Volume: 09 Issue: 06 Pages: 3607-3615 (2018) ISSN: 0975-0290

Bandwidth Estimation Techniques for Relative 'Fair' Sharing in DASH

Koffka Khan
Department of Computing and Information Technology The University of the West Indies, Trinidad and Tobago, W.I
Email: koffka.khan@gmail.com
Wayne Goodridge
Department of Computing and Information Technology The University of the West Indies, Trinidad and Tobago, W.I
Email: wayne.goodridge@sta.uwi.edu.com

ABSTRACT

In the adaptive video streaming (AVS) literature the term fair sharing has been used to describe equal amounts of bandwidth allocated to adaptive client players. However, we argue that even though bandwidth sharing is an important aspect in some problems the same does not apply to AVS. Here the term relative 'fair' sharing is more applicable. The reason is that videos have different quality levels and will require differing amounts of the bandwidth to satisfy their needs. A 90% to 10% sharing may be sufficient for two players, one with high demands and the other with low demands. A 50% sharing may lead the player with the high bandwidth demand to get too little of the needed bandwidth resource. In addition, channel conditions may lead to players requiring different amount of bandwidth. Again, the concept of fair sharing has to be extended to relative 'fair' sharing for such scenarios. Hence, bandwidth estimation techniques players use to estimate the network bandwidth is very important in segment selection. A player utilizes one of the many techniques to determine what share of the bandwidth it can utilize among competing players. In this paper we explore some of the techniques used in state-of-the-art players in their attempt to obtain a 'fair' share of the network bandwidth.

Keywords - **Adaptive video streaming; bandwidth; demand; sharing.**

Date of Submission: May 08, 2018 Date of Acceptance: May 19, 2018

I. INTRODUCTION

The concept of adaptive video streaming is based on the idea to adapt the bandwidth required by the video stream to the throughput available on the network path from the stream source to the client[30]. These algorithms can live at the server[24], at an intermediate network device [18]or at the client[4]. With the client-side approach it is the player that decides what bitrate to request for any fragment, improving server-side scalability. A benefit of this approach is that the player can control its playback buffer size by dynamically adjusting the rate at which new fragments are requested. The adaptation is performed by varying the quality of the streamed video. Multiple video segments constitute a video stream lasting from as little as 2 seconds to as much as having a 10 second chunk delivery rate. Segments are encoded and stored on the server in numerous quality versions, termed representations. Each version has a unique resolution, bitrate and/or quality. A client downloads segments using HTTP GET statements [12]. However, with adaptive streaming a client might request subsequent segments at different quality levels to manage varying network conditions, based on an estimation bandwidth. To do this it uses a manifest file that contains information about the video segments. Protocols and standards such as MPEG Dynamic Adaptive Streaming over HTTP (DASH)[13], Apple HTTP Live Streaming (HLS)[20], Microsoft Smooth Streaming (MSS)[31] or Adobe HTTP Dynamic Streaming (HDS) [10]typically use a media playlist that contains a list of uniform resource identifiers (URIs) that are addresses to media segments. The process of determining the ideal representation for each segment to enhance the user's experience is pivotal to adaptive streaming. The controller algorithm estimates the network bandwidth and chooses the next bitrate level corresponding to the available network bandwidth. Variations in the available bandwidth will result in jerky playback and disruption of the video playback if the throughput falls below the bit rate requirement of the video. This is the major challenge in adaptive video streaming. Selecting appropriate bitrate levels helps to maximize the user experience. Generally, higher bitrates and resolutions will give better user experience. For example, if a client approximates that there is 9.5Mb/s available in the network, it might request the server to stream video compressed to the highest video rate available, 9.5Mb/s, or the next rate below, 9.3Mb/s. If the client picks a video rate that is too high, the viewer will experience annoying re-buffering events; if they pick a streaming rate that is too low, the viewer will experience poor video quality. In both cases, the experience degrades[25], [17], [8]and user may take their viewing elsewhere[14]. It is therefore important for a video streaming service to select the highest safe video rate.

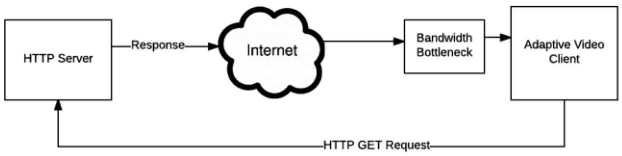

Figure 1: Conventional Adaptive Streaming

Adaptive streaming uses the HTTP/TCP protocol stack to transmit video Web traffic. Thus, the development of

Volume: 09 Issue: 06 Pages: 3607-3615 (2018) ISSN: 0975-0290

this wave of HTTP-based streaming applications is not referred to as adaptive streaming over HTTP. The use of HTTP/TCP protocols for video streaming is because of the advantages that HTTP/TCP offers. It allows standard web servers and caches to be used increasing its' cost effectiveness. Another advantage is that all firewalls are configured to support HTTP connections[29]. In addition, is allows better scaling as HTTP is stateless and the streaming session is managed by the client, thus reducing the load on the server. However, HTTP/TCP use reveals further challenges as adaptation is on top of TCPs congestion control algorithm, which forms nested control loops. As the throughput of the TCP connection depends on both the link capacity and the amount of congestion, the throughput can vary significantly over time[28].

Video over IP is becoming more and more important as we move further into the twenty-first century. The Internet is still growing rapidly and more uses are being found for video users. These include real-time online visual assistance, video learning, live event streaming, smart HDTVs, mobile phones, gaming devices, computers and visual communication among others. As the content quality is improving to meet end-user demands the bandwidth requirement for such devices is rapidly increasing. With increasing bandwidth demands and profuse video content, it is becoming likely that two or more adaptive streaming players may have to share a network bottleneck. This will result in a competition for available bandwidth. Example scenarios where this can take place are, when a number of people in the same household view similar or different videos simultaneously. Here, the domestic broadband access link is the shared bottleneck. Another instance of such competition is when many users watch the same live event (such as World Cup Soccer) online. The shared bottleneck may be an edge network link in this scenario. It has been previously observed that such competition can lead to performance issues[5], [15], [9].

In the presence of competing HTTP-based adaptive streaming (HAS) clients the TCP throughput does not always faithfully represent the fair-share bandwidth [21]. Three performance issues that can take place when two or more adaptive streaming players share a network bottleneck and compete for available bandwidth are instability, unfairness and utilization[16]. It is shown that in the case of two competing video flows FESTIVE [9]and ELASTIC [5]provide a received video rate that oscillates around the fair share, but with an increased number of video level switches. Depending on the temporal overlap of the ON-OFF[5]periods among competing players, they may not estimate their fair share correctly. In the case where both players overestimate their fair share, they may request a video representation with a higher bitrate than the fair share, which causes network congestion. Consequently, the players measure that their TCP throughput is lower than their previous fair share estimate, and so switch down to a lower video bitrate level. This creates a repeating oscillatory scenario, so inducing instability. A scenario can also occur where some players are requesting chunks with lower bitrates than the other players. This can occur as some players observe a throughput lower than the fair share, while others observe a throughput that is more than the fair share. This means that some players overestimate its fair share. When some players overestimate their fair share, it can be that the system of players converge to a stable equilibrium, but unfair. This occurs as the players with the larger fair share estimates request higher bitrate video levels. Even in the case where two players estimate their fair share correctly, bandwidth underutilization can still be prevalent. This occurs as both players request the same lower video bitrate level, which causes underutilization, even though stability and fairness still exist. In reality, several other factors can play an important role in the appearance and extent of instability, unfairness and underutilization, such as the exact player adaptation algorithm, TCP dynamics, bandwidth fluctuations, and the variability of the video encoding rate[26]. We group these problems into three categories: The first relates to the stability of the players in terms of requested bitrates and video quality. The second is the unfairness among competing players. The third is the potential bandwidth underutilization when multiple adaptive players compete.

In this paper we describe a novel sharing metric which adequately quantifies fairness in adaptive video streaming scenarios with multiple users. The metric will work if either similar or different videos are being viewed by the users. It is tested with the Conventional, ELASTIC, PANDA, and SHARE[11] client-based adaptive streaming algorithm.

This work consists of four sections. Section II discusses and extends the concept of network fairness to adaptive video streaming. Section III explores some fair share measures that may be used by adaptive controllers. Section IV reviews some of the state-of-the-art adaptive video players. Section V shows the results of experiments with three players: (1) FESTIVE, (2) PANDA and (3) ELASTIC. Finally, the conclusion is given in section VI.

II. NETWORK FAIRNESS

Given three video represented by vectors x, y, and z, where $x_{i,q1}$, $y_{j,q2}$ and $z_{k,q3}$ are q_i quality levels (q1=1, 2, … , n; q2=1, 2, … , m; q3=1, 2, … , p) allocated to user i, j, and k, how fair is it? Consider two allocations (Mbps) among three users sharing a 5Mbps bottleneck: x = [2; 2; 1] and y = [3; 1; 1]. In adaptive streaming scenarios with multiple users what would qualify a quantitative metric of fairness?

Various fairness measures have been proposed throughout the years. These range from simple ones, e.g., the ratio between the smallest and the largest entries of x, to more sophisticated functions, e.g., Jain's index and the entropy function. Some of these fairness measures map x to a normalized range between 0 and 1, where 0 denotes the minimum fairness, 1 denotes the maximum fairness (often corresponding to an x where all x_i are the same), and a larger value indicates more fairness. For example, min-max ratio is given by the maximum ratio of any two user's resource allocation, while Jain's index computes a normalized square mean. The research question explored

Volume: 09 Issue: 06 Pages: 3607-3615 (2018) ISSN: 0975-0290

in this work is: In adaptive streaming scenarios with multiple users what measures of fairness might be useful? When a user looks at a streaming video the quality level is important. In DASH users may be watching different videos and each video will have different quality levels. This means that in a multi-user DASH environment each user may require different quality levels to satisfy their QoE requirements. Thus, a user viewing a video with lower quality levels will require less resources than one who demands high quality levels. In this case the fairness is not adequately quantified by equal sharing of resources. Though this may occur if users are viewing the same video.

In DASH-based adaptive video multi-streaming scenarios the fairness of a player can be used as a parameter which determines the next segment selection, for example, Jain fairness index. A player can optimize on the fairness metric during streaming or use the metric after the streaming to simply find out the fairness allocated to the player during streaming.

Due to the widespread use of home routers in recent years, we focus on adaptive video streaming players at a single bottleneck link. Players compete at this bottleneck link for video data. Hence, in a scenario where players download different videos it is very important for each player to get a fair share of network resources. TCP is now the favored transport layer protocol for adaptive video streaming. TCP contains its' own rules, which adaptive players have to cope with.

One of the main objectives of TCP is to control the congestionin the Internet [11]. This control is not efficientif it does not ensure a fair sharing of network resources[2].A major problem of TCP is its bias against connectionswith long round-trip times (RTT) [3, 9, 12]. These connectionsare not able to achieve the same throughputas the other connections sharing the same path andhaving a smaller RTT. This is caused by the windowincrease algorithm adopted by TCP. Indeed, TCP usesan additive-increase multiplicative-decrease strategy forcongestion control [11, 17]. It is known that such kindof strategies leads to fairness when all the connectionsincrease their rates at the same rate [6]. In case of TCP and in presence of connections of different RTT, a fairness cannot be ensured since the window increase rate is inversely proportional to RTT (one packet per RTT in the congestion avoidance mode [17]) leading to an increase in the transmission rate at a rate inversely proportional to RTT.

Note that the transmission rate of a window-based protocol as TCP can be approximated at any moment by the window size divided by RTT. The connections with small RTT increase quickly their windows and grab most of the available bandwidth.

III. FAIR SHARE MEASURES

A. Interval measurement fairness

An interval measure takes into considerationthe idea of units. Thus, we can make absolute, rather than simplecomparative statements about the similarity or differencebetween measurements.Hence, we can state the number of units by which observations aremeasured to be different. In addition, meaningful comparisons can be made, such as greater than or less than.

B. Harmonic mean

The Harmonic Mean is used with inverse relationships. For example, speed and time are inversely related: for a fixed distance, increasing the speed results in a quicker journey time and vice versa. Suppose we have an out and back journey of 100 km each way with the speed 25 kph out and 50 kph back (think peak hour / non-peak hour, a cyclist cycling into wind and then with the wind, a vessel sailing against the current then with the current). The outward journey takes 100 divided by 25 = 4 hours and the return only 100 divided by 50 = 2 hours. The total distance is 200km in 6 hours, giving an average speed of 200 divided by 6 = 33.3 kph. This is the Harmonic Mean (of two numbers is twice the multiplication of the two numbers divided by the addition of the two numbers) and can be calculated from as 2×25×50 divided by 25+50 = 2500 divided by 75 = 33.3 kph. (It is necessary for the numerators in the inverse relationship, here 100km, to be the same).

C. Arithmetic mean

The Arithmetic Mean is commonly referred to as the average and has many applications, for example, the average exam mark for a group of students, the average maximum temperature in a calendar month, the average number of calls to a call center between 8am and 9am [27]. To get the arithmetic mean we add up the numbers and divide by how many numbers we have. Another term we can use is balance point. This means the sum of the differences between A and all the numbers greater than A equals the sum of the differences between A and all the numbers less than A. For example, suppose we have the numbers 2, 5, 10 and 19 for which A = 9. The differences between the numbers and 9 are 7, 4, 1 and 10. The sum of the differences to the numbers less than 9 is 7 + 4 = 11 and the sum of the differences to the numbers greater than 9 is 1 + 10 = 11.

D.Max-Min Fairness

In communication networks with the division of scarce resources, max-min fairness is said to be achieved by an allocation if and only if the allocation is feasible and an attempt to increase the allocation of any participant necessarily results in the decrease in the allocation of some other participant with an equal or smaller allocation. We are talkinghere about a fairness in the sharing of the bandwidthof the bottleneck link regardless of the volume of resourcesconsumed by a connection on the other linksof the network. A max-min fair allocationgives the most poorly treated user (i.e., the user who receivesthe lowest rate) the largest possible share, while not wasting anynetwork resources.

E. Weighted Max-Min Fairness

When users (applications) have different service requirements, then the network may not want to share bandwidth equally among users. Instead, the network could assign weights (priorities) to users and allocate bandwidth accordingly [24].

F. Proportional Fairness

A compromise-based scheduling algorithm based upon maintaining a balance between two competing interests: Trying to maximize total throughput while allowing all users at least a minimal level of service.

IV. LITERATURE REVIEW

The literature review is divided into the three most popular methods for video adaptive streaming, proxy-based, server-based and client-based. It has been shown that today's adaptive streaming techniques underperform when multiple clients consume video at the same time, due to fairness issues among clients. Concretely, this means that different clients negatively influence each other as they compete for shared network resources. FINEAS (Fair In-Network Enhanced Adaptive Streaming) is proposed[22]which is capable of increasing clients' Quality of Experience (QoE) and achieving fairness in a multi-client setting. A key element of their approach is an in-network system of coordination proxies in charge of facilitating fair resource sharing among clients. They claim that fairness is achieved without explicit communication among clients. In addition viewers using HTTP Adaptive Streaming (HAS) without sufficient bandwidth undergo frequent quality switches that hinder their watching experience. This situation, known as instability, is produced when HAS players are unable to accurately estimate the available bandwidth. Moreover, when several players stream over a bottleneck link, their individual adaptation techniques may result in an unfair share of the channel. These are two detrimental issues in HAS technology, which is otherwise very attractive. The authors [23] describe an implementation in the form of an HTTP proxy server and show that both stability and fairness are strongly improved. In [6] several network-assisted streaming approaches which rely on active cooperation between video streaming applications and the network are explored. They use a Video Control Plane which enforces Video Quality Fairness among concurrent video flows generated by heterogeneous client devices. A max-min fairness optimization problem is solved at run-time. They compare two approaches to actuate the optimal solution in an SDN network: the first one allocating network bandwidth slices to video flows, the second one guiding video players in the video bitrate selection.

In [24] the bandwidth estimate generated at the server is used for server-side adaptive bit encoding of digital media streams. The server application measures the network bandwidth available to the individual client for TCP/IP downloads of media and accordingly adjusts stream bit rate and composition to allow the client to retrieve the media stream with sufficient time margin to minimize the occurrence of underflow of client playback buffers. The root cause of the instability problem is that, in Steady-State, a player goes through an ON-OFF activity pattern in which it overestimates the available bandwidth [24]. They propose a server-based traffic shaping procedure that can considerably lower such oscillations. Their procedure is only triggered when oscillations are identified, and so the shaping rate is dynamically adjusted. This ensures that the player receives the highest available video profile without being unstable. Using HTTP for video streaming significantly increases the request overhead due to the segmentation of the video content into HTTP resources [19]. This overhead becomes even more substantial when non-multiplexed video and audio segments are deployed. The authors investigate the request overhead problem by employing the server push technology in the new HTTP 2.0 protocol. They develop a set of push strategies that actively deliver video and audio content from the HTTP server without requiring a request for each individual segment.

Chunk scheduling with stateless bitrate selection causes feedback loops, bad bandwidth estimation, bitrate switches and unfair bitrate choices [9]. This paper, which portrays the FESTIVE control algorithm, confirms that numerous problems occur when multiple bitrate-adaptive players (adaptation over HTTP) share a bottleneck link [1]. It uncovers the fact that the feedback signals the player receives is not a true reflection of the network state because of overlaying the adaptation logic over several layers. HTTP-based video delivery issues are elucidated: (1) the granularity of the control decisions, (2) the timescales of adaptation, (3) the nature of feedback from the network and (4) the interactions with other independent control loops in lower layers of the networking stack. FESTIVE uses an abstract player state to analyze commercial players: (1) schedule a video chunk for download, (2) select bitrate for chunk, and (3) estimate bandwidth. It identifies root causes of undesirable interactions with abstract state player framework and saw the need to guide the tradeoffs between stability, fairness and efficiency. As a result, the authors created a robust video adaptation algorithm, which tried to achieve: (1) Fairness – equal allocation of network resources, (2) Efficiency – get highest bitrates for maximum user experience, and (3) Stability – avoid needless bitrate switches. The eventual contributions were a family of adaptation algorithms using the following approaches: (1) Randomized chunk scheduling: to avoid sync biases in network state sampling, (2) Stateful bitrate selection: to compensate between biased bitrate and estimated bandwidth interaction, (3) Delayed update: to account for stability and efficiency tradeoff, and (4) Bandwidth estimator: to increase robustness to outliers.

FESTIVE [9]utilizes a bandwidth estimator that uses the harmonic mean of download speed. This harmonic mean is calculated over last 20 chunks. Authors claim the harmonic mean is more appropriate when computing the average of rates which is the case with throughput

Volume: 09 Issue: 06 Pages: 3607-3615 (2018) ISSN: 0975-0290

estimation and it is also more robust to larger outliers. They show experimentally how the harmonic mean provides reliable bandwidth estimates on which future bitrate update decisions can be made.

The authors in[15], who proposed the PANDA algorithm, noted that since TCP throughput observed by a client would indicate the available network bandwidth, it could be used as a reliable reference for video bitrate selection. However, this is no longer true when HTTP Adaptive Streaming (HAS) [1] becomes a substantial fraction of the total network traffic or when multiple HAS clients compete at a network bottleneck. It was observed that the discrete nature of the video bitrates results in difficulty for a client to correctly perceive its fair-share bandwidth. Hence, this fundamental limitation would lead to video bitrate oscillation and other undesirable behaviors that negatively impact the video viewing experience. They offered a design at the application layer using a "probe and adapt" principle for video bitrate adaptation (where "probe" refers to trial increment of the data rate, instead of sending auxiliary piggybacking traffic), which is akin, but also orthogonal to the transport-layer TCP congestion control. The authors illustrate a four-step state for an HAS rate adaptation algorithm: (1) Estimating: the algorithm starts by estimating the network bandwidth that can legitimately be used, (2) Smoothing: is then noise-filtered to yield the smoothed version, with the aim of removing outliers, (3) Quantizing: the continuous is then mapped to the discrete video bitrate, possibly with the help of side information such as client buffer size etc, and (4) Scheduling: the algorithm selects the target interval until the next download request. The advantages of PANDA are as follows. Firstly, as the bandwidth estimation by probing is quite accurate, one does not need to apply strong smoothing. Secondly, since after a bandwidth drop, the video bitrate reduction is made proportional to the TCP throughput reduction, PANDA is very sensitive to bandwidth drops.

In step (1) of PANDA's rate adaptation algorithm, instead of directly tuning the video bitrate, the algorithm probes the bandwidth based on the average data rate, which in turn determines the selected video bitrate and the fine-granularity inter-request time. In other words, by probing, PANDA determines a target average data rate. This average data rate is subsequently used to determine the video bitrate. The target average data rate estimated by the probing mechanism is neither biased nor have a large variation, enabling the subsequent operations to easily pick a video bitrate without sacrificing responsiveness when network bandwidth changes.

ELASTIC [5]proposes an approach that uses one controller to throttles the video level (t). This drives the playout buffer length (t) to a set-point q_T, which eliminates the ON-OFF traffic pattern. The player is always in ON phase unless (t) is the highest level and q $>Q_{max}$ ($>q_T$). The basic concept is based on the playout buffer state, design a feedback control system that computes l(t) to steer q(t) to a threshold q_T. The received rate r(t), is considered as a (measurable) disturbance since it cannot be manipulated. ELASTIC provides a received video rate that oscillates around the fair share, with an increased number of video level switches. However, the main result involved long-lived TCP flows [19][3], where experimental evaluation showed that ELASTIC can get the fair share when competing with TCP long-lived flows.ELASTIC estimates the received rate by passing the last segment download rate through a harmonic filter over the last download 5 samples.

V. EXPERIMENTS AND RESULTS

A. TAPAS

In communication TAPAS is an open-source Tool for rApid Prototyping of Adaptive Streaming control approaches, De Cicco et al. (2014). This flexible and extensible video streaming player tool is written in python. It allows researchers to easily design and carry out experimental performance evaluations of adaptive streaming controllers. There is no need to write code for the download of video segments, to parse manifest files, and decode video. TAPAS's design minimizes CPU and memory footprint. Thus, experiments involving many concurrent video flows can be carried out. Table 1 shows the main variables kept during streaming with a brief explanation of each one.

TABLE I. TAPAS DATA LOGS

Variable	Explanation
$q(t)$	the amount of data (in bytes) in the playout buffer
$b(t)$	the amount of data (in seconds) in the playout buffer
$B(t)$	last estimated available bandwidth in B/s
$L_r(t)$	current video quality level rate in B/s
$c_q(t)$	index of current level starting from 0 for the lowest video quality level
$c_{qmax}(t)$	index of maximum level starting from 0 for the lowest video quality level
$m_p(t)$	when the media engine is 'onPlaying' or 'onPaused'
$p(t)$	time spent on pause
$D(t)$	total downloaded bytes in B
$s_p(t)$	timestamp when starts the download of the last segment
$s_q(t)$	timestamp when stops the download of the last segment

B. Experimental Setup

The following experiments use the conventional DASH controller and measures five DASH-based performance metrics. All experiments are hosted on a Windows 10 machine, with the following specifications: Intel(R) Core(TM) i7-5500U CPU 2.40GHz processor, 16.00 GBphysical memory and an Intel(R) HD Graphics processor. A virtual networkis setup on the Windows 10 machine for the emulation test bed. Our setupconsists of a

Volume: 09 Issue: 06 Pages: 3607-3615 (2018) ISSN: 0975-0290

video server running Ubuntu (HTTP server), a router running FreeBSD (Home Router) and two real network players (HAS client) hosted on Ubuntu, cf. Figure 1.

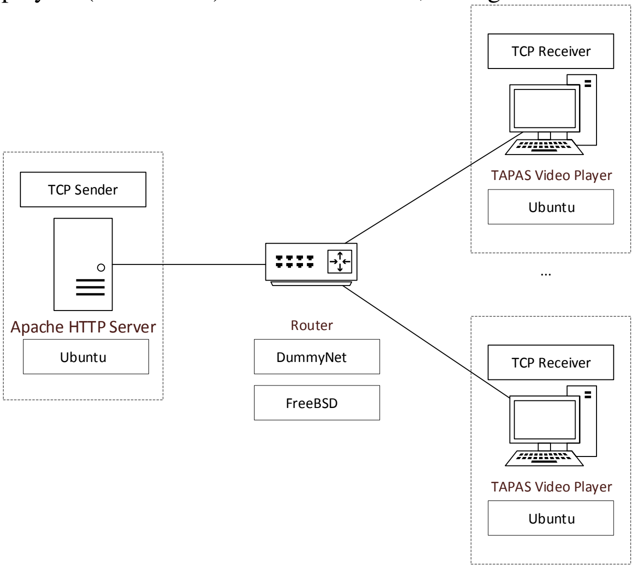

Fig. 1. Experiment testbed setup.

The 647 seconds long MPEG-DASH video sequence Elephant 's Dream4 is stored at the server. The video sequence is pre-encoded at eight different bitrates, ranging from 46 kbps to 4200 kbps. Further, it is divided into 2 second segments and exists in five different screen resolutions, ranging from 320x240 to 1920x1080. This is shown in Table II. The media type for the video is MP4. The video is encoded at 24 frames per second (fps) using the AVC1 (version 42c032) codec5. The Media Presentation Description (MPD) files are generated with GPAC version 0.5.1-DEV-rev53796.

TABLE II. VIDEO LEVELS, BITRATES AND RESOLUTIONS

Video level	Bitrate (kbps)	Resolution
1	46.0	320x240
2	131.0	320x240
3	222.0	480x360
4	328.0	480x360
5	523.0	854x480
6	796.0	1280x720
7	1200.0	1280x720
8	2100.0	1920x1080

We set up two players per experiment. The first experiment test fairness of an adaptive agent using the arithmeticmean (PANDA), and harmonic mean (ELASTIC and FESTIVE).

C. Results

The results are shown on Figures 1, 2, 3, 4, 5 and 6. FESTIVE does well. It is able to better than PANDA and ELASTIC. ELASTIC comes in last of the three approaches compared. Adaptive players using this technique are unable to get fair share of quality many times during the experiment. This can be attributed to the harmonic mean being taken over a 5 segment download sample rate interval compared to the 20 segment download sample rate of FESTIVE. The 20 segment download sample rate of FESTIVE gives a better accuracy in next segment selection []. The result here shows that the average data rate utilized by PANDA gives better performance than the 5 sample harmonic mean utilized by ELASTIC, but it is not good enough to better FESTIVE.Finally, the Conventional players, which is used as a benchmark does the worst.

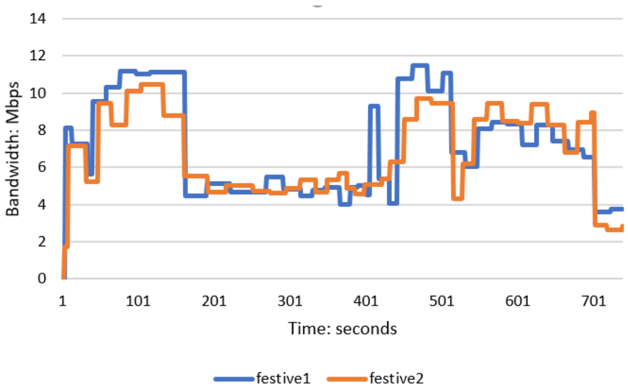

Fig. 1. Quality levels for two competing adaptive FESTIVE players sharing a bottleneck link.

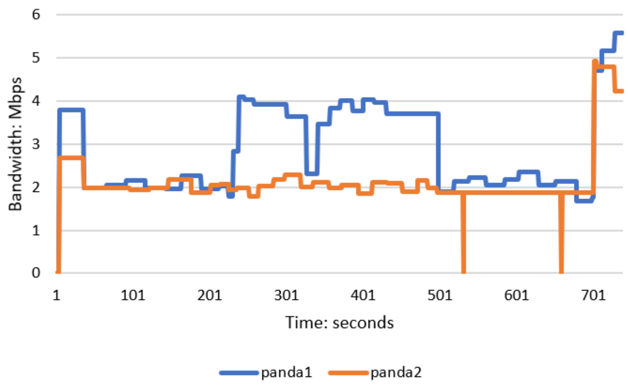

Fig. 2. Quality levels for two competing adaptive PANDA players sharing a bottleneck link.

Volume: 09 Issue: 06 Pages: 3607-3615 (2018) ISSN: 0975-0290

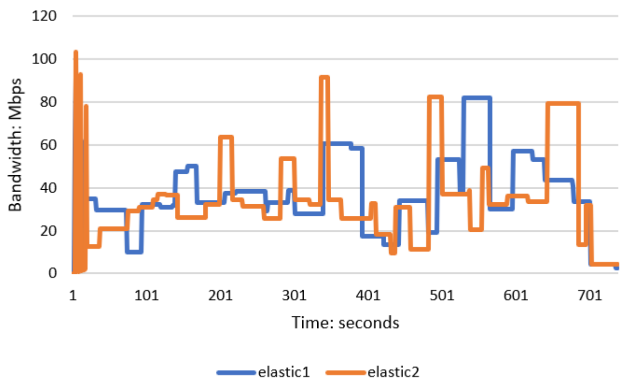

Fig. 3. Quality levels for two competing adaptive ELASTIC players sharing a bottleneck link.

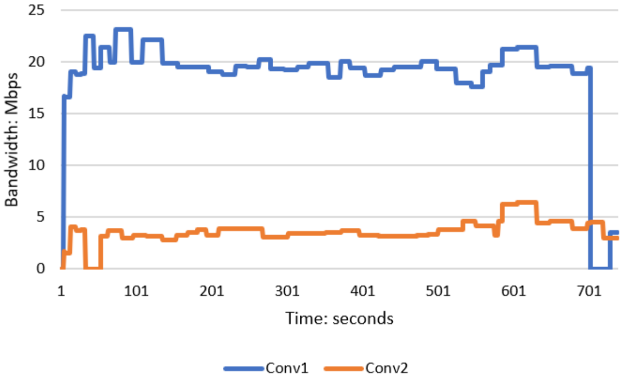

Fig.4. Quality levels for two competing adaptive Conventional players sharing a bottleneck link.

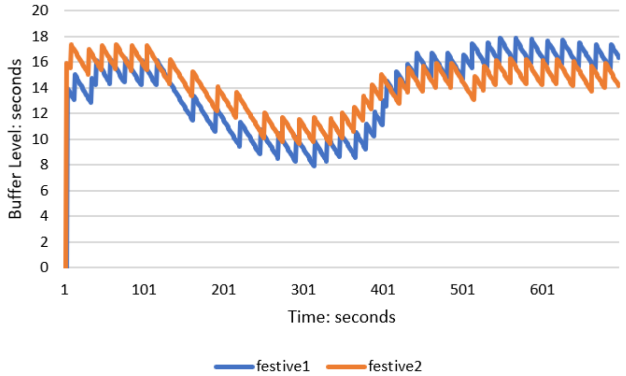

Fig. 5. Buffer levels for two competing adaptive FESTIVE players sharing a bottleneck link.

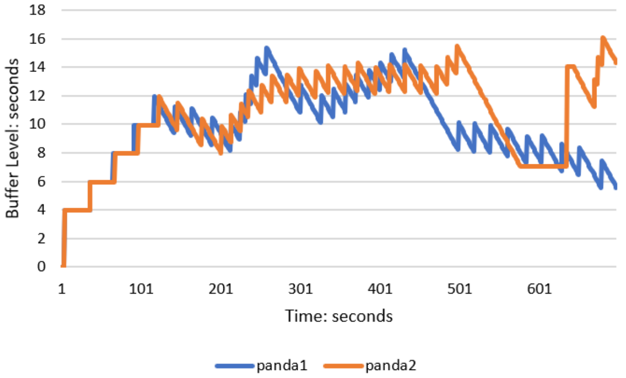

Fig. 6. Buffer levels for two competing adaptive PANDA players sharing a bottleneck link.

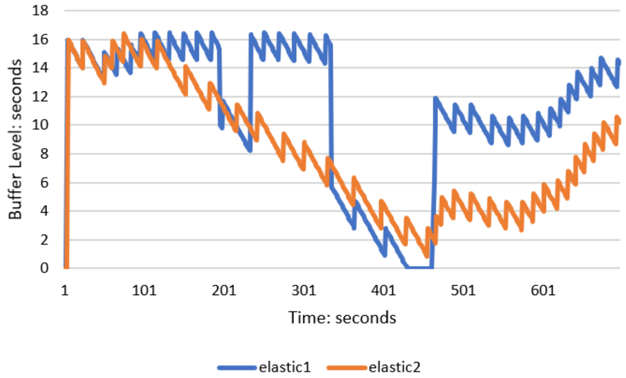

Fig. 7. Buffer levels for two competing adaptive ELASTIC players sharing a bottleneck link

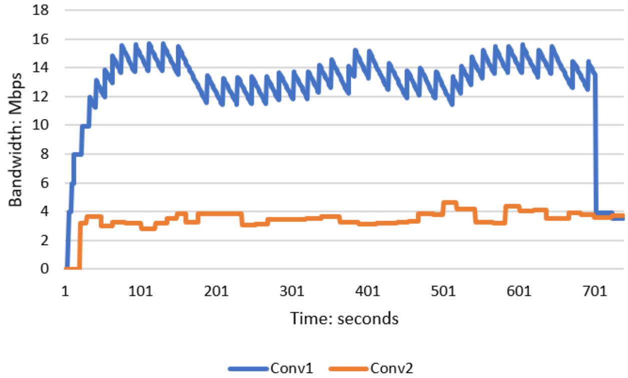

Fig.8. Buffer levels for two competing adaptive Conventional players sharing a bottleneck link

VI. CONCLUSION

This paper demonstrates the need for better bandwidth estimation techniques for adaptive video streaming players. Current state-of-the-art players are unable to accurately predict bandwidth, and this results in inadequate fair sharing among players. Of the three players tested, FESTIVE performs best, with ELASTIC the worst. However, PANDA is able to obtain better estimates when compared to ELASTIC. Future work can involve using scales among players that takes into consideration relative fair share, such as Simple Additive Weighting Method (SAW).

REFERENCES

[1] Akhshabi, Saamer, Lakshmi Anantakrishnan, Ali C. Begen, and Constantine 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, pp. 9-14. ACM, 2012.

[2] Altman, Eitan, Chadi Barakat, Emmanuel Laborde, Patrick Brown, and Denis Collange. "Fairness analysis of TCP/IP." In Decision and Control, 2000. Proceedings of the 39th IEEE Conference on, vol. 1, pp. 61-66. IEEE, 2000.

Volume: 09 Issue: 06 Pages: 3607-3615 (2018) ISSN: 0975-0290

[3] Alvarez-Horine, Rafael, and Melody Moh. "Experimental evaluation of Linux TCP for adaptive video streaming over the cloud." In Globecom Workshops (GC Wkshps), 2012 IEEE, pp. 747-752. IEEE, 2012.

[4] Claeys, Maxim, Steven Latre, Jeroen Famaey, and Filip De Turck. "Design and evaluation of a self-learning HTTP adaptive video streaming client." IEEE communications letters18, no. 4 (2014): 716-719.

[5] De Cicco, Luca, Vito Caldaralo, Vittorio Palmisano, and Saverio Mascolo. "Elastic: a client-side controller for dynamic adaptive streaming over http (dash)." In Packet Video Workshop (PV), 2013 20th International, pp. 1-8. IEEE, 2013.

[6] Georgopoulos, Panagiotis, Yehia Elkhatib, Matthew Broadbent, Mu Mu, and Nicholas 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, pp. 15-20. ACM, 2013.

[7] Goel, Ashvin, Charles Krasic, Kang Li, and Jonathan Walpole. "Supporting low latency TCP-based media streams." In Quality of Service, 2002. Tenth IEEE International Workshop on, pp. 193-203. IEEE, 2002.

[8] Huang, Weiwei, Yipeng Zhou, Xueyan Xie, Di Wu, Min Chen, and Edith Ngai. "Buffer State is Enough: Simplifying the Design of QoE-Aware HTTP Adaptive Video Streaming." IEEE Transactions on Broadcasting (2018).

[9] Jiang, Junchen, Vyas Sekar, and Hui Zhang. "Improving fairness, efficiency, and stability in http-based adaptive videostreaming with festive." IEEE/ACM Transactions on Networking (TON) 22, no. 1 (2014): 326-340.

[10] Kesavan, Selvaraj, and J. Jayakumar. "Network performance analysis of cloud based multimedia streaming service." International Journal of New Computer Architectures and their Applications (IJNCAA) 4, no. 3 (2014): 156-166.

[11] Koffka Khan, Wayne Goodridge, “Markov Decision Processes for bitrate harmony in adaptive video streaming”, 2017 Future Technologies Conference (FTC), Vancouver, Canada, unpublished.

[12] Kong, Fanxin, Xingjian Lu, Mingyuan Xia, Xue Liu, and Haibing Guan. "Distributed optimal datacenter bandwidth allocation for dynamic adaptive video streaming." In Proceedings of the 23rd ACM international conference on Multimedia, pp. 531-540. ACM, 2015.

[13] Kua, Jonathan, Grenville Armitage, and Philip Branch. "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP." IEEE Communications Surveys & Tutorials 19, no. 3 (2017): 1842-1866.

[14] Li, Mingfu, Chien-Lin Yeh, and Shao-Yu Lu. "Real-time QoE monitoring system for video streaming services with adaptive media playout." International Journal of Digital Multimedia Broadcasting 2018 (2018).

[15] Li, Zhi, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C. Begen, and David Oran. "Probe and adapt: Rate adaptation for HTTP video streaming at scale." IEEE Journal on Selected Areas in Communications 32, no. 4 (2014): 719-733.

[16] Mansy, Ahmed, Marwan Fayed, and Mostafa Ammar. "Network-layer fairness for adaptive video streams." In IFIP Networking Conference (IFIP Networking), 2015, pp. 1-9. IEEE, 2015.

[17] Mori, Shungo, and Masaki Bandai. "QoE-aware quality selection method for adaptive video streaming with scalable video coding." In Consumer Electronics (ICCE), 2018 IEEE International Conference on, pp. 1-4. IEEE, 2018.

[18] Mu, Mu, Matthew Broadbent, Arsham Farshad, Nicholas Hart, David Hutchison, Qiang Ni, and Nicholas Race. "A scalable user fairness model for adaptive video streaming over SDN-assisted future networks." IEEE Journal on Selected Areas in Communications 34, no. 8 (2016): 2168-2184.

[19] Oyman, Ozgur, and Sarabjot Singh. "Quality of experience for HTTP adaptive streaming services." IEEE Communications Magazine 50, no. 4 (2012).

[20] Pantos, Roger, and William May. "HTTP live streaming." (2017).

[21] Petrangeli, Stefano, Jeroen Famaey, Maxim Claeys, Steven Latré, and Filip De Turck. "QoE-driven rate adaptation heuristic for fair adaptive video streaming." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 12, no. 2 (2016): 28.

[22] Petrangeli, Stefano, Jeroen Famaey, Maxim Claeys, Steven Latré, and Filip De Turck. "QoE-driven rate adaptation heuristic for fair adaptive video streaming." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 12, no. 2 (2016): 28.

[23] Ramakrishnan, Sangeeta, Xiaoqing Zhu, Frank Chan, and Kashyap Kambhatla. "SDN based QoE optimization for HTTP-based adaptive video

Volume: 09 Issue: 06 Pages: 3607-3615 (2018) ISSN: 0975-0290

streaming." In Multimedia (ISM), 2015 IEEE International Symposium on, pp. 120-123. IEEE, 2015.

[24] Schmidt, Mark S., Praveen N. Moorthy, Ajay Luthra, and Paul Moroney. "Server side adaptive bit rate control for HTTP streaming clients." U.S. Patent 9,769,235, issued September 19, 2017.

[25] Seufert, Michael, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hossfeld, and Phuoc Tran-Gia. "A survey on quality of experience of HTTP adaptive streaming." IEEE Communications Surveys & Tutorials 17, no. 1 (2015): 469-492.

[26] Tanwir, Savera, and Harry Perros. "Modeling live adaptive streaming over HTTP." Computer Communications 85 (2016): 74-88.

[27] Weisstein, Eric W. CRC concise encyclopedia of mathematics. CRC press, 2002.

[28] Xie, Xiufeng, Xinyu Zhang, Swarun Kumar, and Li Erran Li. "pistream: Physical layer informed adaptive video streaming over lte." In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, pp. 413-425. ACM, 2015.

[29] Ye, Zakaria, Rachid El-Azouzi, Tania Jimenez, Eitan Altman, and Stefan Valentin. "Backward-shifted strategies based on SVC for HTTP adaptive video streaming." In IFIP Networking Conference (IFIP Networking) and Workshops, 2016, pp. 494-502. IEEE, 2016.

[30] Yin, Xiaoqi, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. "A control-theoretic approach for dynamic adaptive video streaming over HTTP." In ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp. 325-338. ACM, 2015.

[31] Zambelli, A. "The microsoft ISS smooth streaming (MSS) client." (2015).

AUTHOR DETAILS

Koffka Khan received the M.Sc., and M.Phil. degrees from the University of the West Indies. He is currently a PhD student and has up-to-date, fpublished numerous papers in journals & proceedings of international repute. His research areas are computational intelligence, routing protocols, wireless communications, information security and adaptive streaming controllers.

Wayne Goodridge is a Lecturer in the Department of Computing and Information Technology, The University of the West Indies, St. Augustine. He did is PhD at Dalhousie University and his research interest includes computer communications and security.