Delivery of Adaptive Bit Rate Video: Balancing Fairness, Efficiency and Quality

Jason J. Quinlan[‡], Ahmed H. Zahran[‡⋆], K.K. Ramakrishnan[†], Cormac J. Sreenan[‡]
[‡]Dept. of Computer Science, University College Cork, Ireland
[⋆]Dept. of Electronics and Electrical Communications, Cairo University, Egypt
[†]Dept. of Computer Science and Engineering, University of California, Riverside

***Abstract*—HTTP streaming currently dominates Internet traffic. It is increasingly common that video players employ adaptive bitrate (*ABR*) streaming strategies to maximise the user experience by selecting the highest video representation while targeting stall-free playback. Our interest lies in the common situation where a set of video flows are competing for access to a shared bottleneck link, such as in a cellular radio access network. We observe that ISPs (e.g. cellular operators) are considering in-network techniques for resource allocation and sharing among different users. Buoyed by the ability of software defined networks (*SDN*) to offer flow-specific control and traffic shaping, we focus on traffic shaping techniques, and experimentally analyse the effect on ABR video flows when sharing a bottleneck link. We conduct experiments using the GPAC video player operating over a Mininet virtual network. We conclude that traffic shaping can allow a balance of fairness, efficiency and quality. Traffic shaping ABR videos reduce the number of stalls and quality switches, while also reducing the peaks for the aggregate network traffic.**

***Index Terms*—Adaptive Bitrate Streaming (ABR), Dynamic Adaptive Streaming over HTTP (DASH), Traffic Shaping**

I. Introduction

Adaptive bit rate (*ABR*) streaming over HTTP is considered the default streaming approach for many video providers (*VPs*), such as Netflix, Hulu, and Youtube. Currently, HTTP-based streaming is the dominant multimedia streaming protocol [1]. This choice is driven by the availability of infrastructure and the ability of HTTP to bypass firewalls. By 2012, mobile video traffic was nearly 50% of the total mobile data traffic and is expected to reach to 75% by 2019 [2]. More importantly, video traffic corresponds to a significant portion of the traffic during the peak hours. For example, Netflix traffic alone represents 30% of the traffic during the peak hours in US. Hence, cellular operators seek to optimise the delivery of video, while striking a balance between maintaining quality and being fair to other traffic that is carried on the network. Paramount in this effort is to ensure that the user perceived quality of experience (*QoE*) is maintained even as different parameters and objectives are considered by both the ISP and VP control loops.

The VP manages the streaming control loop to improve the user perceived QoE, which is quantified by several metrics including initial playout latency, received video quality, and seamless stall-free streaming. On receiving a video segment, streaming clients employ different strategies to select the next segment quality to be streamed. These strategies may consider one or more parameters such as the client buffer level, estimated throughput, segment size, last segment quality, and frequency of video representation level changes. ISPs too employ several techniques, such as caching, transrating and rate control/traffic shaping, to manage video traffic. These techniques target improving the utilisation of their network resources and also ensure that reasonable service is provided to other coexisting applications. Some of these ISP techniques may interfere with the application-level decisions and could affect the perceived QoE. For example, transrating would change the video encoding rate to reduce the network load, and may not correspond with the video client's decision making for the ABR video stream. Traffic shaping, sometimes also called rate limiting, of the video flow to a specific rate that seeks to ensure a fair resource allocation among all user flows needs to ensure it cooperates with the VP's control loop. Traffic shaping can also smooth common large variations in the encoded video. Traffic shaping has several advantages including keeping the application in charge of the received segment quality selection, reduced computational overhead in comparison to transrating, and reduces the peaks in the network traffic. Traffic shaping is already supported by the state of the art networking approaches such as software defined networking (*SDN*).

SDN also represents a novel flexible framework that offers different options for improving the streaming quality. SDN programmability allows dynamic routing in the network supported by monitoring capabilities at a flow-level. Hence, different flows can be treated differently. For example, video flows may be assigned routes that satisfy pre-defined quality of service goals and can be routed toward servers for caching and transcoding purposes. Additionally, SDN also supports QoS functions such as traffic shaping that limits the amount of resources consumed by different flows in the network. SDN may offer new opportunities for better coordination, and finer granularity, between the streaming and resource management control loops.

The interaction between the streaming and resource management control loops raises several design concerns. In this work, we focus on investigating the impact of traffic shaping on the performance of ABR streaming. More specifically, we are interested in investigating the impact of aggregate vs individual traffic shaping strategies for bottleneck links. We setup a network testbed as described in Section III and present our performance evaluation results in Section IV. Our results indicate that individual traffic shaping maintains a balance in terms of perceived video quality (data rate and stalls), demand for network bandwidth, and fairness across different users.

II. Background and Related Work

ABR algorithms consider different approaches for identifying the selected segment quality to be streamed. The operation of these algorithms may have two, or more, phases, namely: initial and steady state. The former is considered a probing conservative stage with two main objectives: reducing the initial playout latency and gradually increasing the streaming rate. During the latter stage, the controller targets maintaining the application buffer level at a predefined range while maintaining a fixed quality. However, it is not uncommon that bitrate estimation based on application goodput is employed to identify the instantaneous network capacity as a guideline for the rate bound of the next segment in any of these phases. Jiang et. al. [3] employ a harmonic mean of the last k estimated rates for recently downloaded segments to identify rate bounds. This rate estimate represents a core input for their decision engine that considers other parameters such as current quality and rate of quality switching. On the contrary, Huang et. al. [4] propose a buffer based approach in which the selected quality mainly depends on the buffer level using a predefined rate map. However, rate estimation is suggested to improve the performance during transient periods; i.e. stream start or jumps. These examples illustrates that the reaction of ABR algorithms to rate change may vary depending on the design of the decision engine.

In [5], Akhshabi et. al. investigate the performance of a group of ABR clients competing for a limiting bandwidth during steady state, when the player secures sufficient amount of media in its buffer, using Microsoft smooth streaming (*MSS*) player and their own player for more experimental control. At steady state, the authors point out that the streaming client encounters cycles of activity and inactivity periods during which it is incapable of estimating the link bandwidth leading to performance deterioration including frequent representation switches. Additionally, different users, even using the same player, would have different estimates for the available bandwidth leading to variations in the selected segment quality. A solution for the latter issue is introduced in [3] by enabling randomising segment quality request time. In [6], Houdaille and Gouache investigate the impact of shaping HTTP traffic in a home gateway setting when two clients share an ADSL link using MSS and Apple's HLS client. The authors show that shaping results in reducing the number of quality changes and oscillations for both MSS and HLS.

Software defined networking (*SDN*) is a novel networking architecture in which the control and data planes are split to simplify network management and allow innovations in networking at the speed of software production. Typically, the control plane decisions are taken at a central node that has a global network view enabled by monitoring capabilities for forwarding nodes. Monitoring and control traffic is communicated using a secure interface between the controller and forwarding nodes and is known as the southbound interface.

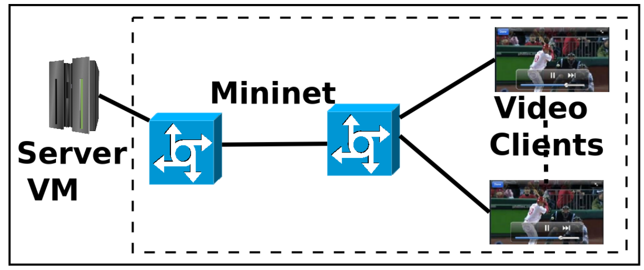


Figure 1: Testbed comprising a video server and clients operating over a virtual network

The Openflow protocol (*OF*) is considered the most widely supported protocol in the southbound. On top of the controller, networking applications are employed to perform different functions including, but not limited to, routing, security, load balancing, and network optimisation.

Several proposals in the literature take advantage of SDN capabilities to improve video streaming performance. Motivated by flexible dynamic routing capabilities of SDN, [7], [8] propose prioritised routing for multimedia streams that are monitored to ensure that selected routes satisfy QoS requirements and reroute of such constraints are violated. Other proposals take advantage of northbound interface to enable interaction between the network and streaming client or server. In [9], [10], client feedback is used to reroute the flow when the buffer level drops. Georgopoulos et. al. [11] propose a framework for fair resource allocation among a group of users and that such decisions are communicated back to the client using the northbound interface. In [12], vSDN is proposed as a novel architecture to enable optimal path selection for multimedia traffic. Additionally, guaranteed bitrate is proposed for multimedia streams by taking advantage of traffic shaping in OF version 1.3. Our work complements these efforts by investigating the interaction between streaming control loop and resource management techniques and its impact on the user perceived QoE. More specifically, we focus on the impact of traffic shaping, such as introduced in OF version 1.3, on the performance of streaming applications.

III. Experimental Setup

Figure 1 illustrates our testbed setup comprising a video server and clients operating over a virtual network. The clients share a bottleneck link, similar in principle to the situation in a cellular access network. Our testbed is based on three distinct Virtual machines (*VM*) using the VirtualBox Manager for virtualisation. Our testbed is composed of a Server VM, a network VM and a Client VM. We use Ubuntu 14.04 desktop for the Server and Client VMs, and Ubuntu 14.04 Server for the network VM. To provide the variation and complexity that can occur in a deployment network, we have implemented the network VM using Mininet [13]. Mininet permits us to not only design and implement large scale networks, upon which we can transmit DASH streams but to also offer the promise of control plane and data plane separation by using in-built SDN techniques and protocols. While currently not implemented, the ultimate goal of our testbed is to provide a means of

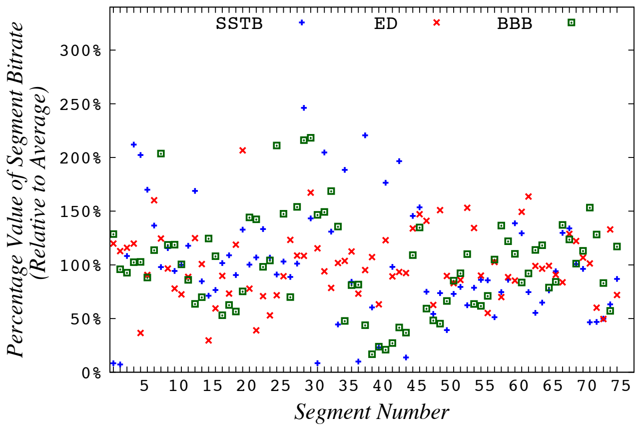


Figure 2: Percentage value of the video segment bitrate relative to the average, for each of the three clips

investigating the benefits provided by SDN in improving QoE for streaming video.

In our testbed, the Server and Client VMs are separately connected to the Mininet VM, such that the requests between the Client and the Server are routed over a defined SDN mininet network. By leveraging a separate VM per component, modifications to any aspect of the testbed can be facilitated with ease without impacting on the existing functionality or deployments within the testbed. Examples of modification include evaluation with an updated O/S, an updated DASH implementation, new adaptation algorithms, etc.

A. DASH Encoding Setup

To create our DASH files, we utilise three well-known videos, Big Buck Bunny (*BBB*), Sita Sings the Blues (*SSTB*), and Elephant Dreams (*ED*), which were obtained as YUV files from [14]. From these YUV, a five-minute DASH clip is generated using MP4Box and X264, as per the instruction provide by Bitdash [15]. We encode nine different representation rates which are averaged at 6Mbps, 5Mbps, 4Mbps, 3Mbps, 2.5Mbps, 2Mbps, 1.5Mbps, 1Mbps, and 0.5Mbps, and we distribute these rates equally between three resolutions, namely 854x480, 640x360, and 480x272. It is important to note that the underlying resolution selected is not important, as the representation rate will govern overall transmission cost, such that by choosing a different resolution during encoding would not increase the transmission cost of the representation level selected.

The DASH content is composed of a global media presentation description (*MPD*) file and a number of segments per representation. Each individual segment provides the DASH client a means of changing quality level as the condition of the network changes. We define a segment duration of 4-seconds. The DASH content is then stored on an Apache server on the Server VM. For the three clips, we would like to point out that the actual encoded video rate of the different segments within the clip, features significant variations from the aforementioned average representation bitrate. To illustrate, Figure 2 plots the average percentage of individual segment rates with respect to the average bitrate. Note that this average value is applicable to all average encoded rates as the scale per segment for different average rate is approximately identical. Hence, on evaluating the fairness across different flows, we use the actual received bitrate from the network in contrast to using the rate of the selected quality.

The GPAC [16] client, MP4Client, is used with its default configuration in all our experiments. In our setup, we assume that a network resource manager allocates the bandwidth for different services; e.g. voice, video, and background. The amount of bandwidth allocated to video traffic would be shared across different video flows. We investigate two different sharing strategies including aggregate (*AG*) and individual (*IN*) schemes. In the former, all video streams share a single bottleneck whose capacity equals the allocated bandwidth while in the latter each active stream is allocated an equal share of the total video bandwidth and would not be able to use more than the allocated rate share.

Our key performance metrics include video quality metrics such as the streamed video bitrate as selected by the client, the fraction of time for different qualities, the number of quality switches performed by the clients for adaptation, number of video stalls, and stall durations. We also consider network related metrics such as resource utilisation and fairness, which is estimated as [17]

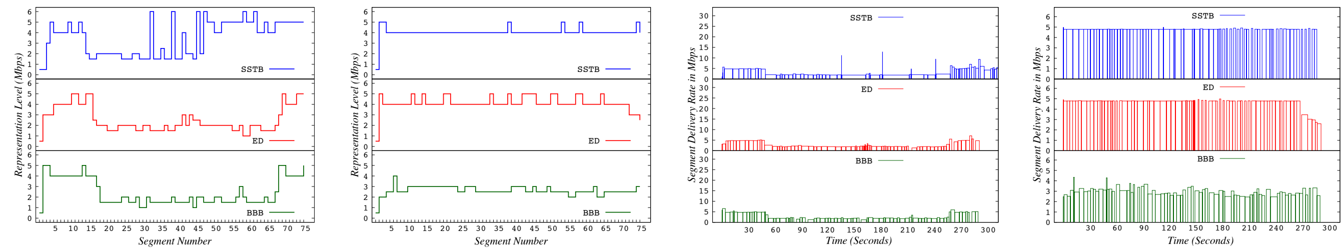
$$FI = \frac{1}{N}\sum_{n-1}^{N}\frac{1}{\overline{x}_n}\sqrt{\frac{1}{K}\sum_{k-1}^{K}(x_{nk}-\overline{x})^2},$$

where K is the number of streaming clients, N is the total number of time points, x_{nk} represents the observed rate by client k at time instant n, and $\overline{x}$ represents the average of all sampled rates.

IV. Performance Evaluation

The aggregate and individual traffic shaping scenarios are simulated five times with six-clients. In the aggregate case the clients are competing for *30Mbps* in every run. In the individual scenario, each user is assigned *5Mbps* of the available bandwidth. Figure 4 plots the fraction of segments received at specific quality for both aggregate and individual cases. The figure suggests that the aggregate case provides more segments at high quality representations. This occurs because the GPAC segment requester is throughput-based. In the aggregated scenario, high representations are requested when the bandwidth estimate is high when the previous segment is delivered quickly. However, in many cases the requested high quality representation is associated with high delivery delay.

Figure 3 illustrates such segment delivery dynamics including the delivered quality and the rate at which segments are delivered. More specifically, Figure 3a shows large oscillation in the requested quality for the aggregate scenario. Such large oscillations may occur when a small segment size is followed by a large segment when the former one is delivered fast, the client assumes that it has a large bandwidth and request a segment at a high quality representation level, which takes a long time for download (indicated as a wide pulse in Figure 3c). For



(a) Segment quality vs. segment ID for aggregate case
(b) Segment quality vs. segment ID for individual case
(c) Segment delivery rate vs. time for aggregate case
(d) Segment delivery rate vs. time for individual case

Figure 3: Segment Delivery Dynamics

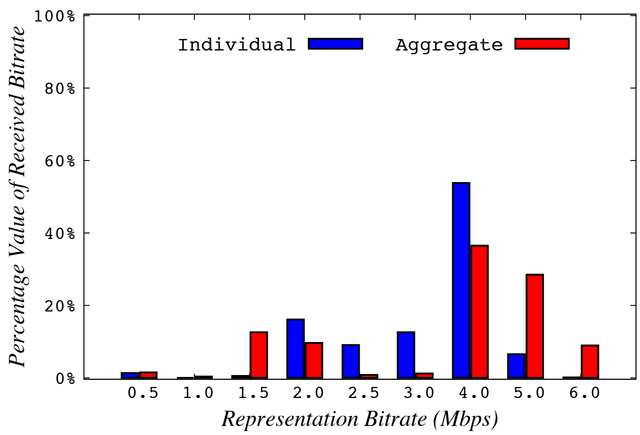


Figure 4: Percentage value of the the received segment representation bitrate for the Individual and Aggregated cases

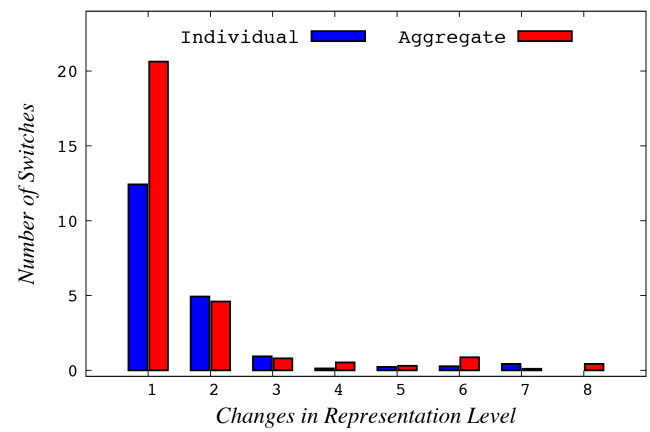


Figure 5: Switching dynamics - representation variations encountered in the Individual and Aggregated cases

example, the segment delivery rate of SSTB in Figure 3c show three spikes which corresponds to small segments delivered to the client in a very short period indicating to the client a high available bandwidth, which is beyond the clients fair share. Consequently, the streaming client requests the next segment, which happens to be a large segment as a result of video encoding, at the highest representation. This segment may require a longer than expected delivery time, that may result in buffer under-run and this switching to a large bitrate change is shown in Figure 3a. On the contrary, the rate control in the individual case does not allow for such huge variation in the requested segment quality, as shown in Figure 3b, and the segment delivery rate as shown in Figure 3d.

Representation switches is another important video quality metric. Our results suggest that both aggregate and individual schemes encounter a smaller number of switches per session, 18 and 25 for the individual and aggregate schemes, respectively. Figure 5 plots a histogram for the representation variations encountered in every scheme. The figure suggests that individual scheme results in smooth switches in comparison to aggregate scheme. To illustrate, Figure 5 shows that the aggregate scheme may result in large bitrate variations that are usually triggered by variations in throughput estimated due to the large available bandwidth. Hence, regulating the rate of every session to a pre-defined bounded rate will help the streaming client to take better decisions.

A more serious drawback in the aggregate scheme is the encountered streaming stalls, which are usually considered the most annoying quality degradation from the users perspective. In this scheme, the individual scheme results in significantly reduced average number of stalls in comparison to the aggregate scheme. Individual scheme sessions encounter on average 1.5 stalls while the streamed videos encounter an average of 8.38 stalls in the aggregate scheme. Additionally, the clients encounter longer stalls in the aggregate scheme. Figure 6 plots a histogram for the durations of the encountered stalls in our experiments for both aggregate and individual schemes. For individual scheme, the average stall duration is 1.58 sec in comparison to 2.38 sec for the aggregate scheme. Additionally, the standard deviation of stall duration for individual and aggregate schemes are 1.5 and 3.8, respectively, indicating that the competition between ABR video traffic for the shared bandwidth in the aggregate case significantly affects the streaming performance.

Average fairness metric for the aggregate and individual metric are 0.27 and 0.29, respectively. These figures suggest that both schemes provides a similar level of fairness on average. However, our further analysis indicates that the aggregate scheme features significant variation in the fairness level encountered across different runs while the individual scheme performance is more consistent across multiple runs. A similar behaviour is observed for the bandwidth utilisation

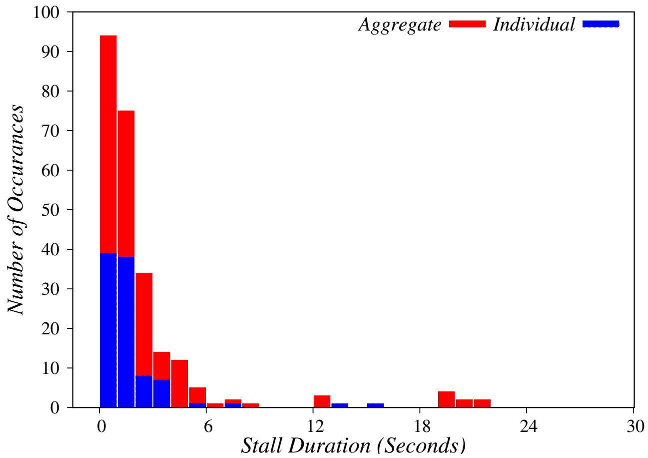
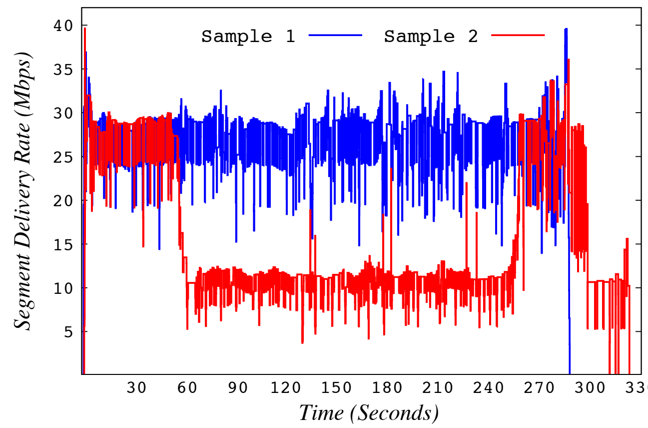


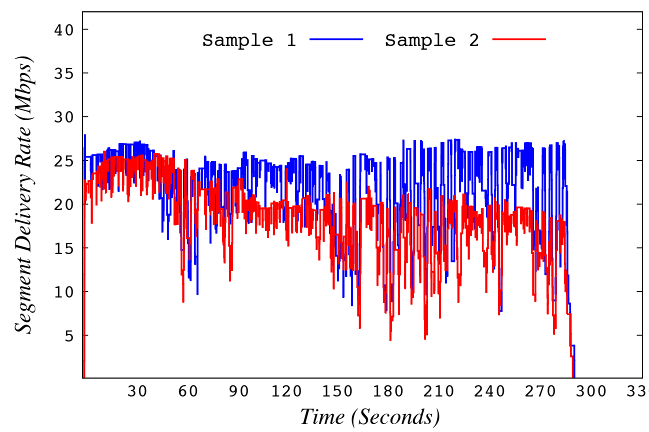
Figure 6: Durations of the encountered video stalls for the Individual and Aggregate cases

for which aggregate scheme shows different patterns. Figure 7 plots the total utilised bandwidth by all users for two sample runs, from the thirty sessions evaluated, for both the aggregate and individual scheme. Figure 7a illustrates that the competition among different video flows in the aggregate scheme could lead to significant under-utilisation of network resources as shown for sample 2, which is operating at a rate much lower than is available to it from the network. Such inefficient utilisation of bandwidth leads to extended session duration (due to stalls) and low video quality. Note that the video session lasts till 330 seconds while the total video length is 300 seconds and should be typically downloaded. Hence, this 30 second difference indicates that some videos stall and needed more time to complete. On the contrary, the utilisation pattern of the individual scheme shows consistent patterns across all runs as illustrated by the samples shown in Figure 7b.

We also investigate the case where allocating the same amount of resources in the individual case may result in different bitrates due to variations in the link spectral efficiency as observed by a user device. In this experiment, we compare the performance of two clients streaming SSTB, which is chosen as it contains the greater distribution of segment bitrate relative to the average. The allocated resources for these clients would allow for bitrates of 5Mbps and 2.5Mbps for the first and second client, respectively. We further consider two scenarios: a case in which the MPD file is modified such that higher representations beyond the allocated bitrate are removed (chopped) and a case when the client uses an unmodified MPD file. Figure 8 plots the representation selection and segment delivery rate for different segments for the aforementioned scenarios. Similar rate switches and segment delivery rates are observed across clients with similar allocated rates for both the modified and unmodified scenarios. This observation suggests that the decisions made by the clients are very similar irrespective of the MPD file provided to the clients when the rate is controlled. We also noticed that the number of stalls, their durations and pattern are quite



(a) Aggregate Case



(b) Individual Streams

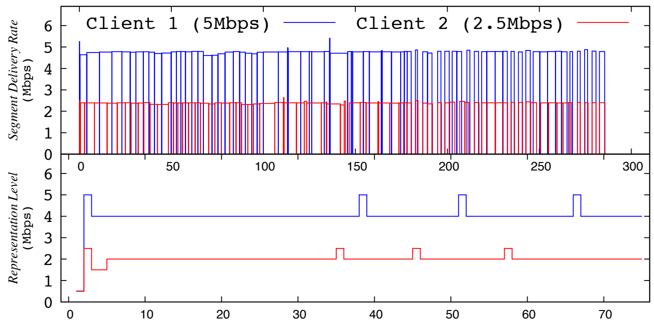
Figure 7: Sample of the total utilised bandwidth in the Individual and Aggregate cases. Two samples for Individual and Aggregate are illustrated from the thirty sessions evaluated

similar. However these metrics are not shown due to space limitation.

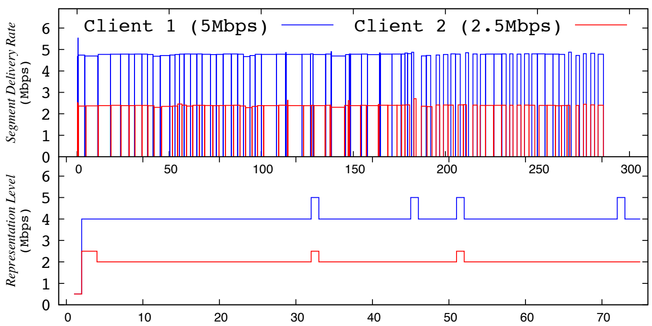
V. Discussion and Conclusions

The interaction between video streaming and resource management strategies has a strong impact on the user perceived quality of service and resource utilisation. In this work, we show that traffic shaping can be employed to balance different design objectives, including perceived video quality, fairness and network utilisation. Using a Mininet testbed and a number of different video clips encoded at multiple representation rates, we showed that traffic shaping in the network can indeed help reduce the number of stalls and rate-switching for ABR video while at the same time reducing the peaks in network bandwidth demand from a set of aggregate video flows. Thus, traffic shaping of ABR video does not result in adverse interactions with the control loop between the ABR client and the video provider based on our initial studies.

However, other relevant questions remain open for investigation in our future work. In this work, we have used the GPAC DASH player MP4Client. It would be interesting to



(a) Streaming Dynamics with an unmodified MPD file



(b) Streaming Dynamics with a modified MPD file

Figure 8: Representation selection and segment delivery rate for 5Mbps and 2.5Mbps

examine other adaptation strategies [3], [4] to identify which would have a better response to changes in network conditions. A more in-depth analysis needs to be conducted for the parameters of different streaming strategies, e.g. the selected buffer levels and rate estimators. The design of the traffic shaping strategy is another interesting problem on its own, e.g. [18]. Also identifying when to start traffic shaping and what could be the best rate bound are also open problems. Should traffic shaping start at the beginning of the session or it is better to delay shaping until the client buffer secures a few segments to reduce the initial playout latency? Would the selected rate only consider the available resources or might it consider other parameters such as the encoding rates? Would the knowledge of encoding rates enable us to use a lower rate limit for the benefit of background traffic without significantly affecting the streaming performance? The triggers and frequency of traffic shaping actions are also important design parameters for the traffic shaping strategy. Another interesting question would be whether the network is able to effectively control the streaming loop if the client behaviour is known, i.e. by setting the rates appropriately. Hence, devising traffic shaping strategies that can achieve an optimal balance between perceived video quality and efficient resource utilisation is the subject of our ongoing work.

Acknowledgment

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

References

[1] A. Gerber and R. Doverspike, "Traffic types and growth in backbone networks," in *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, March 2011, pp. 1–3.

[2] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014-2019," 2014. [Online]. Available: http://goo.gl/jFB2L7

[3] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12. New York, NY, USA: ACM, 2012, pp. 97–108. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413189

[4] 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," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 187–198. [Online]. Available: http://doi.acm.org/10.1145/2619239.2626296

[5] 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. New York, NY, USA: ACM, 2012, pp. 9–14. [Online]. Available: http://doi.acm.org/10.1145/2229087.2229092

[6] 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. New York, NY, USA: ACM, 2012, pp. 1–9. [Online]. Available: http://doi.acm.org/10.1145/2155555.2155557

[7] H. Egilmez, S. Dane, K. Bagci, and A. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks," in *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, Dec 2012, pp. 1–8.

[8] S. Laga, T. Van Cleemput, F. Van Raemdonck, F. Vanhoutte, N. Bouten, M. Claeys, and F. De Turck, "Optimizing scalable video delivery through OpenFlow layer-based routing," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, May 2014, pp. 1–4.

[9] M. Jarschel, F. Wamser, T. Hohn, T. Zinner, and P. Tran-Gia, "SDN-Based Application-Aware Networking on the Example of YouTube Video Streaming," in *Software Defined Networks (EWSDN), 2013 Second European Workshop on*, Oct 2013, pp. 87–92.

[10] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards QoE-aware Video Streaming using SDN," in *Proc. of IEEE Globecom 2014*, 2014.

[11] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards Network-wide QoE Fairness Using Openflow-assisted Adaptive Video Streaming," in *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking*, ser. FhMN '13. New York, NY, USA: ACM, 2013, pp. 15–20. [Online]. Available: http://doi.acm.org/10.1145/2491172.2491181

[12] H. Owens and A. Durresi, "Video over Software-Defined Networking (VSDN)," in *Network-Based Information Systems (NBiS), 2013 16th International Conference on*, Sept 2013, pp. 44–51.

[13] 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. ACM, 2010, pp. 19:1–19:6.

[14] "Video Traces for Network Performance Evaluation. Available at: https://media.xiph.org/video/derf/."

[15] "Bitdash - MPEG-DASH Content Generation Using MP4Box and X264. Available at: http://www.dash-player.com/blog/2014/11/mpeg-dash-content-generation-using-mp4box-and-x264/."

[16] GPAC. [Online]. Available: gpac.wp.mines-telecom.fr/

[17] V. Aggarwal, R. Jana, J. Pang, K. Ramakrishnan, and N. Shankaranarayanan, "Characterizing fairness for 3G wireless networks," in *Local Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on*, Oct 2011, pp. 1–6.

[18] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A Scheduling Framework for Adaptive Video Delivery over Cellular Networks," in *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, ser. MobiCom '13. New York, NY, USA: ACM, 2013, pp. 389–400.