# Sizing Network Buffers: A HTTP Adaptive Streaming Perspective

Darijo Raca*, Ahmed H. Zahran*†, Cormac J. Sreenan*,
*Computer Science Dept., University College Cork, Ireland
†Electronics and Electrical Communications Engineering Dept., Cairo University, Giza, Egypt
E-mail: {d.raca, a.zahran, cjs}@cs.ucc.ie

***Abstract*—HTTP Adaptive video Streaming (HAS) is the dominant traffic type on the Internet. When multiple video clients share a bottleneck link many problems arise, notably bandwidth underutilisation, unfairness and instability. Key findings from previous papers show that the "ON-OFF" behaviour of adaptive video clients is the main culprit. In this paper we focus on the network, and specifically the effects of network queue size when multiple video clients share network resources. We conducted experiments using the Mininet virtual network environment streaming real video content to open-source GPAC video clients. We explored how different network buffer sizes, ranging from 1xBDP to 30xBDP (bandwidth-delay-product), affect clients sharing a bottleneck link. Within GPAC, we implemented the published state-of-the-art adaptive video algorithms FESTIVE and BBA-2. We also evaluated impact of web cross-traffic.**

**Our main findings indicate that the "rule-of-thumb" 1xBDP for network buffer sizing causes bandwidth underutilisation, limiting available bandwidth to 70% for all video clients across different round-trip-times (RTT). Interaction between web and HAS clients depends on multiple factors, including adaptation algorithm, bitrate distribution and offered web traffic load. Additionally, operating in an environment with heterogeneous RTTs causes unfairness among competing HAS clients.**

**Based on our experimental results, we propose 2xBDP as a default network queue size in environments when multiple users share network resources with homogeneous RTTs. With heterogeneous RTTs, a BDP value based on the average RTTs for all clients improves fairness among competing clients by 60%.**

## I. Introduction

Recent years have witnessed a tremendous rise in multimedia communications and content sharing over the Internet. Specifically, HTTP Adaptive Streaming (HAS) has been adopted by the major video content providers because HTTP is widely used, and is supported by Content Delivery Networks (CDNs). This raises several concerns regarding the impact of sharing network bottlenecks among HAS [1] and non-HAS traffic [2]. When multiple video clients share a bottleneck link, performance issues such as unfairness and quality instability arise [1]. Additionally, when video traffic is mixed with interactive traffic (e.g., VoIP or web), Mansy et al. [3] show that a HAS client may saturate the network queue, causing the bufferbloat effect for the VoIP client. Motivated by the issues raised in previous studies, we seek an in-depth understanding of bottleneck queue size selection on the performance of Internet traffic under different operating conditions.

*Modified with Errata Changes

In HAS systems, each video clip is split into a series of fixed-duration segments, the duration being typically 2-10 seconds. Each segment is encoded into different representations that have different data rates. The HAS client implements an adaptation algorithm which takes into account the current operating conditions in order to determine the appropriate segment quality to select. The performance of HAS systems is affected by many parameters including client-, video-, and system-specific parameters. Client-specific parameters include application buffer size and the design of adaptation algorithm, video-specific parameters are mainly related to video encoding parameters such as encoding rates, while system-specific parameters include network delay, capacity, number of competing streams, nature of competing traffic, and traffic management policy.

Most of the proposals in the literature focus on the design of video adaptation at the client [4]–[7] and overlook network configuration. However, several recent studies [8], [9] indicated that advanced traffic management techniques, such as rate shaping, could improve the streaming performance when multiple clients share a bottleneck link. However, such techniques require changes to the configuration and operation of routers, while today most routers still implement simple forwarding techiques and FIFO queues. The size of network queues plays an essential role in absorbing short packet bursts and smoothing outgoing packet departures, but determining the right buffer size can be challenging. In this study, we extensively investigate the impact of dimensioning the network FIFO queue size at the bottleneck link on the performance of HAS and web browsing applications. To the best of our knowledge, this paper is the first study that extensively investigates the impact of network queue size on multiple HAS and non-HAS clients sharing network resources.

In this work, we conduct our performance evaluation in an experimental testbed that uses real video content and Internet traffic. Our video clients use two state-of -the-art streaming adaptation algorithms, namely: FESTIVE [5], and BBA-2 [7], which we implemented in the open-source GPAC video player. These algorithms represent two popular types of adaptation philosophies, namely rate-based and buffer-based. Our web clients browse the most visited 250 web-pages across different user interests. Across a range of different settings we investigated both video-only operation as well as mixed video and web traffic.

Our experimental evaluations reveal many interesting observations. When multiple video clients compete for the bottleneck resource, we conclude:

- increasing the network queue size from the recommended 1xBDP [10] to 2xBDP improves both the average video quality that is delivered and the system resource utilisation;

- when video clients have heterogeneous RTT delays, employing larger network queue sizes assists the system to improve its fairness. Note that this is a critical issue as some content providers site their CDN distribution nodes close to, or within, the network of the Internet service provider.

When multiple web and video users share a bottleneck link we conclude:

- increasing the bottleneck queue size helps web clients that compete with HAS clients by increasing their throughput and decreasing the number of excessively delayed web pages, but would have a scenario-dependent impact on the web page load time.

In Section II, we present background and related work. Our experimental methodology and platform is provided in Section III, followed by performance results in Section IV and V. Finally, Section VI concludes the paper.

## II. Background and Related Work

The majority of video streaming services, including Youtube, Netflix, and Hulu, use HTTP adaptive video streaming and thus it is common that multiple HAS clients share a bottleneck link, along with other traffic. Several recent studies, e.g. [1], [9], [11] blamed the client behaviour for frequent quality switches and unfairness issues when multiple video clients share a bottleneck link. The "ON-OFF" behaviour of HTTP adaptive streaming clients in steady state is recognized as the main source of the aforementioned issues [1]. In this context, ON means that client is downloading a segment while OFF corresponds to a client waiting to have sufficient space in its buffer to request the next segment. In [11], TCP *cwnd* issues are identified as an additional factor, leading to the behaviour recognized as the "downward spiral effect", which can be mitigated through the use of larger segments allowing TCP to have a better estimate of available bandwidth.

With the increasing popularity of video, its impact on other traffic sharing a bottleneck becomes a concern. Video traffic stands accused of triggering bufferbloat phenomena occurring in the Internet [2]. This phenomenon occurs when large network buffers get full with video packets causing unnecessarily high latency. This negatively affects the TCP congestion avoidance algorithm, creating a standing queue, when the number of outstanding packets in the network are larger than $BDP$ [12]. The number of surplus packets causes the delay, which has a negative impact on interactive applications such as web browsing, VoIP and video conferencing. Mansy et al. [3] investigated the performance when one video client shares a bottleneck link with one voice client and showed that HTTP adaptive steaming application can cause bufferbloat and harm other delay-sensitive applications sharing the same bottleneck.

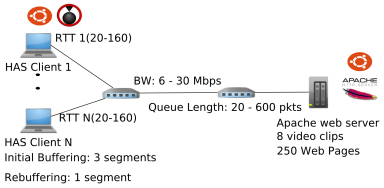



Fig. 1. Testbed Topology

TABLE I
Different Parameter values for experiments

| Parameter | Default | Range |
|---|---|---|
| N | 6 | 2-8 |
| RTT (ms) | 40 | 20-160 |
| C (Mbps) | 6 | 6-30 |
| Client Buffer (s) | 60,240 | 60-640 |
| Network Queue | 1xBDP | 1-30xBDP |

Adaptive Queue management algorithms [13], notably Random Early Detection (RED) and Controlled Delay (CoDel) [12], have often been recognised as the solution for bounding latency at the network queue. Although available in core routers, complex and tedious configuration present a formidable stumbling block, often leading to misconfiguration. Instead, the popular FIFO drop-tail mechanism is still widely used due to its inherent simplicity.

There are two well-recognized approaches for dimensioning network FIFO queues, the rule-of-thumb [10] and the Stanford rule [14]. The rule-of-thumb approach specifies that the bottleneck queue size should equal the $BDP$ (RTTxC, where RTT is round-time-delay and $C$ is the bottleneck capacity). This rule is often applied at the edge part of the network, where the number of flows and bandwidth capacity is relatively small. The Stanford rule is recommended for a large number of TCP flows (over 250) and very high speed links. Then the recommended router queue size is $BDP/\sqrt{F}$, where $F$ is the number of TCP flows sharing the bottleneck link [14]. However, prior work has not investigated if these recommended sizes are appropriate when video traffic dominates.

## III. Methodology

In this section, we present our experimental methodology, including details of our platform and key performance metrics. Regarding notation, we use *queue* to refer to the bottleneck link buffer and *buffer* to refer to the application buffer on the client side.

### A. Experimental Platform

In our experiments we use a common dumbbell topology, depicted in Fig. 1. TABLE I shows default values and ranges for key parameters used in the experiments. We consider $N$ clients sharing a bottleneck link and requesting their data from a remote HTTP server. This topology is commonly used in scenarios where multiple clients share network resources [1], [3]–[7], [9]. Our platform is created using the popular open-source Mininet system [15], which allows emulating large networks in real-time. All the clients, server and switches are modelled as Linux nodes within the Mininet environment.

We evaluate the streaming performance while varying the bottleneck link queue size, bottleneck link bandwidth, and the client-server RTT. We configure the bottleneck link using netem/tc [16]. In our experiments, we vary the bandwidth between $6-30Mbps$ and the RTT delay between $20-160ms$. These values are typical for broadband connections in Europe [17]. Our bottleneck link is configured with a single FIFO queue.

Our HAS client is based on a well-known open-source player, GPAC[1]. In addition to its default adaptation algorithm, we implemented FESTIVE [5] and BBA-2 [7], both well-cited examples of state-of-the-art algorithms from the literature. GPAC's simple default algorithm requests the representation whose rate is just below the throughput of the last downloaded segment. The BBA-2 algorithm [7] mainly selects the next segment representation by mapping the buffer level to a target segment size. It also incorporates a throughput-based decision in its "startup" phase while taking into account future video segment sizes. The FESTIVE algorithm [5] is a rate-based algorithm that uses a harmonic estimator for the network throughput. It employs three components, namely, randomized chunk scheduling (avoiding ON/OFF issues), stateful bitrate selection (cautious up-switching) and a delayed update approach (trade-off between switching and stability). For all algorithms, we consider 12 seconds of initial buffering and 4 seconds of rebuffering, when stalls occur. We consider different application buffer sizes including $60-640$ seconds. Typical buffer sizes observed in the literature vary from 30 sec [5] to 240 sec [7]. While 60s buffer is a typical buffer size, the 640s buffer is selected purely to allow us to observe the impact of eliminating "ON-OFF" behaviour on the streaming performance.

We use eight clips randomly selected from a publicly available HAS dataset [18]. The dataset includes video resolutions up to 1980 x1280 with ten quality representation rates (kbps): 235, 380, 568, 760, 1065, 1777, 2387, 3046, 3906, 4361, as used by popular streaming services [11]. The segment duration for all clips is 4 seconds, which again is commonly used by popular services. All the experiments last for the duration of clips, which are 16 minute long, served from an APACHE server.

Our web clients are based on Firefox that is driven by Selenium, a web browser automation tool[2]. A User typically opens a page, spends some time on it (*dwell time*) before proceeding to the next page [19]. The dwell times depend on the type of the content and attention span of the user [19]. Dwell times are modelled with a Weibull distribution, whose scale parameter $\lambda$ depends on the type of content, as justified and detailed in [20]. In over 80% of their data, dwell times are in the range between 2 and 70 seconds. Our web pages represent the top 250 most visited pages in the following categories [20]: Science, Travel, Recreation, Computers, Entertainment, Finance, Relationships, Education, Society, and Vehicles. These categories were taken from [20], for which the authors have provided parameters in order to generate appropriate dwell times. The HTTTrack tool[3] is used for downloading the entire webpages and storing it on our own web server. The median size for all webpages is 4058 kB, while 1st and 3rd Quartile are 939.5 and 13007 kB, respectively. Our web user randomly selects a page, persists on it for some random time which depends on the content [20], and then proceeds to request the following page. If a page loading time is greater than 15 seconds the user abandons the page, selecting the next one from the list. Generated dwell times are bounded between 2 and 70 seconds.

### B. Key Performance Metrics

Our main HAS performance metrics include player instability ($\theta$) [1], [5], unfairness ($\phi$) [5], bandwidth utilisation ($\vartheta$), average quality representation rate ($\chi$) and stall performance ($\psi$).

The player instability metric captures the frequency of performing quality switches and is evaluated as [1], [5]:

$$\frac{\sum_{d=0}^{k-1} \mid b_{x,t-d} - b_{x,t-d-1} \mid \cdot w(d)}{\sum_{d=1}^{k} b_{x,t-d} \cdot w(d)}, \tag{1}$$

where $k$ is the last 20 seconds, $b_{x,t}$ is the bitrate for client $x$ at time $t$ and $w(d) = k - d$ is a weight function for adding a linear penalty for recent switches. Clients performing more quality switches have higher instability values.

The unfairness metric captures the interaction between streaming clients that are sharing the link and is estimated as $\sqrt{1-J}$, where $J$ is Jain fairness index [5]. Smaller values indicate that clients fairly share the link.

The bandwidth utilisation metric captures the total amount of bandwidth that all video clients are using. Even if they have equal shares of available bandwidth, there is a chance that they do not utilise the bandwidth efficiently. The average bandwidth utilisation represents the percentage of bandwidth used over the session lifetime and is estimated as the ratio of the transmitted data to the maximum amount of data that could have been transmitted during the session activity.

Stall performance is one of the most important factors that can influence the Quality of Experience [21]. A stall is experienced when the client is forced to pause the video because it has insufficient data in its buffer. We capture stall performance using a three-element tuple $(s_1, s_2, s_3)$, where $s_1, s_2, s_3$ respectively represent the percentage of clients experiencing at least one stall, total numbers of stalls and total stall duration.

For the web clients, we capture two key performance metrics: page-loading time (PLT) and the fraction of abandoned pages.

The results shown represent the average of five runs, with 95% confidence intervals. In every run, each client randomly requests a video clip from the server and starts at most 4 seconds (randomly selected) after the previous client.

[1]https://gpac.wp.mines-telecom.fr/

[2]http://www.seleniumhq.org/

[3]https://www.httrack.com/

## IV. Video Traffic Experiments

In this section, we investigate the performance when only video clients share the bottleneck link queue. We vary different parameters including the queue size, the link capacity, the RTT, and number of clients.

### A. Impact of Queue Size

We consider six video clients competing for a $6Mbps$ link with a $40ms$ RTT. We vary the size of the network queue from 1xBDP to 30xBDP. The 30xBDP is selected purely to allow us to observe the case where there would be no packet loss.

*1) Bandwidth Utilisation:* Fig. 2a shows bandwidth utilisation versus the queue size for GPAC default, BBA-2 and FESTIVE with 60sec and 640sec client buffers. With the 1xBDP queue size (rule-of-thumb [10]) the link utilization is 70% on average. The impact of removing "OFF" periods (640s application buffer) has a marginal positive effect on bandwidth utilisation. For queue sizes ranging from 2xBDP to 20xBDP, the bandwidth hovers around 85%. In the extreme case of 30xBDP, clients achieve the highest bandwidth utilisation of around 92%.

*2) Instability:* Fig. 2b plots the instability metric versus the queue size for different algorithms and buffer sizes. Increasing the queue size has an insignificant impact on the instability of FESTIVE except for the extra large queue size (30xBDP). This implies FESTIVE would be more stable when the network does not drop packets. We also noticed that the client buffer duration marginally affects FESTIVE's switching behaviour. The queue size has a slight impact on the switching behaviour of BBA-2. On the contrary, a larger application buffer significantly improves BBA-2 switching behaviour. We believe that with larger buffer sizes, BBA-2 has a larger cushion region and hence performs fewer switches. When the application buffer is small, then the cushion region for each nominal rate is also small, forcing the algorithm to switch too often. To illustrate, in the case of the 60 second buffer, clients are making on average 200 switches per clip. The clients request a new rate approximately every couple of segments, making them very unstable. Hence, it may be unfair to draw conclusions on BBA-2 performance with small or medium buffer sizes.

*3) Fairness:* Fig. 2c plots the unfairness metric versus queue size for different algorithms and buffer sizes. We observe that all tested algorithms achieve the smallest unfairness metric with extra large queue size. At the rule-of-thumb size, the streaming algorithms show a higher level of unfairness in comparison to slightly larger queue sizes (2-6xBDP). Generally, FESTIVE is most fair. BBA-2 shows a high sensitivity to the buffer size. For the large client buffer BBA-2 shows a reduction in unfairness value by $50\%$.

*4) Average Representation Rate:* Fig. 2d shows the average representation rate across different queue and buffer lengths. All three algorithms request lower representation rates on average for the 1xBDP case. In the middle region (2-20xBDP), algorithms show improvement, requesting on average $900kbps$. When the queue is large enough to eliminate the packet-loss component at the bottleneck link, algorithms request higher representation rates (averaging $1Mbps$). The client buffer impacts the average representation rate significantly, resulting in higher rates when the client buffer duration is 60 seconds and vice versa. This holds for all the algorithms. We believe these results have roots in the algorithm design. When using a large application buffer, clients access the network all the time, causing it to estimate lower throughput, while in the case of a smaller buffer, clients request segments at different times (randomized scheduling component of the FESTIVE) which leads to the estimation of a larger throughput values. BBA-2 with a large buffer has slower up-switches because of the larger cushion regions, which forces a client to download more segments with lower quality before it can switch to higher representation rates.

TABLE II
Stall Performance across different queue lengths and client buffer durations

| Algorithm | Low | Middle | High |
|---|---|---|---|
| **Default 60s** | (3,1,1) | (5,31,38) | (100,645,1348) |
| **FESTIVE 60s** | (0,0,0) | (0,0,0) | (13,20,28) |
| **BBA-2 60s** | (3,1,46) | (0,0,0) | (0,0,0) |
| **Default 640s** | (3,7,44) | (4,41,39) | (100,642,1239) |
| **FESTIVE 640s** | (0,0,0) | (0,0,0) | (0,0,0) |
| **BBA-2 640s** | (0,0,0) | (0,0,0) | (0,0,0) |

*5) Stall Performance:* Table II shows stall performance for all three algorithms with two values for the client buffer duration. Queue lengths between 2-20xBDP (**Middle**) are merged and results shown represent the sum across all values for each particular queue length. Configurations 1xBDP and 30xBDP are shown as **Low** and **High**, respectively. When the queue length is big enough to accommodate all the outstanding packets in the network, queueing delay negatively impacts the stalls if the client algorithm is rate-based as in FESTIVE and GPAC default. When the clients use a large buffer, FESTIVE manages to stream all clips without a single stall. In the case of BBA-2, small buffer and queue length can lead to one client experiencing a relatively long stall.

Table III summarizes the key performance metrics for FESTIVE and BBA-2 for different queue sizes for RTT $40ms$. In this table, a small queue size corresponds to the 1xBDP case, while the large queue size corresponds to the no-loss network (30xBDP), and medium metrics are averaged for queue sizes 2-20xBDP. Each cell is coloured with one of the colours green, yellow, and red, representing good, average and bad relative performance, respectively.

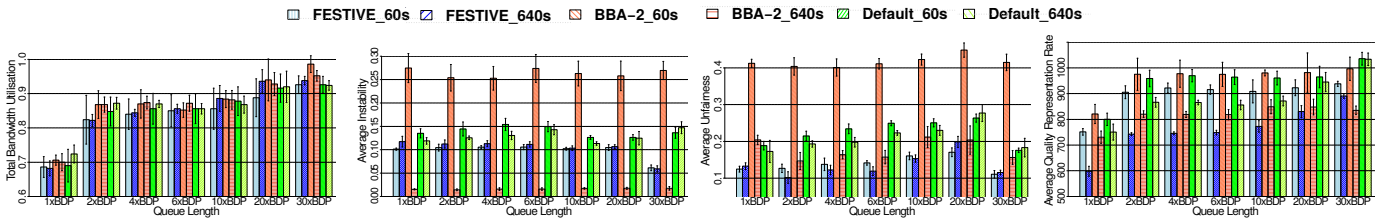
Our key observations can be summarized as follows:

**Observation 1**: the queue size has limited impact on instability and unfairness performance metrics for medium and large queue lengths;

**Observation 2**: the rule-of-thumb queue size leads to a relatively lower bandwidth utilisation and lower average streaming rates, and has no performance merit for other metrics;

**Observation 3**: the best performance is attained when both network queue and application buffer are large in size;

**Observation 4**: the performance degradation of low RTT (Section IV-D) can be avoided by using a queue size larger than 1xBDP.




(a) total bandwidth utilisation (b) avg. instability (c) avg. unfairness (d) average quality rep. rate

Fig. 2. HAS performance metrics vs. network queue size

| | | Low | Mid | High | Low | Mid | High |
|---|---|---|---|---|---|---|---|
| 640s Buffer | $\vartheta$ | 70% | 88% | 95% | 68% | 86% | 94% |
| | $\theta$ | 0.02 | 0.02 | 0.02 | 0.12 | 0.11 | 0.06 |
| | $\phi$ | 0.20 | 0.17 | 0.16 | 0.13 | 0.13 | 0.12 |
| | $\chi$ | 731 | 829 | 835 | 598 | 761 | 890 |
| | $\psi$ | (0,0,0) | (0,0,0) | (0,0,0) | (0,0,0) | (0,0,0) | (0,0,0) |
| 60s Buffer | $\vartheta$ | 70% | 88% | 98% | 68% | 85% | 92% |
| | $\theta$ | 0.27 | 0.26 | 0.27 | 0.10 | 0.10 | 0.06 |
| | $\phi$ | 0.41 | 0.42 | 0.42 | 0.12 | 0.14 | 0.11 |
| | $\chi$ | 821 | 978 | 997 | 751 | 912 | 938 |
| | $\psi$ | (3,1,46) | (0,0,0) | (0,0,0) | (0,0,0) | (0,0,0) | (13,20,28) |

(a) BBA-2 (b) FESTIVE

TABLE III
SUMMARY RESULTS (BW. UTILISATION - $\vartheta$ AVG. INSTABILITY - $\theta$ AVG. UNFAIRNESS - $\phi$ AVG. RATE - $\chi$ STALL PERF. - $\psi$) ACROSS DIFFERENT QUEUE SIZE REGIONS (LOW: 1XBDP, MID: (2-20XBDP), HIGH: 30XBDP) AND 40MS RTT

Based on these results, the rule-of-thumb 1xBDP queue size does not seem to be a good choice for HAS clients sharing a bottleneck. Additionally, the increase in average rate and bandwidth utilisation when using extra large queue sizes is not significant in comparison to medium queue sizes. Hence, we argue that a medium queue size strikes a sweet operating point for video clients with homogeneous RTT times.

FESTIVE and BBA-2, represent two philosophies for rate picking approaches: rate-based and buffer-based. Although the scope of the paper is not driven by any conclusion regarding what type of the algorithm is better, we found that using a smaller buffer for BBA-2 leads to stability and fairness problems. This is due to shrinking the cushion region in the buffer. Hence, we expect that using a default 240sec [7] would be ideal for buffer-based strategies. That said, using a large buffer leads to inefficient use of resources if the user prematurely abandons the session.

In the rest of the paper, we drop the evaluation of the GPAC default algorithm because most of HAS players estimate an average value for throughput samples rather than using raw throughput values for driving their decisions. In addition, we configure BBA-2 with a 240sec application buffer.

### B. Impact of Link Capacity

In this section we explore how the performance of HAS clients would vary for different bottleneck link capacity configurations. We vary the bottleneck link capacity ($6-30Mbps$) and investigate the impact on key performance metrics. Due to space constraints, we provide a summary of key conclusions.

Firstly, using different bottleneck link capacities confirms our previous findings. For all link capacity values and algorithm type combinations, clients underutilise resources when the network queue length equals 1xBDP, using only on average $70\%$ of the link capacity.

The unfairness metric value does not change significantly (less than $10\%$ on average) across different network queue lengths for any particular link capacity value. For our recommended queue length of 2xBDP, the unfairness metric drops by 80% and 60%, for FESTIVE and BBA-2 respectively, when capacity increases from 6 to $30Mbps$. This change is intuitive as all clients stream with the highest rate. Additionally, instability drops by 90% and 50%, as the capacity changes from 6Mbps to 30Mbps for FESTIVE and BBA-2, respectively. This happens because the client performs fewer switches as the gap between subsequent quality-rates improves, for the rates higher than $1Mbps$.

Our results also show that increasing the network queue size above 1xBDP results in quality improvement, which is more pronounced as the capacity increases. For the $30Mbps$ case, quality improves 40% and 45% for BBA-2 and FESTIVE, respectively.

### C. Impact of Changing the Number of Clients

In this section, we investigate the impact of changing the queue size and number of clients on the performance of HAS. We run experiments with default settings, as shown in TABLE I, only changing the number of clients from 2 to 8 for the network queue lengths equal to 1 and 2xBDP. Let $V_i$ represent a scenario in which $i$ HAS clients share the bottleneck link. TABLE IVa and TABLE IVb show four key performance metrics for FESTIVE and BBA-2, respectively, for different $V_i$ scenarios with network queue sizes of 1x and 2xBDP. In all scenarios, using a larger queue size enables increasing network utilisation and achievable quality rate. For FESTIVE, the average quality rate increases as the number of clients increases. Similarly, bandwidth utilisation increases by 17% and 12% for 1xBDP and 2xBDP, respectively, as the number of clients increases from 2 to 8. This confirms results from [4], where authors also show that FESTIVE utilisation increases with the number of clients. The instability metric doubles for both cases. Unfairness values increase two and three times for 1xBDP and 2xBDP case, respectively. These results are also consistent with results from [6] regarding unfairness and instability trends, where bandwidth overesti-

TABLE IV
KEY PERFROMANCE METRICS ACROSS DIFFERENT NUMBER OF CLIENTS

| | 1xBDP | | | | 2xBDP | | | |
|---|---|---|---|---|---|---|---|---|
| # | $V_2$ | $V_4$ | $V_6$ | $V_8$ | $V_2$ | $V_4$ | $V_6$ | $V_8$ |
| $\theta$ | 0.06 | 0.09 | 0.1 | 0.12 | 0.053 | 0.13 | 0.1 | 0.11 |
| $\phi$ | 0.07 | 0.08 | 0.12 | 0.14 | 0.05 | 0.14 | 0.12 | 0.14 |
| $\vartheta$ | 57 | 63 | 69 | 74 | 72 | 75 | 82 | 84 |
| $\chi$ | 1802 | 1003 | 751 | 625 | 2246 | 1206 | 905 | 709 |

(a) FESTIVE

| | 1xBDP | | | | 2xBDP | | | |
|---|---|---|---|---|---|---|---|---|
| # | $V_2$ | $V_4$ | $V_6$ | $V_8$ | $V_2$ | $V_4$ | $V_6$ | $V_8$ |
| $\theta$ | 0.02 | 0.05 | 0.06 | 0.08 | 0.02 | 0.04 | 0.05 | 0.07 |
| $\phi$ | 0.09 | 0.27 | 0.28 | 0.26 | 0.07 | 0.21 | 0.27 | 0.25 |
| $\vartheta$ | 68 | 68 | 69 | 74 | 84 | 84 | 84 | 84 |
| $\chi$ | 1995 | 1097 | 780 | 638 | 2425 | 1336 | 910 | 711 |

(b) BBA-2

mation and bitrate levels were pointed out as the possible root causes. We also believe that by increasing the number of TCP connections, the requirement for a 1xBDP network queue lessens, causing higher bandwidth utilisation. As showed in [14], a large number of TCP connections do not require a 1xBDP network queue.

For BBA-2 in the 1xBDP case, the utilisation increases by 8% as the number of clients increases from 2 to 8. Note, that BBA-2 achieves higher utilisation than FESTIVE for scenarios with a few clients (2,4). When 2xBDP is used for network queue size, the impact of the number of clients on utilisation is not observed. For both cases, instability increases linearly with the number of clients. In the case with only two clients, BBA-2 achieves low unfairness (0.09 and 0.07 for 1xBDP and 2xBDP, respectively). This value increases three times as the number of clients increases from 2 to 4, for both cases. As the number of clients increases, the impact of unfairness is marginal for 1xBDP. The unfairness value peaks when 6 clients share resources, resulting in a 30% increase from the 4 clients case. One of the reason is spacing between subsequent rates, where with 6 clients the gap is around $300kbps$ (clients mostly choose between 765 and $1065kbps$), which together with segment size variation, prompts clients to pick a higher rate. Because of the cushion space in application buffer, which smooths delivery rates, these changes will not occur frequently, but will force other clients to select a lower rate. With 4 clients they are mostly selecting between 1065 and 1777 kbps on average. As the distance between adjacent rates increases, the impact of bandwidth estimation variation is less significant, which forces clients to pick a higher rate more conservatively. As a consequence, the unfairness metric decreases.

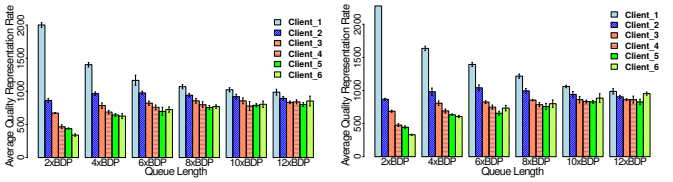
### D. Impact of homogeneous RTT

In this section, we investigate the impact of varying RTT on the performance of FESTIVE and BBA-2. As previoulsy mentioned, it is important to note that low RTT for video streaming is expected to be more common with the current trend of bringing the content distribution network closer to the edge of access networks.

We tested different RTT values including $20ms$, $40ms$, $80ms$, and $160ms$. For every RTT value, we investigated

TABLE V
AVERAGE UNFAIRNESS ACROSS DIFFERENT NETWORK QUEUE LENGTHS (PRODUCTS OF BDP) FOR HETEROGENEOUS RTTS

| Algorithm \ Queue Length | 2x | 4x | 6x | 8x | 10x | 12x |
|---|---|---|---|---|---|---|
| **BBA-2** | 0.60 | 0.43 | 0.37 | 0.34 | 0.30 | 0.30 |
| **FESTIVE** | 0.53 | 0.29 | 0.21 | 0.17 | 0.16 | 0.15 |




(a) FESTIVE (b) BBA-2

Fig. 3. Average Quality Representation Rates for each client across different network queue lengths for heterogeneous RTTs

the impact of varying the bottleneck queue size from 1xBDP to 10xBDP. We only present our key findings due to space limitations. For large RTTs ($80ms$ and $160ms$), BBA-2 and FESTIVE show insignificant changes in bandwidth utilisation, unfairness, and instability in comparison to the $40ms$ RTT case. A slight drop in the average representation rate is observed as the RTT increases. Additionally, a noticeable drop in stalls is observed for $80ms$ case and no stalls are encountered in the $160ms$ RTT case.

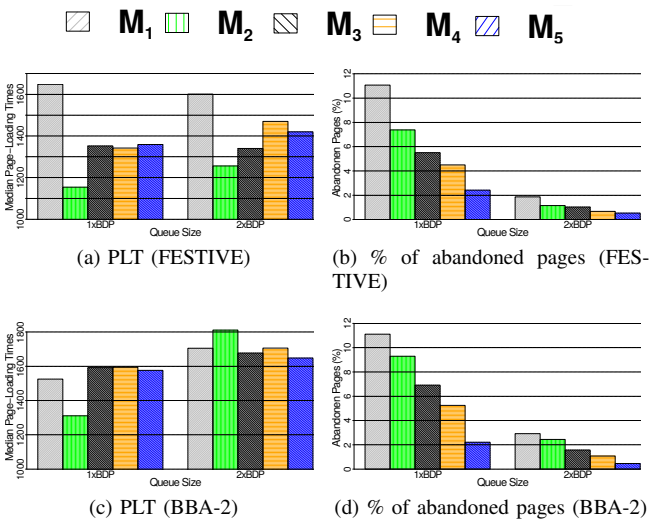
The $20ms$ RTT case shows a noticeable impact on both fairness and stall performance. More specifically, 30% FESTIVE clients encounter at least one stall for the 1xBDP case. Additionally, the unfairness increased by 30% in comparison to the $40ms$ case. Similarly, BBA-2 clients encounter a 30% and 13% increase in unfairness and number of stalls, respectively. For larger queue sizes higher than 1xBDP, FESTIVE and BBA-2 clients show very close performance metrics similar to corresponding queue sizes for the $40ms$ case.

Our results indicate that the streaming performance degradation could be avoided in the case with low RTT, by properly dimensioning the bottleneck queue size.

### E. Impact of heterogeneous RTTs

In practice, clients may not have homogeneous RTT values. For example, with CDN nodes close to the access network there will be lower RTT values for customers of the corresponding video service when compared to other services that are using further away servers. Thus we tested BBA-2 and FESTIVE with 240 and 60 sec client buffers respectively, and for six clients, competing for resources, with their RTT values set to: 20,40,60,80,100,120$ms$. Experiments were repeated with different network queue lengths. All network queue lengths are estimated based on the RTT of the first client ($20ms$). Hence, 4xBDP for the first client is 2xBDP queue for the second client, which has $40ms$ RTT.

When the network queue length is set according to 2xBDP, TABLE V, unfairness has the highest value. This means that the quality rate varies significantly across clients, as depicted in Fig. 3 for both FESTIVE and BBA-2. As the network queue




(a) PLT (FESTIVE) (b) % of abandoned pages (FESTIVE)

(c) PLT (BBA-2) (d) % of abandoned pages (BBA-2)

Fig. 4. Web performance metrics for FESTIVE and BBA-2 vs. network queue size

length increases, clients share resources more fairly. When the network queue equals 12xBDP, unfairness drops significantly, by 70% for BBA-2 and FESTIVE. For the range from 8 to 12xBDP, unfairness drops by around 10% on average, for both algorithms.

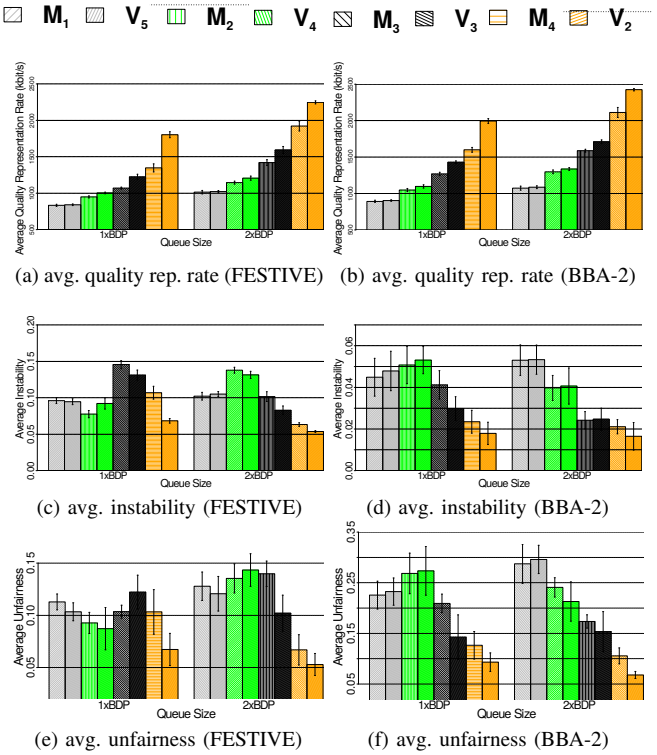
When clients use FESTIVE, Fig. 3a, the client with $20ms$ RTT streams at $2Mbps$ which is 6x higher than the sixth client, who can only manage to use the two lowest rates. With BBA-2 this is even more evident, where the first client manages to stream at $2.2Mbps$, 7x higher than the sixth client. When the network queue length equals 12xBDP, the first client's quality rate drops by 60% and 50% for BBA-2 and FESTIVE, respectively. For both algorithms, all clients stream at close to the average quality rate. Starting from 8xBDP, the difference in rates is 19% and 10% for BBA-2 and FESTIVE, respectively.

As our results show, clients with the smallest RTTs have preferential access to network resources dominating all other clients with higher RTTs. Content providers can take advantage of storing their content closer to the end-user, in order to achieve a better performance. 7xBDP represents average network queue length when all RTTs are taken into account. We argue that dimensioning the network queue length with average RTT value will help improve the system fairness.

## V. Mixed Traffic Experiments

The bottleneck link may be shared among different types of traffic, including video, and interactive, delay-sensitive applications. Hence, our next set of experiments considers a mixture of HAS and web clients sharing the bottleneck link. With six clients sharing the link, we consider scenarios with $n$, where $n \in 1..6$, web clients and $6-n$ HAS clients. In the rest of the paper, we use $M_n$ to refer to sharing scenario with $n$ web clients. We again used the default simulation parameters shown in TABLE I.

First, we considered $M_6$ scenario;i.e. six web clients sharing the link. We found that increasing the queue size, reduces the PLT and the percentage of abandoned sessions to 10/11




(a) avg. quality rep. rate (FESTIVE) (b) avg. quality rep. rate (BBA-2)

(c) avg. instability (FESTIVE) (d) avg. instability (BBA-2)

(e) avg. unfairness (FESTIVE) (f) avg. unfairness (BBA-2)

Fig. 5. HAS performance metrics with web traffic vs. network queue different Y-aes ranges

and 1/3, respectively, as the network queue increases from 1x to 2xBDP. We believe this is because small network queues drop packets more frequently, causing TCP to retransmit lost packets. This adds to the overall delay, causing the user to abandon more web sites. Note that our user abandons the webpage if it doesn't load in 15 sec [20]. This result indicates that larger buffers could also benefit web clients sharing a bottleneck link. Next, we focus on mixed video and browsing traffic scenarios, i.e., $M_1$ - $M_5$. Fig. 4 and Fig. 5 show the key performance metrics for web and HAS clients, respectively, versus the network queue size for $M_1 - M_5$ scenarios. For web clients, we found that the percentage of abandoned pages decreases as the queue size increases and when the number of video client decreases as shown in Fig. 4b and Fig. 4d. However, Fig. 4a and Fig. 4c suggest that the web PLT is scenario-dependent. E.g., for the $M_2$ scenario with 1xBDP, PLT values drop for FESTIVE and BBA-2. Looking at the video client's rates, we notice that four video clients have similar rates as for $M_1$ scenario, which has five video clients beside one web client. Because of the gap between 1 and $1.7Mbps$ rate, four clients don't have enough resources to compete for the next higher rate. As a result, the network queue is less occupied with video traffic, allowing web clients to download resources quickly. Fig. 5a and 5b show that the average video rate of HAS clients increases with the number of web clients as fewer HAS clients would be competing for the link. However, the perceived average quality rate noticeably decreases as the number of web clients increases in comparison to corresponding $V_{6-n}$ scenarios. Fig. 5c- 5f illustrate that the unfairness

and instability are scenario-dependent as web clients may change these metrics positively or negatively in comparison to $V_{6-n}$ scenarios. We believe that such scenario dependency is due to the interaction between the link capacity, number of clients, and available video encoding rate. To illustrate, fairness and instability changes differently across different $M_n$ scenarios in comparison to $V_{6-n}$ scenarios. In the $M_3$ and $V_3$ scenarios with 1xBDP for FESTIVE, median quality rate is $1Mbps$. In $M_3$ case, unfairness metric drops because $700kbps$ generated by web clients (237 $kbps$ per client), fills the gap between 1065 and $1777kbps$. This forces all the video clients to stream at the $1Mbps$. Without the web traffic, case $V_3$, two clients select higher rates, 1777 and $2335kbps$, forcing third client to use $1Mbps$. This increases unfairness and instability, which seems counter-intuitive. However, FESTIVE's probe component forces clients to periodically select a higher rate, as a function of rate. For lower rates, the algorithm will try next higher rate more often. This explains why instability increases by 10%. For the 2xBDP case, the same pattern can be observed, but because the larger network queue increases available throughput, aforementioned interpretations shift to scenarios $M_2$ and $V_4$, respectively. BBA-2 relies on buffer levels to make a decision. When video clients operate in a region around $1Mbps$, the amount of web traffic helps to decrease both instability and unfairness. This is a consequence of slower buffer filling, which essentially causes clients to be more stable and fair. When the number of web clients increase, variability in perceived throughput increases and video clients switch more often, increasing the unfairness metric.

## VI. Conclusion

Selecting optimal network queue length can have a significant impact on HTTP Adaptive video Streaming in scenarios where multiple such HAS clients share a bottleneck. In this realistic context, we studied the effect of network queue size. Our experimental study shows that the recommended "rule-of-thumb" for queue size can lead to underutilisation of network resources, where clients use only 70% of available bandwidth. We demonstrated that setting the buffer size to 2xBDP would increase both system utilisation and average video quality rate by 15% on average. Additionally, we show that increasing the buffer size can also help to improve the system fairness when the video clients have different heterogeneous RTTs. For mixed traffic scenarios where video and web traffic share a bottleneck, we also show that increasing the queue size would reduce the percentage of abandoned web pages but its impact on page loading times is scenario-dependent and would vary depending on bitrate distribution, video adaptation algorithm and offered web traffic load.

Future work includes testing different scheduling mechanism and TCP variants and quantifying their impact on HAS performance.

## Acknowledgment

The authors acknowledge the support of Science Foundation Ireland (SFI) under Research Grant 13/IA/1892.

## References

[1] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What Happens when HTTP Adaptive Streaming Players Compete for Bandwidth?" in *Proceedings of the 22Nd International Workshop on Network and Operating System Support for Digital Audio and Video*, ser. NOSSDAV '12, 2012, pp. 9–14.

[2] J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet," *Queue*, vol. 9, no. 11, pp. 40:40–40:54, Nov. 2011.

[3] A. Mansy, B. Ver Steeg, and M. Ammar, "SABRE: A Client Based Technique for Mitigating the Buffer Bloat Effect of Adaptive Video Flows," in *Proceedings of the 4th ACM Multimedia Systems Conference*, ser. MMSys '13, 2013, pp. 214–225.

[4] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH)," in *Packet Video Workshop (PV), 2013 20th International*, Dec 2013, pp. 1–8.

[5] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive," *Networking, IEEE/ACM Transactions on*, vol. 22, no. 1, pp. 326–340, Feb 2014.

[6] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, April 2014.

[7] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, Aug. 2014.

[8] J. J. Quinlan, A. H. Zahran, K. K. Ramakrishnan, and C. J. Sreenan, "Delivery of adaptive bit rate video: balancing fairness, efficiency and quality," in *The 21st IEEE International Workshop on Local and Metropolitan Area Networks*, April 2015, pp. 1–6.

[9] R. Houdaille and S. Gouache, "Shaping HTTP Adaptive Streams for a Better User Experience," in *Proceedings of the 3rd Multimedia Systems Conference*, ser. MMSys '12, 2012, pp. 1–9.

[10] C. Villamizar and C. Song, "High Performance Tcp in ANSNET," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 45–60, Oct. 1994.

[11] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12, 2012, pp. 225–238.

[12] K. Nichols and V. Jacobson, "Controlling Queue Delay," *Queue*, vol. 10, no. 5, pp. 20:20–20:34, May 2012.

[13] R. Adams, "Active Queue Management: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1425–1476, Third 2013.

[14] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing Router Buffers," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 281–292, Aug. 2004.

[15] B. Lantz, B. Heller, and N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-defined Networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX, 2010, pp. 19:1–19:6.

[16] A. Keller, *tc Packet Filtering and netem*, ETH Zurich.

[17] European Union, "Quality of Broadband Services in the EU," http://ec.europa.eu/newsroom/dae/document.cfm?action=display&doc_id=10816, 2013.

[18] Jason J. Quinlan, Ahmed H. Zahran, Cormac J. Sreenan, "Datasets for AVC (H.264) and HEVC (H.265) Evaluation of Dynamic Adaptive Streaming over HTTP (DASH)," in *Proceedings of the 7th ACM Multimedia Systems Conference*, ser. MMSys '16, 2016.

[19] I. Tsompanidis, A. H. Zahran, and C. J. Sreenan, "Mobile network traffic: A user behaviour model," in *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP*, May 2014, pp. 1–8.

[20] C. Liu, R. W. White, and S. Dumais, "Understanding web browsing behaviors through weibull analysis of dwell time," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '10, 2010, pp. 379–386.

[21] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *Communications Surveys Tutorials, IEEE*, vol. 17, no. 1, pp. 469–492, Firstquarter 2015.