Application

We next use the analogy between QoE and system theory to introduce two classes of QoE models in this framework, namely *parametric* and *signal-based* models. As an illustration, we describe the metrics we use for our VoIP evaluation (see § 6). Since the metrics depend on the applications, we refer the reader to the corresponding section for a discussion of the metrics for video and Web.

For the VoIP quality evaluation, x corresponds to a voice signal. Note, throughout the paper, u captures both the workload as well as the buffer size configurations (see § 4.2 and § 4.3 for details). To capture the conversational quality of a phone call, the score z is computed as a function of the (measured) end-to-end delay Δ of the voice signal y, *i.e.,* (see [4]):

$$z = 100 - \begin{cases} 0 & \text{if } y \leq 100 \text{ ms} \\ 25\left[\sqrt[6]{1+\alpha^6} - 3\sqrt[6]{1+\left(\frac{\alpha}{3}\right)^6} + 2\right] & \text{otherwise ,} \end{cases}$$

where $\alpha = \frac{\log_{10} \Delta/100}{\log_{10} 2}$. The range of this function is $[0, 100]$, where 100 corresponds to excellent QoE. It is worth noting that the actual function used is not a convolution in the sense of LTI. This function describes the delay impairment factor of the E-Model [4]. We also note that the E-Model includes other impairment factors that are not used in this paper. The E-Model is called a *parametric model* since z only depends on parameters like Δ. It is also referred to as a *no*-reference model, as it does not depend on x.

An example of a *signal-based model* is PESQ [3], which can be used for the same VoIP scenario to account for loss and jitter effects imposed by the 'Testbed'. The PESQ algorithm uses as input both the input voice signal x as well as the output perturbed voice signal y. As such, the PESQ algorithm roughly corresponds to the second part of Eq. (3); see the shortcut for x in Figure 1. The output of the PESQ algorithm is a score z in the range of $[1, 5]$. Note that PESQ does not consider Δ. It is referred to as a *full*-reference, as z depends on both x and y.

4. METHODOLOGY

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

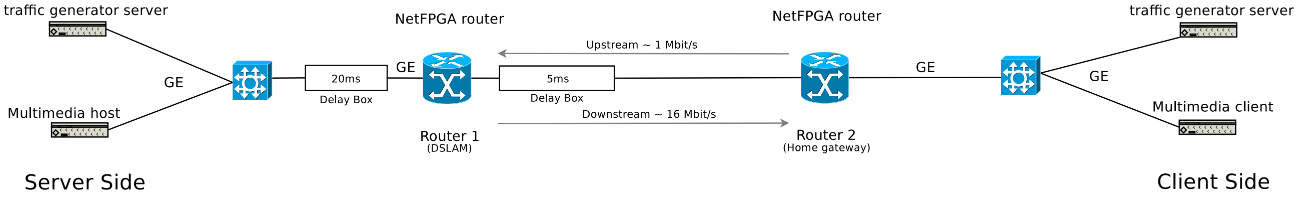
4.1 Testbed Setup

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 2 (a) and (b). We use a testbed setup to have full control over all parameters including buffer sizes and generated workload (the factor u from Figure 1).

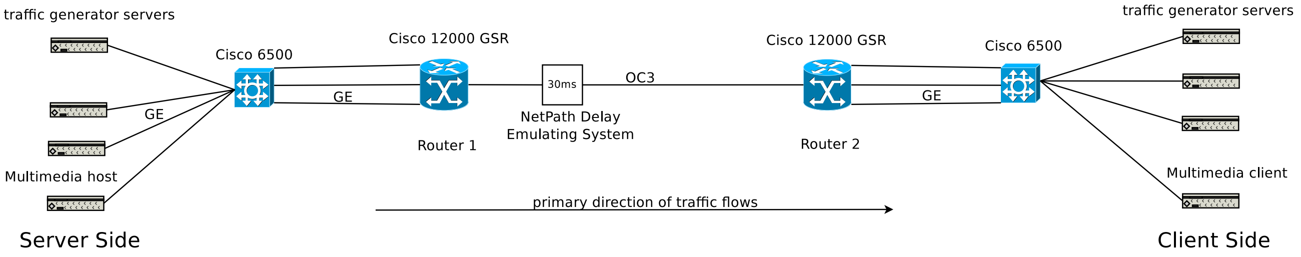
As most flows typically experience only a single bottleneck link, both testbeds are organized as a dumbbell topology with a single bottleneck link, configurable buffer sizes, and a client and a server network. The hosts within the server (client) network on the left (right) side act as servers (clients), respectively. In the backbone 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 [17].

The access network testbed, see Figure 2a, 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 [11]. 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 delay variation is beyond the scope of this paper. This is also the reason

[1] The differential perception dP is proportional to the relative differential $\frac{dS}{S}$ of a stimuli, whence the logarithmic perception-stimuli relationship by integration.



(a) Access Network Testbed



(b) Backbone Network Testbed

Figure 2: Testbeds used in the study

we decided to omit WiFi connectivity which adds its own variable delay characteristics due to layer-2 retransmissions.

To be able to scale up the background traffic to the backbone network, see Figure 2b, 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 delays in the Internet are not constant. However, variable delays are beyond the scope of this paper. Moreover, we eliminate most synchronization potential by our choice of workload.

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).

4.2 Traffic Scenarios

We use the Harpoon flow level network traffic generator [35] 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 twelve scenarios for the access testbed and six for the backbone. We consider more for the access as we distinguish on which links, 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. 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".

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 [35] 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 highlights that using a TCP variant optimized for high latency does not change the overall behavior even if the buffers are large.

4.3 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

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.

Access					Backbone		
Buffer	Uplink		Downlink		Buffer		
Size (Pkts)	Delay (ms)	Scheme	Delay (ms)	Scheme	Size (Pkts)	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 (Assumption: packet size = 1500 bytes.

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.

For the backbone network we use *i)* the same minimum buffer size of 8 packets, which resembles the TinyBuffer scheme [16], 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 [10], 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 × BDP packets an excessive buffering scheme.

5. 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 conduct experiments to measure the latency introduced by the buffers. For this purpose we use the detailed buffer fill statistics of the FPGA cards. Figure 3 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. The values in the heatmaps show the mean delays based on a specific buffer size configuration and a specific workload scenario. Each value is based on one single experiment of two hours length including a multitude of delay measurements. The color of the heatmap cells are chosen according to the ITU-T Recommendation G.114 which captures what kind of delays have the potential to significantly degrade the QoE of interactive applications: green (light gray) is OK, 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 [8] which started the buffer bloat discussion.

Given these high latencies, we investigate the link utilization. Figure 4 shows a boxplot[2] of the link utilization for the specific scenario with simultaneously downloads and uploads (bidirectional workloads) for the various buffer sizes. The left/right half focuses

[2] Boxplots are used to display the location, the spread and the skewness of several data sets in one plot: the box shows the limits of the middle half of the data; the dot inside the box represents the median; whiskers are drawn to the nearest value not beyond a standard span from the quantiles; points beyond (outliers) are drawn individually.

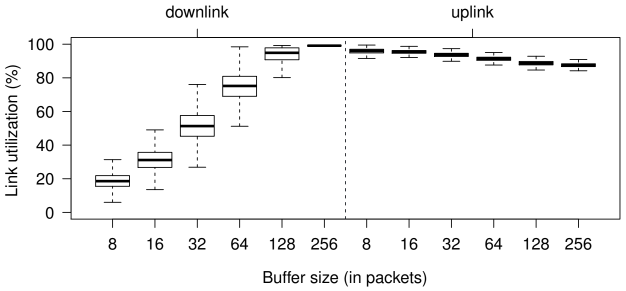


Figure 4: Link utilization in an asymmetric access link with various buffer size configurations. The up- and downlink are simultaneously congested by 8 and 64 long-lived TCP flows, respectively.

on the downlink/uplink utilization. We see that the uplink utilization is almost 100% while the downlink utilization ranges from 20% to 100%. Thus, 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.

The phenomena of low link utilization can be mitigated by counter-intuitively "bloating" the downlink buffer. Considering the delays observed in Figure 3b, 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.

6. VOICE OVER IP

We start our discussion of application QoE with Voice over IP (VoIP). In IP networks speech signals can be impaired by packet loss, jitter, and/or delay. To be more specific, 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. This also degrades speech quality. Moreover, excessive delays impairs any bidirectional conversation as it changes the conversational dynamics in turn taking behavior.

6.1 Approach

We use a set of 20 speech samples recommended by the ITU [6] for speech quality assessment. Each sample is an error-free recording of a male or female Dutch speaker, encoded with G.711.a (PCMA) narrow-band audio codec, and lasts for eight seconds. In our QoE model from Figure 1, the speech samples correspond to the input signal x.

Each of the 20 samples is automatically streamed, using the PjSIP library, over our two evaluation testbeds, see § 4 and subjected to the various workloads. This corresponds to the 'Testbed' from Figure 1. 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) [28] which can also affect the perceived speech quality (see § 3).

For evaluating the QoE of the voice calls we first evaluate the loss and jitter effects, and then the delay effects, using two QoE models, PESQ and E-Model; the resulting scores are then combined to the final QoE score.

Loss and Jitter effects. Each received output audio signal corresponds to the perturbed signal y from Figure 1. To assess the speech quality of y, relative to the error-free sample signal x, we use the Perceptual Speech Quality Measure (PESQ) [3] model for the 'Human' subsystem from Figure 1. PESQ takes as input both the error-free signal x and the perturbed signal y, and computes the score z_1.

Delay effects. 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 [28, 22, 30, 33]. Thus, according to the ITU-T recommendation G.114 one-way delays should be below 150 ms (or at most 400 ms).

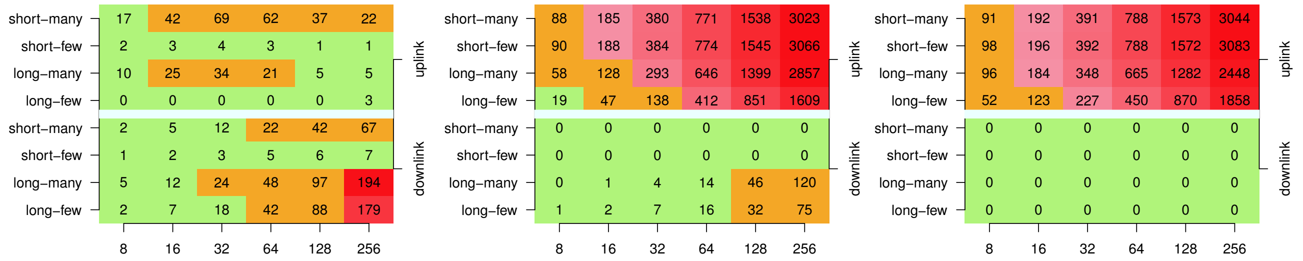
Therefore, we measure the packet delay during the VoIP calls. This delay corresponds to Δ from Figure 1. We now use the delay impairment factor of the ITU-T E-Model [4] to capture the 'Human' subsystem and to get a score z_2. We remark that even though z_2 is computed using a standardized and widely 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 [22, 30, 15]. 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 [22]. 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 [36]. 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-R recommendation P.862.2, see Figure 5a; in the end, low values correspond to bad quality and high values to excellent quality.

6.2 Access networks results

Figures 6a and 6b 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 5a. 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

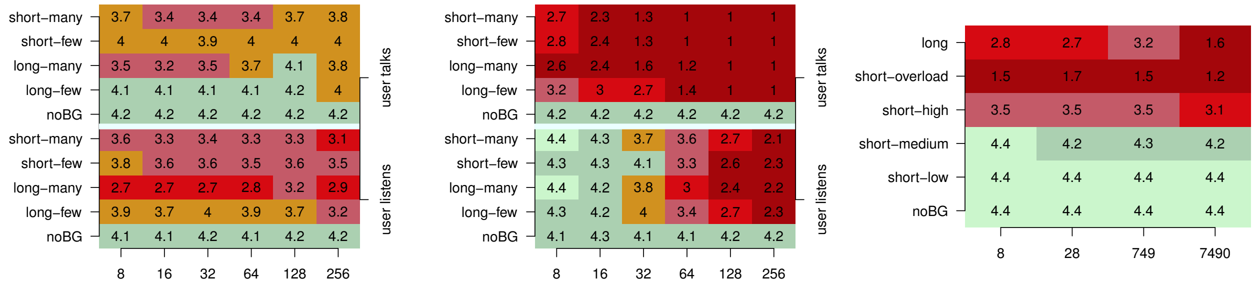


(a) Only downstream workload

(b) Up and downstream workloads

(c) Only upstream workload

Figure 3: 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 Recommendation G.114) are colored in red.

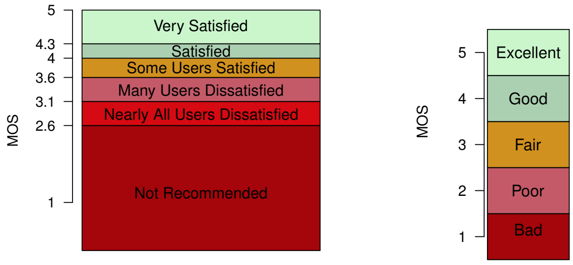


(a) Access: download activity

(b) Access: upload activity

(c) Backbone

Figure 6: 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).



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

(b) Video & Web

Figure 5: MOS scales used in this paper

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 6a 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 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 points from this worst score (e.g., for the 8 packets buffer), suggesting that smaller buffers do not significantly improve audio quality.

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

Upload activity. Figure 6b 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.

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

In the context of the bufferbloat discussion, Figure 6b 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).

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 points (thus mapping to a different rating scale).

For all these reasons, it may be generally a good strategy to isolate VoIP calls in a separate QoS class when the access link is subject to congestion.

6.3 Backbone networks results

Similar to the access network scenario, we show the voice quality in the backbone network scenario as a heatmap in Figure 6c. 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.

6.4 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 highlited 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 voice might be ok, 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 ligh 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.

7. RTP VIDEO STREAMING

Next, we move to video streaming. More specifically, we explore the QoE of streaming using the Real-time Transport Protocol (RTP) which is commonly used by IPTV service providers. RTP streaming can again 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.

7.1 Approach

We chose three different video clips from various genres as input signal x. Each has 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.

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. The sequence of frames received at the multimedia client corresponds to the perturbed signal y.

We note that Set-top-Boxes in IPTV networks often use proprietary retransmission schemes that request lost packets once [19]. 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 QoE.

We use two different full-reference metrics, PSNR and SSIM, for our QoE estimation to compute the scores z_1 and z_2 from the streams x and y. PSNR (Peak Signal Noise Ratio) enables the ranking of the same video content subject to different impairments [38, 23]. However, it does not necessarily correlate well with human-perception in general settings. SSIM (Structural SIMilarity) [42] has been shown to correlate better with human perception [43]. We map PSNR and SSIM scores to QoE MOS scores according to [44].

7.2 Access network results

We show our results as heatmap in Figure 7a. The heatmap shows the QoE 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 perceptive MOS score (see Figure 5b); 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 perceptive 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.

Intuitively, the perceived quality is related to jitter and packet losses, causing artifacts in the video. 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 bottleneck link.

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 ratings. Moreover, much higher loss rates (one order of magnitude bigger) can yield the same ratings. 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 context of the bufferbloat discussion, our results exclude large buffers from being the "criminal mastermind" [1] causing quality degradation, at least for IPTV services. In the case of UDP video streaming in access networks, what matters is the available bandwidth. 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.

7.3 Backbone network results

Similar to the previous access network scenario, we show the video QoE scores obtained for the same video C as a heatmap in Figure 7b, 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 5b). 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.

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

Workloads that sustainably utilize the bottleneck link, i.e., short-high, short-overload, and long, yield bad QoE 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 QoE scores among the three different videos clips leads to minor differences in QoE scores. These differences result from different encoding efficiencies that lead to different levels of burstiness in the streamed video. However, the QoE scores of all video clips lead to the same primary observation: QoE mainly depends on the workload configuration and decreases with link utilization. Increasing the buffer size generally helps to lower the loss rate and therefore to marginally improve the video quality.

7.4 Key findings for RTP video QoE

Our results indicate a roughly binary behavior of QoE: *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 and sometimes ok. This results in an overall degradation that increases with link utilization. Using HD videos yields marginally better QoE scores even though they use higher bandwidth. This is a result of the smaller visual extent of packet losses. We find that the influence of the buffer size is marginal as delay does not play a major role for IPTV. We did not include QoE metrics relevant for interactive TV or video-calls. Thus, what mainly matters for RTP video streaming is the available bandwidth.

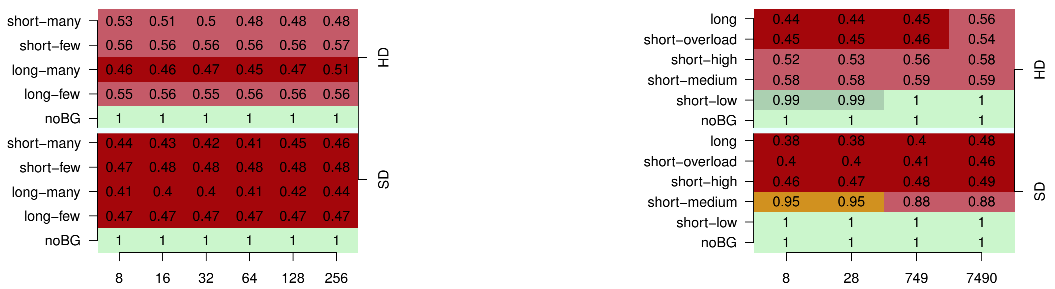
8. WEB BROWSING

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, 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. Consequently, factoring in various workloads and buffer sizing configurations—which influence the TCP performance—is particularly relevant for understanding web browsing QoE.

8.1 Approach

In our QoE model from Figure 1, the PLT corresponds to Δ. To evaluate the WebQoE, we map the PLT Δ to a user score z by using the ITU recommendation G.1030 [5]. We consider the one-page version of the ITU model, which logarithmically maps *single*



(a) Access: download activity

(b) Backbone

Figure 7: Median MOS (color) and SSIM (text) for HD and SD RTP video streams with different buffer size (x-axis) and workloads (y-axis).

PLT's Δ 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 5b). 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.

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.

8.2 Access network results

Figures 8a and 8b 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 5b.

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.

Download activity. Figure 8a 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 end-users perceived quality by limiting the loss rate.

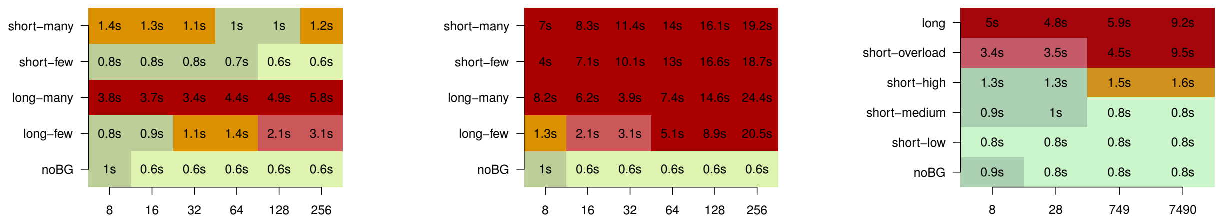
Bufferbloat is visible for the long-few scenario, where the median PLT increases with the buffer size, as the PLT is dominated by RTTs caused by large queueing delays. As for the previous scenario, the effects of various buffer sizes are clearly perceived by the end-user (yet in a different manner).

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 thus the PLT increases beyond the users' threshold of acceptance. 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 8b 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 minimum 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 buffer size. 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 § 5), 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 $\leq$ 128 packets.



(a) Access: only download activity

(b) Access: only upload activity

(c) Backbone

Figure 8: Median MOS (color) and page loading times (text) with different buffer size (x-axis) and workload (y-axis) configurations.

8.3 Backbone networks results

The median PLT and the corresponding QoE scores in the backbone setup are shown as a heatmap in Figure 8c. 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 5b 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 § 4.1)) 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. With respect to 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 and lead to bad quality 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.

8.4 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. Moreover, 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" performance regardless of the users' expectations.

9. SUMMARY

Over the past several years, the negative consequences of increasingly large buffers (*i.e., buffer bloat*) deployed in network systems has become a popular topic. In this paper, we present the first sensitivity study on the impact of buffer sizing on user experience quantified by QoE metrics. We used a testbed-driven approach to study three standard application classes (voice, video, and web) in an access network testbed—with NetFPGA hardware—and a backbone testbed—with carrier grade routers. We consider various scenarios over a range of buffer sizes and congestion conditions imposed by realistic background traffic.

We document the extent to which excessive buffering can lead to degradations in VoIP and Web QoE. However, surprisingly, we find that application QoE degradations due to the level of congestion is often more significant than the those due to excessively large buffers. This leads us to conclude that limiting the congestion, *e.g.,* via QoS mechanisms, may actually yield more immediate improvements in QoE than efforts to reduce buffering. There are, however, some subtle dependencies. For example, excessive buffering in one direction, *e.g.,* upstream, can render the downstream buffer undersized in terms of their ability to effectively absorb transient traffic spikes.

The next step in our work is to consider more diverse traffic types such as HTTP video streaming (*e.g.,* YouTube). Preliminary results with these traffic types are consistent with our above observations.

10. REFERENCES

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

[2] ITU-T recommendation P.800: Methods for objective and subjective assessment of quality, 1996.

[3] 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.

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

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

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

[7] ITU-T recommendation P.910: Subjective video quality assessment methods for multimedia applications, 2008.

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

[9] 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., Lausanne, Switzerland, Version 1.1, June 2012.

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

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

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

[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, 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.

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

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

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

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

[20] T. Hoßfeld, R. Schatz, and S. Egger. SOS: The MOS is not enough! In *IEEE QoMEX*, 2011.

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

[22] 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.

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

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

[25] 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.

[26] E. A. Lee and P. Varaiya. *Structure and Interpretation of Signals and Systems*. Lee and Varaiya, 2.01 edition, 2011.

[27] 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.

[28] 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.

[29] 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.

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

[31] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo. The logarithmic nature of QoE and the role of the weber-fechner law in qoe assessment. In *IEEE ICC*, 2010.

[32] P. Reichl, P. Kurtansky, J. Fabini, and B. Stiller. A stimulus-response mechanism for charging enhanced quality-of-user experience in next generation all-IP networks. In *Latin Ibero-American Operations Research Conference*, 2006.

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

[34] 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.

[35] 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.

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

[37] 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.

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

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

[40] 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.

[41] 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.

[42] 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.

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

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