QoE Evaluation of Legacy TCP Variants over 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

Even though there is considerable work in adaptive video streaming, video players still suffer drawbacks which include: inadequate fair share, poor bandwidth utilization, frequent level shifts and high re-buffering ratios. These draw- backs result in poor perceived video quality which degrades a viewers quality of experience (QoE). DASH approaches have been known to deliver higher QoE to viewers by improving the video players segment selection. However, the transport layer protocol is another important aspect of video streaming. Hypertext Transfer Protocol (HTTP) is the most used application layer protocol over the Internet. It utilizes Transmission Control Protocol (TCP) as its transport layer protocol. TCP variants use different mechanisms for congestion control. The DASH standard uses HTTP, thus TCP variant selection becomes very important for effective video streaming. In this paper we test the performance of four linux-based TCP variants using the Conventional, PANDA and ELASTIC client-side DASH players in congested bottleneck link conditions. These experiments illustrate the varying impact on viewer QoE when using these various TCP congestion control mechanisms. The importance of TCP variant selection is exemplified as we observe that Westwood+ and YeAH are the most promising variants.

Keywords -QoE, DASH, HTTP, TCP, congestion, PANDA, ELASTIC, Westwood, YeAH.

Date of Submission: Apr 04, 2021

Date of Acceptance: Apr 19, 2021

I. INTRODUCTION

 $C_{\text{omputing devices capable of displaying high definition}}$ video have become common place, and high-speed wireless networks are available in most populated areas in some developing and developed countries. An important application of the technologies is video streaming at home or coop-erate settings. Consequently, the number of video streaming providers targeting the video streaming market has exploded. The issue of fair treatment of services when there is a shortage of a certain resource is relevant in many contexts, for example, bandwidth. DASH players would want to reach their maximum quality level and to remain at this level. However, when players are sharing a bandwidth constrained bottleneck link some players would potentially be forced to remain at one of their lower quality levels, while others use high quality levels. The competing environment increases the chances of players fluctuating between quality levels. This lowers the player stability (e.g. video stalling, skipping and switching increases).

On-demand adaptive video streaming over the Internet often suffers from adverse conditions, which may lead topo or segment selection by the client player. Video segment selection have a significant effect on the userperceived quality of experience (QoE). Another important aspect related to adaptive video streaming is the selection of an appropriate application and transport layer networking protocols. In some of the early work on video transport, protocols such as Rate Adaptation Protocol (RAP)[1] and TCP Friendly Rate Control (TFRC)[2] were defined on top of the transport layer that put the sender in charge of varying the sending rate (and consequently the video rate) based on feedback being received from either the network or the receiver, forming a combination of congestion control and flow control. RAP used a TCP-like additive in-crease/ multiplicative decrease (AIMD) scheme. TFRC used an additive increase/additive decrease (AIAD) scheme to adjust the server's sending rate by estimating the path's through put based on TCP square root formula using the path's Round Trip Time (RTT) and packet loss rate.

There were many deficiencies in the early network protocols implemented. This lead researchers to create and adopt new solutions to all eviate some of these existing problems. Hyper-text Transfer Protocol (HTTP) is the stateless protocol which is used in most of the modern video streaming applications. The HTTP [3] on top of Transmission Control Protocol (TCP) [4] has become the primary protocol for multimedia content delivery over the Internet, also widely known as over-the-top (OTT) or Internet Protocol (IP)-based content delivery. HTTP avoids NAT and firewall traversal issues and provides reliability and deployment simplicity because of the widely implemented and deployed underlying TCP/IP protocol. HTTP streaming uses a fast startup by downloading lowest quality and smallest segment first and adjusting its' rate afterwards. By using HTTP on top of TCP, Dynamic Adaptive Streaming over HTTP (DASH) yields the following benefits. Firstly, clients use the standard HTTP protocol which provides more ubiquitous reach as HTTP

traffic can traverse NAT sand fire-walls [5]. Secondly, DASH servers are regular commodity Web servers, which significantly reduces the operational costs and allow the deployment of caches to improve the performance and reduce the network load. Thirdly, a client requests each video chunk independently and maintains the playback session state, so servers do not need to track session state. Maintaining session state at the client means client scan retrieve video chunks from multiple servers with loadbalancing and fault tolerance between commodity HTTP servers [6]. Lastly, rely-ing on TCP reliability and interflow friendliness improves the likelihood that streaming traffic consumes only a fair fraction of the network bandwidth when sharing with other traffic.

In spite of these advantages there are different variations of TCP congestion control algorithms. Thus, their performance will vary depending on network and application factors during a client-server session. At present the performance of many variants has not been tested over DASH in bottleneck network conditions. This paper attempts to bridge this important gap which is missing from the literature. We consider the performance of four linux TCP variants [7]:Westwood+[8],Cubic, [9], YeAH [10] and Illinois [11]for client-side DASH players[12] at house hold bottleneck links [12] and [13]. Objectives: (1) First time the performance of these four TCP variants are studied under the same DASH test bed. (2) Subjective QoE metrics have never been used to evaluate various TCP variants in various video streaming environments. (3) Experimental work merging TCP design mechanisms and client-side DASH behaviour (Conventional, PANDA and ELASTIC) have never been attempted.

This paper consists of 5 sections. Section two describes the Conventional, PANDA and ELASTIC DASH models \cite{li2014probe}. The Conventional model is used as the benchmark player against which selected TCP variants are tested. Section three outlines the congestion avoidance algorithms of the TCP variants. In section four we give the experimental emulated testbed. Section five discusses the role of QoE in DASH and lists three important objective QoE metrics used in our experimental evaluations. This section also describes the subjective QoE evaluations using Mean Opinion Score (MOS) scales. In section six we list the results of our experiments presenting both objective and subjective findings. Finally, we conclude our work in section seven.

II. DASH MODELS

A. Conventional DASH model

In this model the video is pre-encoded and stored on the server at l video bitrates denoted by $R = \{R_1,...,R_l\}$. Each video stream is broken up into segments or chunks of seconds each. The streaming process for each client is divided into sequential segment downloading steps $n = \{1, 2, ..., N\}$. Variable durations between consecutive segment requests are incorporated in the Conventional model. At the beginning of downloading of each sequence

n, two important decisions are made: the video bit rate of the next segment to be downloaded, r[n];and, the target inter-request time $\hat{T}[n]$. The client also determines the time it takes to download the n^{th} segment and this is notated by $\tilde{T}[n]$. If the download duration $\tilde{T}[n]$ is shorter than the target delay $\hat{T}[n]$, the client experiences an off time or wait time of $\hat{T}[n] - \tilde{T}[n]$. Otherwise, the download starts immediately. The actual inter-request time T[n] is given as:

$$T[n] = \max(\hat{T}[n]; \tilde{T}[n])$$
(1)

The downloaded segments are queued on the video player's buffer and are de-queued during playout. Suppose B[n] represents the buffer duration (measured in time) at the end of segment n, then the B[n] can be formulated as:

 $B[n]=\max(0; B[n-1]+\tau-T[n])$ (2) The second aspect of the Conventional model is the rate adaptation approach which uses four objectives:

 Estimating the bandwidth x^[n] by equating it to the TCP throughput x̂[n] by equating it to the TCP throughput:

 $\hat{x}[n] = \tilde{x}[n-1]$ (3) where the TCP throughput, x~[n] during the segment download.

2) Smoothing x^[n] to give an estimated bandwidth y^[n] This smoothing involves using a smoothing function S(.) [15] which takes as it input m n values of x^[m]. Possible smoothing functions are sliding-window, mov-ing average, exponential weighted moving average and harmonic mean. Equation 4 summarizes the smoothing function.

$$\hat{y}[n] = S(\hat{x}[m]; m \le n) \tag{4}$$

- 3) The real value ŷ[n]is then mapped to the discrete video bitrate r[n]∈Rusing a quantization function Q(:) which accepts as input ŷ[n]. Some conventional approaches may use additional buffer information like buffer occupancy level. r[n]=Q(ŷ) (5)
- 4) Scheduling. The next download is then scheduled based on the size of the buffer and given by Equation 6.

$$\hat{T}[n] = \begin{cases} 0, \text{ if } B[n-1] < B \text{max} \\ \tau, \text{ otherwise} \end{cases}$$
(6)

where B_{max} is a threshold value for the play out buffer length and represents the length of bytes that must be buffered before playback begins.

The first output of the conventional approach is the bit rate level for the next download and the second output is the time to start the next download. The next segment download starts immediately after the current download finishes (buffering mode) or starts at the inter-request time, set to a fixed duration forcing OFF periods (steadystate mode). The main drawback of conventional approaches are that in the presence of competing DASH clients, the estimated bandwidth based on the observed TCP throughput during the ON-intervals does not represent the fair-share bandwidth. Possible use-cases that result from this improper bandwidth estimate are:

- 1) Where competing players overestimate their fair share they may request video representations with a higher bit rate than the fair share. This leads to network con-gestion. When TCP detects congestion, the players in turn estimate lower bandwidth than their previous fair share estimate and select a lower video bit rate level. This environment creates a repeating oscillatory scenario and results in instability.
- 2) Where some players overestimate their fair share while others underestimate their fair share. In this situation players may converge to a stable equilibrium, but with-out fair allocation of bandwidth.
- 3) Where players estimate their fair share correctly, yet the total bandwidth capacity of the network is not utilized. This occurs as players may be requesting sub-optimal video bit rate levels.

B. PANDA

A bottleneck link include residential, campus, and corpo-rate networks, along with publicly available hotspots. Players cannot correctly estimate their fair share when video flows compete for bottleneck bandwidth. This results in oscillating segment rate requests. Multiple Microsoft Smooth Streaming (MSS) players sharing a bottleneck experience synchronized quality switches [14]. This poor performance of MSS players are caused by the discrete nature of available video bitrates which cause inaccurate bandwidth estimates and overlapping ON-OFF periods. Underutilization of network bandwidth is a major outcome. User-QoE suffers. Thus, a "probe and adapt" strategy was proposed to give the player an accurate estimate for the measured segment throughput. TCP download through-put is used by PANDA as an input only if its assessment is a true guide of the fair-share bandwidth. PANDA tries to continually probe the link by incrementing the data rate during the OFF-intervals to efficiently utilize the bandwidth in the hope of improving its throughput estimate. By using conservative rate estimators over a smoothed average network bandwidth PANDA aims to avoid bandwidth overestimation issues that are caused by overlapping ON-OFF periods as it tries to repress high bitrate selection to maintain network health. This effectively diminishes the bitrate fluctuations in relatively unchanging network conditions. PANDA is conservative since it prefers to slightly underutilize the channel at the benefit of having a more constant bit rate selection. Moreover, PANDA does not always fully exploit the accessible bandwidth capacity. Another strategy of PANDA's heuristic rules includes design elements to counter stalls such as maintaining a high level of playout buffer occupancy to help improve quality.

C.ELASTIC

ELASTIC [16] manipulates a control technique known as feedback linearization which varies the video bitrate to control the playout buffer length. A harmonic mean rate estimator is used to circumvent outliers. The adaptation logic of ELASTIC integrates a proportional integral controller with the harmonic mean rate estimator to determine the quality level of the next video segment. However, harmonic mean estimates usually gives a conservative estimate which could result in bandwidth underestimation. ELASTIC targets fair share from the network's frame of reference among competing players. Full bandwidth utilization is one of ELASTIC's primary goals. To achieve this goal, it has a very aggressive bufferbased controller which tries to keep the buffer level close to a set-point. However, this is achieved at the price of more often quality level switches. Low QoE results. ELASTIC can overcome issues such as bandwidth underutilization and unfairness with greedy TCP flows. However, again the price is an increase of video level switches.

III. TCP VARIANTS

The slow start and congestion avoidance phases are common to all TCP congestion control algorithms. An increasing com-mand window (cwnd) of one maximum segment size (MSS) for each received acknowledgment (ACK) occurs during the slow start phase. As a result the cwnd value is doubled for each round trip time (RTT). The objective of this exponential increase is achieving high bitrates as quick as possible. The slow start threshold (ssthresh) defines the limit of the cwnd size in the slow start phase. Thus, when the cwnd size exceeds the ssthresh the congestion avoidance phase starts. In this phase the cwnd is also increased but slowly. The cwnd increases until a congestion event is occurs. When congestion occurs the cwnd window is reduced in size and then increased again until another congestion event occurs. The increase of cwnd during congestion avoidance and decrease following congestion detection is one way to distinguish among the various TCP congestion control algorithms.

The TCP variants [7] differs from each other as they are specially designed by considering the particular network con-ditions and scenarios but not considering the requirement of the application. Some attempts have even been made on multi-path TCP [17] and [18]. TCP variants are categorized based on their features, network environment, congestion detection (loss or/and delay), congestion avoidance method and associations between them. TCP variants (such as YeAH, Cubic, Illinois, and Westwood+) specific to high bandwidth-delay product networks achieve high bandwidth more quickly and seem to give better per-formance for DASH service than classical TCP variants, but the improvement is limited. In order to facilitate our study, we chose four well known congestion control variants, and we classify them according to the criteria specified in the previous paragraph.

A. Westwood+

Westwood+ [8] utilizes end-to-end available bandwidth es-timation along the TCP path [19]. A filter on the returning ACK packets enables this estimation value to be found. This estimate sets the control windows adaptively, for example, when network congestion occurs. The pseudo code of the Westwood+ algorithm is reported below:

1) On ACK reception:

cwnd is increased accordingly to the Reno algorithm; the end-to-end bandwidth estimate BWE is computed.

2) When 3 DUPACKs are received:

Set systhesh = $max(2, (BWE* RT T_{min}) / seg$ _ size); Set cwnd = systhesh;

3) When coarse timeout expires:

Set systhesh = $max(2,(BWE* RT T_{min}) / seg$ size); Set cwnd = 1;

In the case of receiving ACKs Westwood+ increases additively the cwnd. The slow start threshold (ssthresh) is set to the estimated bandwidth (EBW) times the minimum round trip time (RT T_{min}) in the case where three duplicate ACKs (DUPACKs) are received. Further, the congestion window (cwnd) is set to this same value. However, in the case of a course time expiring the cwnd is set to one. Thus, in the case of congestion Westwood+ adaptively sets its cwnd and ssthresh in an Additive-Increase/Adaptive-Decrease (AIAD) manner [14].

B. YeAH

YeAH [10] utilizes a "Fast" and "Slow" mode. In the "Fast" mode, YeAH increments the congestion window aggressively, for example, using the STCP rule. During the "Slow" mode, YeAH acts as TCP Reno as it increases the congestion window by one every RTT. When three duplicate ACKs are received (if a loss is detected) YeAH halves its cwnd. YeAH shows high efficiency and fairness but it still has the same problem of RTT estimation.

C. CUBIC

CUBIC cwnd size is a cubic function of time since the last congestion event, with the inflection point set to the

window size prior to the event []. The first portion of cwnd growth is concave. The cwnd size rapidly shifts up to the size measured at the last congestion event. The second portion is cwnd growth is convex. CUBIC searches for more bandwidth, slowly at first then very rapidly. CUBIC spends a lot of time at a plateau between the concave and convex growth region which allows the network to stabilize before CUBIC begins looking for more bandwidth.

CUBIC's [10] window size is dependent only on the last congestion event. Hence, CUBIC allows for more fairness between flows since the window growth is independent of RTT.

D. Illinois

Illinois [11] increases the window size W bya/W for each received ACK. It decreases $W by \beta W$ for each loss event. A and β are functions of average queuing delay. Thus, Illinois increases the throughput very quickly when congestion is far and increases the throughput very slowly when congestion is near. The window curve is concave, and the average throughput achieved is high as a result.

IV. EXPERIMENTAL TESTBED

A. Household viewer environments

1) Environment 1: All family members view videos: All five family members start to view their video simultaneously and continue for 10 min. This scenario illustrates how players compete with each TCP variant.

2) Environment 2: Increasing number of family members view videos: Family member one starts to view their video. Further each family member starts to view their video at 2 min intervals. This means the fifth family member will start at 8 min and all five will be viewing their video for the last 2 min. This scenario shows how the transition from one viewing family member to multiple occurs.

3) Environment 3: Decreasing number of family members view videos: Five family members start to view their video simultaneously, family member five stop after 2 min. Consequently, other family members stop watching their video at 2 min intervals. This means the last family member will watch their video alone for the last 2 min. This scenario shows how a transition from five family members to one takes place.

4) Environment 4: Family members compete with other network flows: Only one family member starts to view their video and continues for 10 min. At 30 s, we simulate a heavy congestion event with a provoked packet loss of 50 % of the received packets at the server over a 1-s period. This scenario shows the robustness of each combination against the congestions that are induced by

external factors, such as by other concurrent flows in the home network.

B. Household network

To imitate realistic viewing environments, we test video content in the household setting of the subjects. The findings obtained during this study offer insights into the viewer QoE under different video consumption behaviors of household family members.

TAPAS is an open-source tool for adaptive streaming approaches [20]. This software is written in the python program-ming language. TAPAS makes it easy to plan and tests involving streaming performance execute evaluations. There is code available to retrieve video segments, parse manifest files, and decode the video. The architecture of TAPAS minimizes the memory footprint such that tests can be done for a large number of concurrent video flows. TAPAS is a full player that decodes and renders the raw video on the screen. The HTTP Live Streaming (HLS) and Dynamic Adaptive Streaming over HTTP (DASH) specifications are supported by TAPAS.

A video server running Ubuntu, a household router and two or three Ubuntu TAPAS plays are part of our system as shown in Figure 1. For both client and server computers, Ubuntu Linux release 15.04 is used. The server software is Apache HTTP version 2.4.9. For the tests, all communication between the client and the server passes via the household router with a fixed network bandwidth (20 Mbps).

The 10-minute long Elephant's Dream MPEG-DASH video sequence is saved on the server. Seven separate bitrates,



Fig.1. Household network setup

¹http://www.free-codecs.com/download/x264`vfw.htm ²http://mirrors.vbi.vt.edu/linux/opensuse/repositories/pack man/11.0/x86`64/ ³http://www-itec.uni-

klu.ac.at/ftp/datasets/DASHDataset2014/ElephantsDream

TABLE I

VIDEO ATTRIBUTES

Video Level	Bitrate (kbps)	Resolution
l ₀	328.0	320x240
l_1	796.0	480x360
l_2	1500.0	480x360
l ₃	2400.0	1280x720
l_4	3000.0	1280x720
l ₅	3800.0	1920x1080
l ₆	4200.0	1920x1080

ranging from 328 kbps to 4200 kbps, pre-encode the video series. It is further broken down into 2 second parts and is available in five different screen resolutions, ranging from 480x360 to 1920x1080. This is shown in Table I. For the video, the media type is MP4. The video is encoded using the AVC1 (version 42c032) codec at 24 frames per second (fps)¹. GPAC version 0.5.1-DEV-rev53799 is the created MPD format². Videos of Big Buck Bunny, Tears of Steel, Sintel, and Of Forest and Men were treated similarly and put on the server, see footnote³ for details. Thus, a separate video was watched by each family member. We used the conventional, PANDA and ELASTIC client-side DASH approaches to carry out the experimental evaluation. Since we were evaluating legacy TCP variants we selected adaptive streaming video players that were created around the same time. This encouraged research involving timebased technological advances. However, our companion paper tests more recent transport layer network protocols with current adaptive video players [21].

V. QOE AND DASH

The video distribution environment is a constantly shifting world with current demands for the creation of immersive, personal and socially linked networks [22]. The Internet is a ubiquitous network of high-speed networking that facilitates this evolution. However, no quality of service (QoS) [23] or quality of experience (QoE) [24] and [25] assurances are offered on the Internet. Applications thus dynamically adjust their specifications to the QoS or QoE level required. Network infrastructure assessment scales are characterized by measured quantitative performance of technological device actors and subjective research of individuals. Scales for userperceived QoS are subjective. They receive opinions from participants who rate the consistency or improvement of an actor's experience. The benefits of this approach are (1)it is very user-centered, (2) well defined evaluations, (3) it includes common repeatable elements, and (4) it can be readily contrasted with the performance of the technological frame-work. Disadvantages, though, such as potential deficiencies in actual human perceptual features (e.g. ear infection during testing) and latent psychological influences [26], can make this approach challenging.

As a multidisciplinary area, Quality of Experience (QoE) rapidly emerges. Social psychology, brain science, economics and engineering science was formulated by QoE. It reflects on understanding the general criteria of human quality, [27]. Human aspirations and quality requirements are met by the design of QoE. QoE is the conventional equivalent of (QoS). QoS criteria guarantee a certain level of performance for end users. Research in [28] defines Quality of Experience (QoE) as "the overall acceptability of an application or service, as perceived subjectively by the end user." QoE combines individuals' aesthetic and parsimonious desires. These scales are absent from QoS.

In video streaming environments QoE collects measures from end viewers, while QoS measures technological trends that influence the performance of the device. Subjective QoE stretches beyond the content of viewerperceived quality. It tests the satisfaction levels of viewers. Subjective viewer perception relies mostly on questionnaires and scales of evalu-ations [29]. Viewer conduct based entirely on opinion, though, is not consistent. Objective steps typically concern the device and not the end viewers. These objective measurements are collected and monitored by QoS instruments. In a network, for instance, it is possible to monitor the buffer levels. These indirect technological viewer measures are inferred as OoE. Consequently, the association between technological measures and viewer behaviour is therefore confirmed by these viewer measures.

A. Objective QoE

Computational media quality models are objective when they model observable technical parameters [30]. They to accumulated improvements have access in technological criteria, including those that impact quality. However, the value of these models is diminished since, with the addition of new parameters, they must be constantly checked against new viewer test results.QoE metrics apply to viewer performance dependent on real use. For laboratory and field experiments or diverse types of facilities and customer cases, there are several ways to collect objective QoE. A central argument applies to objective QoE indicators for viewer perception and beyond [31]. This brings QoE into the viewer experience realm. However, the subjective viewer measure remains a significant element in the QoE analysis. Thus, the dynamic essence of OoE is best expressed by a mixture of quantitative and subjective QoE measurements.

TCP efficiency declines with mishaps in the network route, for example, packet errors and reordering, within the scope of a DASH streaming session. Video playback waits for new video data under those situations. This affects user-QoE badly. Many other variables influence user-QoE, such as display consistency and replay smoothness [32]. QoE is represented by a Mean Opinion Score (MOS) of 1 ('Bad') to 5 ('Excellent') [33].Mean opinion ratings are accumulated through subjective or objective tests. For HTTP video streaming, however, quantitative measurements such as Peak-Signal-to-Noise-Ratio (PSNR) [34] and Mean Square Error (MSE) [35] are not appropriate. They analyze spatial video quality only. In improving the delivery of DASH services, the development of QoE measurement methodologies, performance measurements and reporting protocols play a key role [36].

Objective OoE performance metrics: From a human perspective quality of experience (QoE), playback smoothness and video fidelity are inevitably linked. For understanding the perceived user-QoE, QoE metrics provide several dimensions [37]. A case can arise during streaming, where the video streaming program uses more data than incoming video bi-trates. The buffer for playouts gets less. However, the video inevitably stops allowing playback to resume because there is inadequate data in the playout buffer. Stalling prevents video replay by under running or being low/empty of the required video segments [38]. Re-buffering is driven by the duration of an interruption. Viewers face longer durations of stalling, and significant buffering of playtime. When compared to a streaming session with repeated brief freezes, viewers choose a scenario with a single long freeze [39]. Researchers observe a decline in average video output as the length of the impairment increases [40]. In comparison, video stalls are greater than a drop in frame rate. They observe that video stalls are weaker at odd intervals than those at normal intervals. Bandwidth variations can cause video playback interruptions. Packet loss due to buffer overruns (buffer is full, so packet reception results in drops occurring) leads to certain packets being retransmitted and consequent delays. Playout interruptions disturb viewers and delays, such as flashing, generate negative results.Researchers have also observed that switching to an intermediate rate is favored over several broad magnitude rate switches before switching to a higher rate, and a continuous rate is preferred over switches if the continuous rate is higher than the base rate [41]. These occurrences should be taken into account in the calculation of QoE. We selected three typical target QoE metrics used in the adaptive video streaming literature for these purposes: (i) Rebuffering ratio [42], (ii) Stability [43] and [44], and (iii) Bandwidth utilization [45]. For these measurements, we are now look at their formal descriptions.

1) Rebuffering Ratio: If the downloading period is greater than the refilling of the buffer, playback must be paused to buffer further segments. The length of rebuffering is the cumulative time that playback has been paused. The playback time is the real video play duration. The rebuffering ratio over the playback time is defined as the rebuffering duration. Essentially, when viewing a film, it is the fraction of time that a viewer encounters rebuffers. Let us say, for example, that rebuffing happens for thirty seconds when a viewer watches a video for ninety seconds. In this case, the rebuffering ratio will be 30/(30+90) = 0.25. In our experiments we use the term 'Buffer Level' to delineate an adaptive player's rebuffering ratio.

2) Stability: Stability is the fraction of a player's consecutive segment requests in which the demanded bitrate does not rise or fall. It is the ratio of successive consistent bitrate requests over the number of streamed segments. For every six segments in the video (12s for 2s video segment sizes), we quantify this. For the viewer, a higher stability measure implies a higher QoE. In our experiments we use the term 'Switches' to delineate an adaptive player's stability.

3) Network Bandwidth Utilization: If the end-to-end band-width is smaller than the requested video segment bitrate, buffer underflows and replay interruptions can be encountered by the viewer. Conversely, the viewer experiences sub-optimal video quality if the end-to-end bandwidth is higher than the requested video segment bitrate. In comparison, the key cause of network underutilization is conservative video bitrate selection in DASH side approaches Bandwidth client [46]. consumption is then the occupied portion of the bandwidth of the bottleneck connection assigned to stream video, i.e. the capacity segregated from background traffic. In our experi-ments we use the term 'Utilization' to delineate an adaptive player's network bandwidth utilization.

B. Subjective QoE tests

We performed subjective experiments to verify the efficiency of TCP algorithms. In the experiments, we used five videos (i.e., Elephant's Dream, Big Buck Bunny, Steel Tears, Sintel, and Of Forest and Men). Six versions of these videos were encoded at a rate of 350 Kbps, 750 Kbps, 1.5 Mbps, 2.5 Mbps, 3.5 Mbps, and 4.5 Mbps. Every version was further separated into 2-second video segments of equal length. All the approaches were checked against the same traces of bandwidth that were obtained from experiments undertaken, making for a fair contrast. Then, in a single video format, we merged the video fragments. For each rate adaptation method, this procedure was conducted. We asked 76 participants to engage in the subjective assessments in our experiments. All participants were not color blind and had natural vision clarity, and had no experience of the method of streaming used in the test. Twenty sequences were tested by participants. They were only asked to screen five sequences of a single video in order to maintain their attention on the assessment. Each test sequence was, thus, tested by 19 subjects. This was considered necessary to ensure that a few participants did not skew the findings. The subjective assessments were conducted with controlled ambient light in a laboratory. The monitor used to present the sequences was 30 inches, with a resolution of 1920 x 1080 and an aspect ratio of 16:9.

VI. EXPERIMENTAL RESULTS

Readers looking for a comprehensive analysis of Illinois and CUBIC could refer to the following work [47] in this and the rest of the experiments.

A. Objective QoE evaluation

1) Environment 1: The Conventional results show West-wood+ outperforming the others, see Table II. The goal of TCP's congestion control mechanism is to fairly share the bandwidth at a bottleneck link amongst multiple TCP connections. [48] provides an elegant and intuitive explanation on TCP congestion control showing why a fair share of a bottleneck bandwidth is obtained among competing TCP connections. However, in the presence of competing video flows, sharing is no longer fair or stable. This happens as players requesting video segments experience overlapping ON-OFF periods.TCP congestion mechanism forces the use of all bandwidth when Westwood+ players are in their ON period. This results in bandwidth oscillation amongst them. Results confirm that the bandwidth estimate obtained by filtering the ACKs is more accurate and less oscillating than the one obtained by filtering the input rate, which, in fact, provides an overestimate of the available bandwidth. The Westwood+ adaptive window shrinking provides a congestion window that is decreased enough in the presence of heavy congestion. Hence, the demand for bandwidth changes. This may result in a change of bottleneck link bandwidth usage with decreases in amplitudes of the existing oscillatory effects. Fair share improves for Westwood+ players. Thus, stability improves for players using the Westwood+ approach. YeAH comes in second. This is because its precautionary decongestion prevents the bottleneck queue from building up too much, reducing queuing delays and diminishing packet losses due to buffer overflow. The effect of achieving fair share in this manner lowers player bitrate switches. PANDA's probe of the available bandwidth in player OFF periods and its conservative nature compliments Westwood+ adaptive window shrinking, thus enabling this combination to provide best QoE to users, see Tables III. ELASTIC always generates a traffic pattern that is identical to any long-lived TCP flow but Westwood+ con-gestion algorithm is able to cope when these flows compete, see IV.

2) Environment 2: The Conventional results show West-wood+ outperforming the others, see Table V. When additional players arrive the available bandwidth decreases for remain-ing players. The window setting of Westwood+ TCP tracks

ENVIRONMENT 1: CONVENTIONAL

Variant	Switches	Buffer Level	Utilization
Westwood+	42	15.44	2.02
YeAH	62	14.61	1.92
CUBIC	65	14.52	1.83
Illinois	72	13.69	1.68

TABLE III ENVIRONMENT 1: PANDA

Variant	Switches	Buffer Level	Utilization
Westwood+	38	17.12	2.16
YeAH	47	16.51	2.08
CUBIC	63	15.02	1.92
Illinois	47	16.49	2.00

the estimated bandwidth so that, if this estimate is a good measurement of the fair share, then the fairness is improved. YeAH comes in second. When the number of competing flows increases, every flow attempts to fill the buffer by the same number of packets independently of the perceived RTT, achieving the internal RTT fairness. PANDA's conservative rate estimators over a smoothed average network bandwidth helps avoid bandwidth overestimation caused by an increase in the number of players. This together with Westwood+ tracking of the estimated bandwidth enables this combination to perform best, see Table VI, Each new ELASTIC player aggressively competes for bandwidth but Westwood+ is still able to maintain better performance compared to other TCP variants, see Table VII.

3) Environment 3: The Conventional results show West-wood+ outperforming the others, see Table VIII. When players leave the congestion level decreases for remaining players. The adaptive window shrinking of Westwood+ provides a conges-tion window that is decreased not too much in the presence of light congestion. YeAH comes in second. This is because the precautionary decongestion prevents the bottleneck queue

TABLE IV

ENVIRONMENT 1: ELASTIC

Variant	Switches	Buffer Level	Utilization
Westwood+	45	13.18	2.00
YeAH	55	12.32	1.84
CUBIC	65	11.27	1.60
Illinois	59	12.11	1.76

TABLE VENVIRONMENT 2: CONVENTIONAL

Variant	Switches	Buffer Level	Utilization
Westwood+	45	18.75	1.76
YeAH	57	18.67	1.74
Illinois	65	18.71	1.62
CUBIC	108	18.50	1.38

TABLE VI

ENVIRONMENT 2: PANDA

Variant	Switches	Buffer Level	Utilization
Westwood+	36	20.02	1.97
YeAH	50	19.71	1.83
Illinois	54	19.24	1.64
CUBIC	81	19.11	1.51

TABLE VII

ENVIRONMENT 2: ELASTIC

Westwood+5918.211.62YeAH6217.331.59Illinois8117.961.54CUBIC11916.341.41))

TABLE VIII

ENVIRONMENT 3: CONVENTIONAL

Variant	Switches	Buffer Level	Utilization
Westwood+	49	19.85	1.97
YeAH	53	19.75	1.86
Illinois	61	19.75	1.79
CUBIC	84	19.21	1.52

TABLE IX ENVIRONMENT 3: PANDA

Variant	Switches	Buffer Level	Utilization
Westwood+	41	20.88	2.19
YeAH	55	19.61	1.75
Illinois	58	18.98	1.64
CUBIC	76	18.53	1.49

ENVIRONMENT 3: ELASTIC

Variant	Switches	Buffer Level	Utilization
Westwood+	71	18.04	1.84
YeAH	63	18.39	1.95
Illinois	78	17.93	1.55
CUBIC	91	17.77	1.37

from building up too much, reducing queuing delays and diminishing packet losses due to buffer overflow. PANDA's adaptive nature works well with Westwood+ slight decrease in bandwidth as congestion lowers because players leave the network, see Table IX. However, YeAH increments the congestion window aggressively which suits ELASTIC only ON periods making YeAH most appropriate for decreases in the number of ELASTIC players as more bandwidth becomes available, see Table X.

Environment 4: The Conventional results show 4) West-wood+ outperforming the others, see Table XI. This is because the adaptive window shrinking provides a congestion window that is decreased not too much in the presence of losses that are not due to congestion. YeAH comes in second. When a loss is detected by three duplicate ACKs, the current estimate of the bottleneck queue Q, can be exploited to find the value of packets that should be removed from the congestion window to empty the bottleneck buffer, yet leaving the pipe full. This rule is similar in principle to the one used by Westwood TCP [16]. This rule permits to obtain the full link utilization after a loss, for every value of the bottleneck buffer sizeand in case of losses independent of the congestion of the network (e.g. wireless links). To reduce the effects of ON-OFF periods PANDA tries to repress high bitrate selection. This maintains network health. This together with Westwood+ adaptive window shrinking gives this combination the best QoE performance in a congested bottleneck environment with additional packet loss, see Tables XII and XIII.

B. Subjective QoE evaluation

Compared with the existing TCP variants, Westwood+ pro-motes a conservative bitrate switching rate, fair sharing of bandwidth and rapid replenishment of buffer levels, thereby leading to a superior subjective user experience, as justified by its high MOS rating in all evaluation environments, see Figures 2, 3, 4 and 5.

TABLE XI
ENVIRONMENT 4: CONVENTIONAL

Variant	Switches	Buffer Level	Utilization
Westwood+	66	14.05	1.64
YeAH	85	13.99	1.34
Illinois	107	11.71	1.05
CUBIC	133	8.49	0.74

TABLE XII

ENVIRONMENT 4: PANDA

Variant	Switches	Buffer Level	Utilization
Westwood+	42	15.43	1.73
YeAH	53	14.01	1.42
Illinois	86	12.20	1.19
CUBIC	97	10.11	0.91

TABLE XIII ENVIRONMENT 4: ELASTIC

Switches	Buffer Level	Utilization
70	14.62	1.44
75	14.51	1.37
91	12.21	0.98
115	10.11	0.62
	Switches 70 75 91 115	Switches Buffer Level 70 14.62 75 14.51 91 12.21 115 10.11

VII. CONCLUSION

As network bandwidth requirements grow especially with video services there is an increasing need for better perfor-mance at the transport (TCP) and application areas (DASH over HTTP). We analyze combinations of clientbased DASH players with four high speed TCP variants (Westwood+, YeAH, Illinois and CUBIC). The Conventional, PANDA and ELASTIC client-side DASH players in congested bottleneck link conditions were tested against the TCP variants. Both objective and subjective QoE measures were obtained. Objec-tive measures includes video freezing, switching and stalling, while subjective measures were obtained from MOS scales. We observe that Westwood+ and YeAH are the with the benchmark most promising variants Conventional player. However, Westwood+ and PANDA gave the best overall results, except in the case of decreasing players where YeAH proved the most successful. In the future we believe this work would help researchers develop better TCP mechanisms that works best for certain DASH architectures for instance a DASH approach using PANDA's probing and Westwood+ AIAD

command window increase could be beneficial. In other future work we plan to test legacy TCP variants with current adaptive video streaming players, such as Pensieve [49] and S-MDP [13].



Fig.2. Environment 1: Subjective visual quality comparison in terms of average MOS scores for different streaming methods.



Fig.3. Environment 2: Subjective visual quality comparison in terms of average MOS scores for different streaming methods.



Fig.4. Environment 3: Subjective visual quality comparison in terms of average MOS scores for different streaming methods.



Fig 5. Environment 4: Subjective visual quality comparison in terms of average MOS scores for different streaming methods.

REFERENCES

- R. Rejaie, M. Handley, and D. Estrin, "Rap: An endto-end rate-based congestion control mechanism for realtime streams in the internet," in INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 3. IEEE, 1999, pp. 1337– 1345.
- [2] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "Tcp friendly rate control (tfrc): Protocol specification," Tech. Rep., 2002.
- [3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol-http/1.1," Tech. Rep., 1999.
- [4] J. Postel, "Transmission control protocol," 1981.
- [5] S. Guha and P. Francis, "Characterization and measurement of tcp traversal through nats and firewalls," in Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement. USENIX Association, 2005, pp. 18–18.
- [6] L. F. Sarmenta and S. Hirano, "Bayanihan: Building and studying web-based volunteer computing systems using java," Future Generation Computer Systems, vol. 15, no. 5, pp. 675–686, 1999.
- [7] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Behavior analysis of tcplinux variants," Computer Networks, vol. 56, no. 1, pp. 462–476, 2012.
- [8] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "Tcpwestwood: Bandwidth estimation for enhanced transport over wireless links," in Proceedings of the 7th annual international conference on Mobile computing and networking. ACM, 2001, pp. 287–297.
- [9] S. Ha, I. Rhee, and L. Xu, "Cubic: a new tcp-friendly high-speed tcp variant," ACM SIGOPS operating systems review, vol. 42, no. 5, pp. 64–74, 2008.
- [10] A. Baiocchi, A. P. Castellani, and F. Vacirca, "Yeahtcp: yet another highspeed tcp," in Proc. PFLDnet, vol. 7, 2007, pp. 37–42.
- [11] S. Liu, T. Bas, ar, and R. Srikant, "Tcp-illinois: A loss-and delay-based congestion control algorithm for

high-speed networks," Performance Evaluation, vol. 65, no. 6-7, pp. 417–440, 2008.

- [12] K. Khan and W. Goodridge, "B-dash: broadcast-based dynamic adaptive streaming over http," International Journal of Autonomous and Adaptive Communications Systems, vol. 12, no. 1, pp. 50–74, 2019.
- [13] Khan, Koffka, and Wayne Goodridge. "S-mdp: Streaming with markov decision processes." IEEE Transactions on Multimedia 21, no. 8 (2019): 2012-2025.
- [14]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, 2014.
- [15]S. Sen, J. L. Rexford, J. K. Dey, J. F. Kurose, and D. F. Towsley, "Online smoothing of variable-bit-rate streaming video," IEEE Transactions on Multimedia, vol. 2, no. 1, pp. 37–48, 2000.
- [16] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "Elastic: a client-side controller for dynamic adaptive streaming over http (dash)," in 2013 20th International Packet Video Workshop. IEEE, 2013, pp. 1–8.
- [17]S. Barre, O. Bonaventure, C. Raiciu, and M. Handley, "Experimenting with multipath tcp," ACM SIGCOMM Computer Communication Re-view, vol. 41, no. 4, pp. 443–444, 2011.
- [18] K. Khan and W. Goodridge, "Energy aware ad-hoc on demand multipath distance vector routing," International Journal of Intelligent Systems and Applications, vol. 7, no. 7, pp. 50–56, 2015.
- [19] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mas-colo, "Tcpwestwood: Congestion window control using bandwidth estimation," in GLOBECOM'01. IEEE Global Telecommunications Con-ference (Cat. No. 01CH37270), vol. 3. IEEE, 2001, pp. 1698–1702.
- [20] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "Tapas: a tool for rapid prototyping of adaptive streaming algorithms," in Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming. ACM, 2014, pp. 1–6.
- [21]K. Khan and W. Goodridge, "Qoe evaluation of dynamic adaptive streaming over http (dash) with promising transport layer protocols," CCF Transactions on Networking, pp. 1–16, 2020.
- [22] M. Obrist, P. Cesar, D. Geerts, T. Bartindale, and E. F. Churchill, "Online video and interactive tv experiences," interactions, vol. 22, no. 5, pp. 32–37, 2015.
- [23] A. McZeal Jr, "Multifunctional world wide walkie talkie, a tri-frequency cellular-satellite wireless instant messenger computer and network for establishing global wireless volp quality of service (qos) communica-tions, unified messaging, and video conferencing via the internet," Jul. 13 2004, uS Patent 6,763,226.

- [24] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," in ACM SIGCOMM Computer Communication Review, vol. 43, no. 4. ACM, 2013, pp. 339–350.
- [25] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, "Internet video delivery in youtube: From traffic measurements to quality of experience," in Data Traffic Monitoring and Analysis. Springer, 2013, pp. 264–301.
- [26] P. Brooks and B. Hestnes, "User measures of quality of experience: why being objective and quantitative is important," IEEE network, vol. 24, no. 2, pp. 8–13, 2010.
- [27] K. U. R. Laghari and K. Connelly, "Toward total quality of experience: A qoe model in a communication ecosystem," IEEE Communications Magazine, vol. 50, no. 4, pp. 58–65, 2012.
- [28] A. Takahashi, D. Hands, and V. Barriac, "Standardization activities in the itu for a qoe assessment of iptv," IEEE Communications Magazine, vol. 46, no. 2, pp. 78–84, 2008.
- [29]S. Tavakoli, K. Brunnstrom," J. Gutierrez, and N. Garc'ıa, "Quality of experience of adaptive video streaming: investigation in service parameters and subjective quality assessment methodology," Signal Processing: Image Communication, vol. 39, pp. 432–443, 2015.
- [30] M. T. Vega, C. Perra, F. De Turck, and A. Liotta, "A review of predictive quality of experience management in video streaming services," IEEE Transactions on Broadcasting, vol. 64, no. 2, pp. 432– 445, 2018.
- [31] Y. Chen, K. Wu, and Q. Zhang, "From qos to qoe: A tutorial on video quality assessment," IEEE Communications Surveys & Tutorials, vol. 17, no. 2, pp. 1126–1165, 2014.
- [32] N. Eswara, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. Channappayya, "Perceptual qoeoptimal resource allocation for adaptive video streaming," IEEE Transactions on Broadcasting, vol. 66, no. 2, pp. 346–358, 2019.
- [33] M. Seufert, "Fundamental advantages of considering quality of experi-ence distributions over mean opinion scores," in 2019 Eleventh Inter-national Conference on Quality of Multimedia Experience (QoMEX). IEEE, 2019, pp. 1–6.
- [34] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in im-age/video quality assessment," Electronics letters, vol. 44, no. 13, pp. 800–801, 2008.
- [35]D. M. Allen, "Mean square error of prediction as a criterion for selecting variables," Technometrics, vol. 13, no. 3, pp. 469–475, 1971.
- [36] A. Martin, J. Egana, J. Florez, J. Montalban, I. G. Olaizola, M. Quartulli, Viola, and M. Zorrilla, "Network resource allocation system for qoe-aware delivery of media services in 5g networks," IEEE Transactions on Broadcasting, vol. 64, no. 2, pp. 561– 574, 2018.

- [37]L. Skorin-Kapov, M. Varela, T. Hoßfeld, and K.-T. Chen, "A survey of emerging concepts and challenges for qoe management of multimedia services," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 14, no. 2s, pp. 1–29, 2018.
- [38] K. Khan and W. Goodridge, "Qoe in dash," International Journal of Advanced Networking and Applications, vol. 9, no. 4, pp. 3515–3522, 2018.
- [39] M. T. A. Abdullah, J. Lloret, A. Canovas' Solbes, and L. Garc'ia-Garc'ia, "Survey of transportation of adaptive multimedia streaming service in internet," Network Protocols and Algorithms, vol. 9, no. 1-2, pp. 85–125, 2017.
- [40] A. Ahmed, Z. Shafiq, H. Bedi, and A. Khakpour, "Suffering from buffering? detecting qoe impairments in live video streams," in 2017 IEEE 25th International Conference on Network Protocols (ICNP). IEEE, 2017, pp. 1–10.
- [41] Staelens, Nicolas, Jonas De Meulenaere, Maxim Claeys, Glenn Van Wallendael, Wendy Van den Broeck, Jan De Cock, Rik Van de Walle, Piet Demeester, and Filip De Turck. "Subjective quality assessment of longer duration video sequences delivered over HTTP adaptive streaming to tablet devices." IEEE Transactions on Broadcasting 60, no. 4 (2014): 707-714.
- [42] J. Chen, Z. Wei, S. Li, and B. Cao, "Artificial intelligence aided joint bit rate selection and radio resource allocation for adaptive video streaming over f-rans," IEEE Wireless Communications, vol. 27, no. 2, pp. 36–43, 2020.
- [43]O. El Marai, T. Taleb, M. Menacer, and M. Koudil, "On improving video streaming efficiency, fairness, stability, and convergence time through client–server cooperation," IEEE Transactions on Broadcasting, vol. 64, no. 1, pp. 11–25, 2017.
- [44] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, "Tfdash: A fairness, stability, and efficiency aware rate control approach for multiple clients over dash," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 1, pp. 198–211, 2017.
- [45] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu, "Adap-tive wireless video streaming based on edge computing: Opportunities and approaches," IEEE Transactions on services Computing, vol. 12, no. 5, pp. 685–697, 2018.
- [46] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over http," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 562–585, 2018.
- [47] C. B. Ameur, E. Mory, and B. Cousin, "Combining traffic-shaping methods with congestion control variants for http adaptive streaming," Multimedia Systems, vol. 24, no. 1, pp. 1–18, 2018.
- [48] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease al-gorithms for congestion avoidance in computer networks," Computer Networks and ISDN systems, vol. 17, no. 1, pp. 1–14, 1989.

[49] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video stream-ing with pensieve," in Proceedings of the Conference of the ACM Special Interest Group on Data Communication, 2017, pp. 197–210.