

# Transport layer performance in DASH bottlenecks

**Koffka Khan**

Department of Computing and Information Technology The University of the West Indies, Trinidad and Tobago, W.I  
Email: koffka.khan@gmail.com

**Leah Joseph**

Department of Computing and Information Technology The University of the West Indies, Trinidad and Tobago, W.I  
Email: joseph.leah23@hotmail.com

**Emilie Ramsahai**

The School of Science, Artificial Intelligence and Computing, The University of the West Indies, Five Islands Campus,  
Trinidad and Tobago, W.I.  
Email: emilieannramsahai@gmail.com

---

## ABSTRACT

---

**A lot of research has been done in the area of TCP long-lived (TLL) and short-lived flows and well as UDP but there is limited research done on how they can affect adaptive video streaming at a single household bottleneck router. The goal of Adaptive streaming is to deliver videos to the user in the most efficient way possible and in the best quality possible. Several factors can affect the quality of videos including internet connection and other applications using the same router. It is important to note which applications can negatively affect streaming and cause issues such as poor-quality videos and buffering. This paper investigates the impact that various applications that use TCP long-lived flows have on streaming and its effect is compared to applications that use UDP flows. The aim is to determine which type of flow affects adaptive streaming the most. This investigation is carried out with the use of the TAPAS (a Tool for rApid Prototyping of Adaptive Streaming) player. Live experiments were conducted at a household bottleneck link with a client player, a TCP or UDP application. We show that UDP flows disrupt adaptive video streaming to a greater extent when compared to TCP TLL flows.**

**Keywords – Adaptive streaming, bottleneck, live experiments, metrics, TAPAS, TCP long-lived flows, TCP short-lived flows, UDP flows.**

---

Date of Submission: Oct 01, 2021

Date of Acceptance: Dec 24, 2021

---

## I. INTRODUCTION

Video streaming [27][30][31] makes up more than half of the Internet's traffic and it was found that 69% of respondents use at least one streaming service as it is cheaper than traditional tv subscriptions [51]. Due to the data intensive nature of video streaming, throughput variations are common and without sufficient resources the video playback will freeze, or the video would be streamed at a lower bitrate [50]. Adaptive streaming aims to solve these issues. Adaptive streaming is the number one video streaming technology in today's world with several well-known companies such as Microsoft, Adobe, Netflix, and Apple opting to use this technology [9]. Adaptive streaming is defined as a technology designed to deliver video to the user in the most efficient way possible and in the highest usable quality for each specific user [44]. The main advantage of adaptive streaming over other streaming methods is as it says in its name, it adapts, it adjusts its quality to match bandwidth. Due to the adaptation to bandwidth there is a controlled degradation in quality and better QoE [1].

Adaptive streaming works by detecting a user's bandwidth and CPU capacity in real time and adjusting the quality of the media stream accordingly. A media file is divided into several slices called chunks or segments that are a few seconds long. These segments are then encoded into different quality levels called representations and stored at the server. An encoder is used to encode the audio or video file at multiple bit rates into an adaptive format.

Several versions of a segment exist, as they are of different bitrates, resolution, and quality [45]. During a player's session, encodings are swapped in and out in order to meet the available resources. This approach results in minimal buffering and smoother video playback. When a client makes a request to adaptively stream media a manifest file (a media presentation description or MPD file) is generated [46]. This file gives hierarchical metadata information about the structure and availability of media for streaming. It essentially gives information about quality and rate for different media portions. After the file is fully generated it is then sent to the client that requested it. Currently adaptive video streaming players are severely impacted by applications using TCP long-lived flows and there is no known solution for this except ELASTIC (fEedback Linearization Adaptive STrEaming Controller) [16].

It was found that more than 90% of internet traffic comes from TCP [40] and TCP flows are found in several daily applications for chat or file transfer. TCP connections usually affect normal network functioning because of its high resource consumption. TCP carries bursty, congestion sensitive traffic and long TCP connections are susceptible to several issues such as TCP congestion and TCP connection recovery. Short-lived TCP flows spend most of its duration in the slow start phase when the congestion window is increased by a significant amount [19]. However, long-lived TCP flows start in the slow start phase, but they spend most of their time in the congestion avoidance phase where Additive Increase Multiplicative

Decrease (AIMD) congestion control is carried out [18]. Aside from TCP, it is important to examine the impact of User Datagram Protocol (UDP) [41] on adaptive streaming.

UDP is a known competitor with TCP flows since TCP's performance degrades whenever they are competing in the same network [28]. This is due to the type of traffic that it carries (unresponsive and real-time). UDP is the second most popular transport protocol on the internet. Unlike TCP, UDP is connectionless and does not guarantee delivery of packets through error checking. It is connectionless as it does not initiate handshaking to establish a secure connection like TCP applications do. When data is sent in packets on a UDP connection data can be easily lost or duplicated and it does not verify whether the packets sent were received at its destination. The loss of packets in UDP connections result in errors that can spread across several frames causing negative effects on user QoE such as frame distortion [23].

Often, TCP or UDP flows can be found competing for bandwidth at a household router at the same time as an adaptive streaming process. If there is a single application running on a connection it does not need to overwhelm the bottleneck router or drop packets. However, at times different applications are simultaneously being used which causes competition at the router. Certain applications are more bandwidth-intensive than others and as a result there is packet loss which causes the quality of some applications to decrease [21]. TCP long-lived flows tend to increase the competition at the router as each individual TCP connection needs to evaluate how it can maximize its share of bandwidth. TCP flows causes what is called a downward spiral which is a dramatic anomalous drop in the video playback rate [33]. During streaming, applications usually select a streaming rate and if done incorrectly a feedback loop can occur which causes the video quality to become very variable and low. Whenever a TCP flow is introduced into a network videos tend to have a playback rate that is extremely below the bandwidth available as TCP flows do not allow a fair share of bandwidth [10].

The work in this paper observes and analyzes the behavior of an adaptive streaming player when it is used alongside other internet applications. The purpose of this paper is to evaluate the effect of TCP long-lived flows [6] on adaptive video streaming. The paper will investigate the impact that various applications using TCP long-live flows have on streaming. Further, it will also test against applications using UDP flows. The evaluation will be done using live experiments on the TAPAS player [17]. The main objective in this paper is to find out whether TCP long-lived flows or UDP flows affect adaptive streaming more and to emphasize the importance of studying transport layer protocols in network congestion.

In the following sections, the paper is broken down and analyzed. In section 2 there is a literature review which highlights related work based on the main topics explored in this paper. Section 3 explores the methodologies used to go about finding the results of experiments. Section 4 describes the experiments and gives an in depth write up

on how tests were carried out. Section 5 and 6 lists the results of experiments, analyses, and discusses it. A summary of findings, intentions for future work and lessons learnt would be found in the conclusion.

## II. LITERATURE REVIEW

Some applications require the use of TCP or UDP flows to communicate and send and receive packets, cf. Table 1. These internet flows were chosen mainly because they are the most popular types of flows on the internet as they make up a majority of the traffic. TCP flows are also known to adversely affect video streaming so this experiment aims to analyze how much of an impact it can have. The experiments in the following sections will be using applications such as Skype, YouTube and Video Games which all use TCP long lived flows as well as Discord which uses UDP flows.

**Table 1: Online Applications – TCP and UDP usage**

Application	SLF	LLF	UDP
<i>BitTorrent</i>	*		*
<i>Discord</i>			*
<i>Facebook Messenger</i>	*		*
<i>Hulu</i>	*		
<i>Netflix</i>	*	*	
<i>OpenVPN</i>			*
<i>Pandora</i>		*	
<i>Skype</i>		*	*
<i>UFTP</i>			*
<i>Viber</i>			*
<i>Video games</i>		*	*
<i>YouTube</i>		*	

Most commercial video games use either TCP or UDP as their transport protocol [22]. Massively Multiplayer Online Role-Playing Games (MMORPGs) are the most popular type of game in the gaming industry [35]. Several MMORPGs use the TCP protocol to send packets between the client and the server [14]. Despite the protocol being used in so many gaming applications, TCP is not the most effective transport protocol as there can be a poor user experience because of TCP's need to ensure the delivery of packets. Although UDP is a popular transport protocol it may be difficult to apply it directly to a game due to its unreliable data transfer. The protocol chosen all depends of the type of game and the nature of the packets sent. If the MMORPG has a lot of kinetics UDP would be the suited protocol while games that are slower should use TCP [43].

YouTube is another popular streaming application that uses TCP over HTTP. However, unlike Netflix, a very popular DASH application, it uses progressive download. The flash video format (FLV) [37] is used for a majority of video clips on YouTube. The FLV data stored in the packets received is placed in a buffer by the player, the buffer then supplies the packets to video data playback. The use of TCP connections can result in negative effects

such as delay of TCP transmissions and throughput degradation. The buffer's goal is to lessen the delay of TCP transmissions and throughput degradation. If TCP throughput reduction occurs continuously, this can result in a rebuffering event [25]. Aside from the cons of TCP usage on YouTube, Bampis states that TCP is very useful for this application because it guarantees that packets are delivered without loss or error [11]. He goes on to state that the only thing that is really affected using TCP is video playback.

Discord is a VoIP application that is used widely in the gaming community. UDP is used for voice receiving and transferring. Discord documentation explains that after voice properties are received from the server a handshaking process then takes place [38]. This is done by first opening the UDP connection and once the UDP port is discovered data can be sent and received. Skype is also a VoIP application that is very popular. Korczyński and Duda points out the fact that Skype's design allows it to use both TCP and UDP flows [34]. It uses them to its advantage since it can switch between the flows based on the restrictions in the network. Skype usually uses TCP to make its initial connection. Skype video tend to favor TCP as it is very responsive in the presence of congestion [15]. Freire, Zivianni and Salles point out that because of the mix of protocols it is often difficult to determine the type of traffic being used [20].

### III. METHODOLOGY

In our simulated environment we use software to generate TCP and UDP flows [26]. This mimics the TCP and UDP applications described in Section 2. Our set-up is shown in Figure 3. We vary the strength of the TCP, UDP or both flows during the experiment. For all experiments there are two adaptive video streaming players using the Conventional player algorithm, see Figure 1.

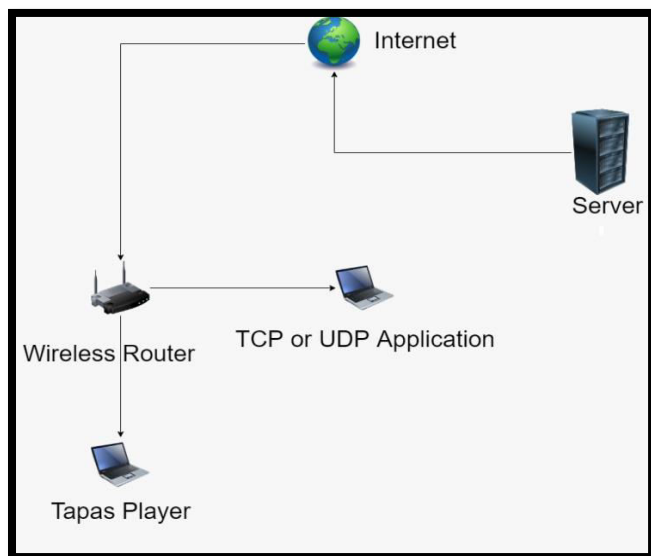


Fig. 1. Network setup.

#### A. Scenario 1

We use TCP LLF flows only. However, during the first three minutes of the experiment we allow it to occupy one quarter of the bandwidth. Then for the next three minutes we allow it to occupy one half of the bandwidth. Finally, in the last three minutes of the experiment we allow it to occupy three quarters of the bandwidth. This scenario mimics a TCP LLF application, see Figure 2.

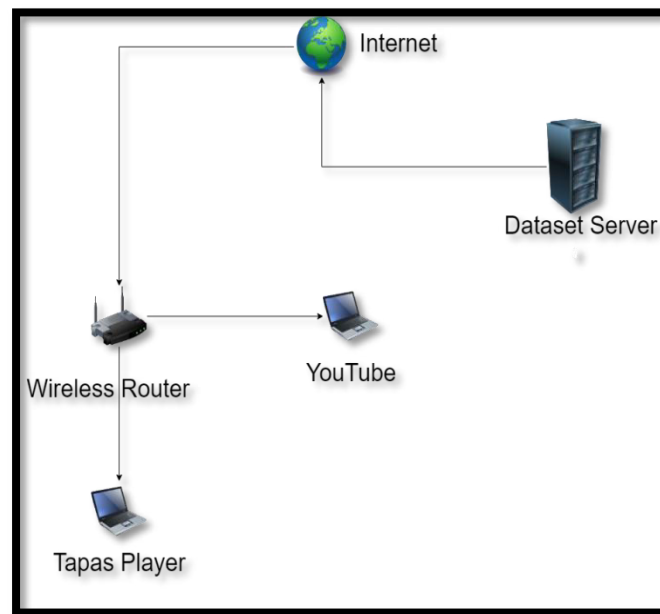


Fig. 2. Network setup: Scenario 1

#### B. Scenario 2

We use UDP flows only. However, during the first three minutes of the experiment we allow it to occupy one quarter of the bandwidth. Then for the next three minutes we allow it to occupy one half of the bandwidth. Finally, in the last three minutes of the experiment we allow it to occupy three quarters of the bandwidth. This scenario mimics a UDP application, see Figure 3.

#### C. Scenario 3

We use TCP LLF and UDP flows. However, during the first three minutes of the experiment we allow them to occupy one quarter of the bandwidth (one eighth each). Then for the next three minutes we allow them to occupy one half of the bandwidth (one quarter each). Finally, in the last three minutes of the experiment we allow them to occupy three quarters of the bandwidth (three eighths each). This scenario mimics an application that use UDP and TCP LLF, see Figure 4.

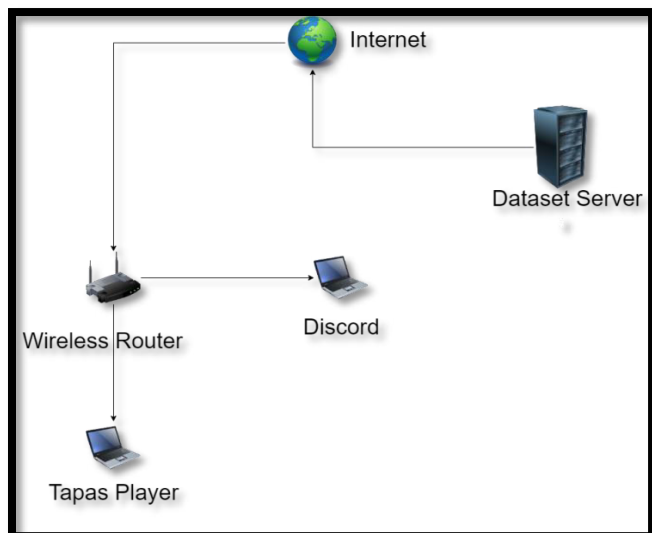


Fig. 3. Network setup. Scenario 2

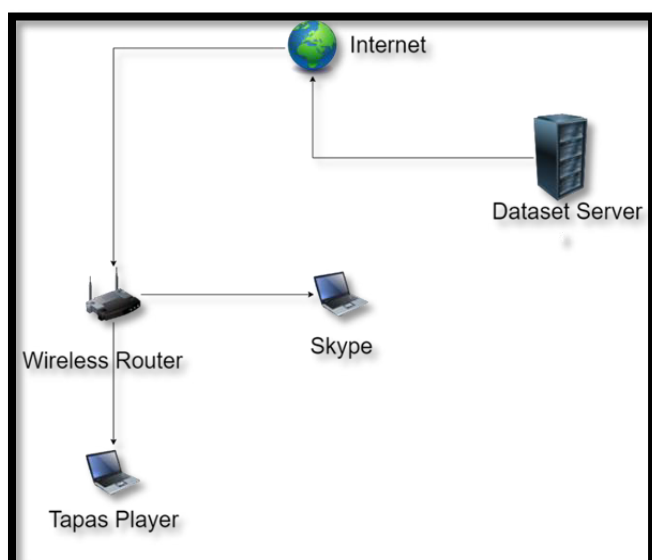


Fig. 4. Network setup. Scenario 3

#### IV. EXPERIMENTAL SETUP

##### A. Details of DASH-based Experiment Setup

On the same host system, a virtual network is set up, resulting in a bespoke emulation framework. Client players, video servers, and a bottleneck link make up our configuration. The server is running on a Windows 10 PC. All tests are run on a Windows 10 client with an Intel(R) Core(TM)i7-5500U processor running at 2.40GHz, 16.00 GB of physical memory, and an Intel(R) HD Graphics processor. It provides video data to clients running an Ubuntu operating system on VMware. Physical memory for the virtual machine is set to 12GB.

Ubuntu 15.04 Linux is used to run TAPAS. Figure 5 shows how the TAPAS Adaptive Video Controller client sends various video segment bitrate level requests to the Apache server.

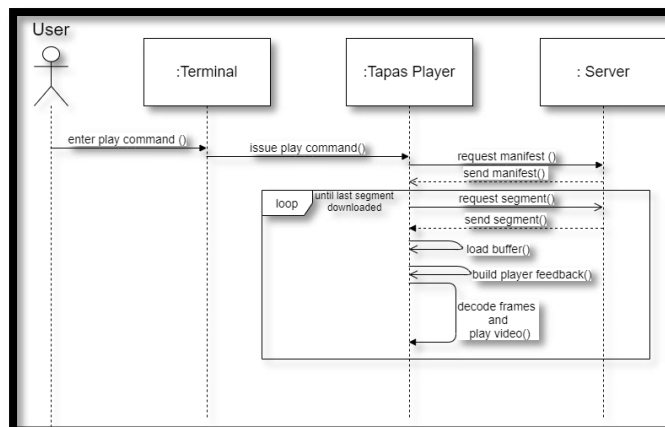


Fig. 5. TAPAS player streaming: sequence diagram [17].

TAPAS enables multi-client scenarios by allowing numerous instances of the player to be generated. The connection between the adaptive streaming algorithm at the controller and the TAPAS player is the focus of this research (cf. Figure 6).

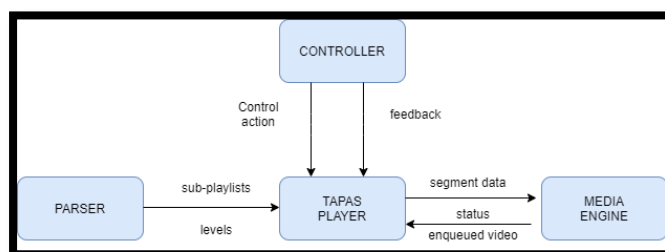


Fig. 6. TAPAS adaptive player components

All communication between clients and servers passes via the bottleneck, which is controlled by VMware settings that allow bandwidth limitations to be established throughout the experiment, as shown in Figure 7. Both the HTTP Live Streaming (HLS) and Dynamic Adaptive Streaming over HTTP (DASH) formats are supported by TAPAS. Scenario 1, 2, and 3 algorithms have been tested and proved to function on MPEG-DASH [48], and Apple HTTP Live Streaming (HLS). This makes it suitable for video-on-demand (VOD) [42] and live streaming [32] applications, such as real-time video chats. However, in this study article, the MPEG-DASH standard is utilized for testing since it makes the trials more comparable to those in the literature, for example, [52][8][2].

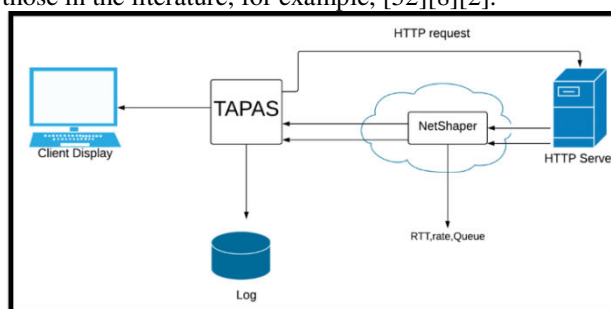


Fig. 7. TAPAS adaptive player simulation environment

The trials were conducted using the ten-minute MPEG-DASH video sequence "Elephant's Dream"1, which was encoded at twenty various bitrates ranging from 46 to 4200 Kbps and five different resolutions ranging from 320x240 to 1920x1080 (cf. Table II). The video is encoded using the AVC1 codec at 24 frames per second (fps) [13]. The duration of the fragment is 2 seconds, and it is recorded in the mpd playlist appropriately. The Apache server holds all of the DASH files (.m4s fragments and.mpd playlists). In the TAPAS controller, we used the conventional client-side algorithm. The traditional method is enabled by default and serves as a benchmark against which alternative algorithms can be measured.

### B. QoE Metrics

When they represent observable technical factors, computational media quality models are objective [49]. They have access to cumulative technical advancements, especially those that have an influence on quality. However, the utility of these models is decreased since they must be continually verified against fresh viewer test results as new parameters are added. The quality of experience (QoE) measures how well viewers perform in real-world situations. There are numerous methods for collecting objective QoE in laboratory and field studies, as well as various types of facilities and customer instances. A fundamental point is that objective QoE metrics for viewer perception and beyond are applicable [29]. This extends QoE to the domain of viewing experience. The subjective viewer metric, on the other hand, remains an important component of the QoE study. As a result, a combination of quantitative and subjective QoE assessments best expresses the dynamic nature of QoE. Within the context of a DASH streaming session, accidents in the network path, such as packet failures and reordering, reduce TCP efficiency. When this happens, video playing is paused until new video data is available. This has a negative impact on user-quality-of-experience. Many additional factors, such as display uniformity and replay smoothness, have an impact on user-QoE [54]. QoE is represented by a Mean Opinion Score (MOS) of 1 ('Bad') to 5 ('Excellent') [47]. Subjective or objective tests are used to get mean opinion ratings. Quantitative measures like the Peak-Signal-to-Noise-Ratio (PSNR) [19] and Mean Square Error (MSE) [36] aren't acceptable for HTTP video streaming. They solely look at spatial video quality. The creation of QoE measuring techniques, performance measures, and reporting procedures play a significant role in enhancing DASH service delivery. Playback smoothness and video fidelity are inextricably related from a human perspective when it comes to quality of experience (QoE). QoE measurements give various aspects for assessing user-perceived QoE [7][12]. During streaming, a situation may develop in which the video streaming application consumes more data than the incoming video bitrates. The margin for payouts is getting smaller. However, due to insufficient data in the playout buffer, the video eventually pauses enabling playing to

continue. By underrunning or being low on/empty of the needed video segments, stalling inhibits video replay [39]. The duration of an interruption determines whether or not to re-buffer. Viewers will experience prolonged periods of pausing and substantial runtime buffering. Viewers prefer a situation with a single extended freeze over a streaming session with multiple small freezes [53]. As the duration of the impairment grows longer, researchers see a decrease in average video output. Video stalls, on the other hand, are more severe than a decrease in frame rate. They discovered that video pauses at odd intervals are weaker than those at regular intervals. Variations in bandwidth might cause video playback to be interrupted. Packet loss due to buffer overruns (buffer is full, therefore packet reception results in drops) causes packets to be retransmitted, resulting in delays. Viewers are irritated by playback interruptions, and delays, such as flashing, have detrimental consequences. Moving to an intermediate rate is preferred over multiple broad magnitude rate shifts before switching to a higher rate, and a continuous rate is preferred over switches if the continuous rate is higher than the base rate, according to researchers [24]. These events should be accounted for when calculating QoE. For these objectives, we used three common target QoE measures from the adaptive video streaming literature:

- i. The aggregate throughput during an experiment divided by the available bandwidth in that experiment is the utilization metric [4] (cf. Equation 1, where  $t$  is the throughput at time and  $bw$  is the experimental available bandwidth).

$$Utilization = \frac{\sum_{i=0}^{n-1} tp_i}{bw} \quad (6)$$

- ii. Unfairness: Let  $JainFair_t$  be the Jain fairness index (cf. Equation 2) computed on the average received rates [5],  $r_i$ , (cf. Equation 3) at time  $t$  over all players. The unfairness at time  $t$  is defined as  $\sqrt{1 - JainFair_t}$ . A lower value implies a more fair allocation.

$$r_i = \frac{downloaded\ bytes}{time\ interval} \quad (9)$$

$$JFI = \frac{(\sum_{i=1}^n r_i)^2}{n \sum_{i=1}^n r_i^2} \quad (10)$$

- iii. Instability: The instability [3] for player  $i$  at time  $t$  is given in Equation 4, where  $w(d) = k - d$  is a weight function that puts more weight on more recent samples.  $k$  is selected as 20 seconds.

$$Instability = \frac{\sum_{d=0}^{k-1} |r_{i,t-d} - r_{i,t-d-1}| * w(d)}{\sum_{d=0}^{k-1} r_{i,t-d} * w(d)} \quad (7)$$

C. Videos

The segment specifications (bitrate and resolution) of one test movie (Elephant's Dream) is displayed here, cf. Table 2. Red Bull Playstreets is the video on the second player.

**Table 2: Elephant's Dream: Video levels, bitrates and resolutions**

Video level	Bitrate (kbps)	Resolution
$l_0$	46.0	320x240
$l_1$	91.0	320x240
$l_2$	131.0	320x240
$l_3$	180.0	480x360
$l_4$	222.0	480x360
$l_5$	261.0	480x360
$l_6$	328.0	480x360
$l_7$	382.0	480x360
$l_8$	523.0	854x480
$l_9$	594.0	854x480
$l_{10}$	796.0	1280x720
$l_{11}$	1000.0	1280x720
$l_{12}$	1200.0	1280x720
$l_{13}$	1500.0	1280x720
$l_{14}$	2100.0	1920x1080
$l_{15}$	2400.0	1920x1080
$l_{16}$	3000.0	1920x1080
$l_{17}$	3400.0	1920x1080
$l_{18}$	3800.0	1920x1080
$l_{19}$	4200.0	1920x1080

V. RESULTS

The first experiment illustrates two players competing at a 20Mbps bottleneck link. Table 1 gives the results of Scenario 1, see Section 3. As TCP LLF flows increase the utilization, unfairness and instability of the two players sharing the bottleneck link gets worst, see Table 3.

**Table 3: Scenario 1 – TCP LLF**

Flow % of bandwidth	25%	50%	75%
Time	0-3 min	3-6 min	6-9 min
Utilization	0.81	0.75	0.64
Unfairness	0.008	0.015	0.091
Instability	0.025	0.037	0.079

The second experiment illustrates two players competing at a 20Mbps bottleneck link and stopping or pausing during the experiment. Table 2 gives the results of Scenario 2, see Section 2. As UDP flows increase the utilization, unfairness and instability of the two players sharing the bottleneck link gets worst, see Table 4.

The UDP flow's throughput increases the source rate faster than TCP. This is because of the TCP connection's slow start phase. Thus, for the first three minutes we observe UDP having a more pronounced effect on streaming utilization, unfairness and instability, see Table 1 and 2. Afterwards, because of the TCP ACKs broadcast in the uplink, which impact TCP downlink consumption, UDP marginally outperforms TCP consuming more bandwidth. Thus, streaming applications competing with UDP flows are affected more than TCP LLFs.

**Table 4: Scenario 2 - UDP**

Flow % of bandwidth	25%	50%	75%
Time	0-3 min	3-6 min	6-9 min
Utilization	0.78	0.73	0.61
Unfairness	0.011	0.018	0.096
Instability	0.034	0.046	0.086

The third experiment illustrates two players competing at a 100Mbps bottleneck link with increasing number of players up to 20. Table 3 gives the results of Scenario 3, see Section 3. As TCP LLF and UDP flows increase the utilization, unfairness and instability of the two players sharing the bottleneck link gets worst, see Table 5.

**Table 5: Scenario 3: TCP LLF and UDP**

Flow % of bandwidth	25%	50%	75%
Time	0-3 min	3-6 min	6-9 min
Utilization	0.77	0.68	0.54
Unfairness	0.027	0.038	0.105
Instability	0.037	0.061	0.097

VI. CONCLUSION

Transport layer protocols such as TCP and UDP are essential in enabling proper function of Internet applications. They are used in many gaming, video streaming and communication-based applications. These are common in household use and usually will involve competition at the home router when many family members are performing Internet tasks. We investigate how TCP long-lived and UDP flows affect shared bottleneck video streaming. We develop simulations across a shared bottleneck link with video streaming applications and TCP or UDP flows. It was found that both flows impacted streaming significantly. However, UDP was more disruptive to the streaming performance. In future work we intend to investigate the performance of these flows in 5G networks.

REFERENCES

[1] Abar, Tasnim, Asma Ben Letaifa, and Sadok El Asmi. "Heterogeneous multiuser QoE enhancement over DASH in SDN networks." *Wireless Personal Communications* 114, no. 4 (2020): 2975-3001.

[2] Abar, Tasnim, Asma Ben Letaifa, and Sadok El Asmi. "User behavior-ensemble learning based improving QoE fairness in HTTP adaptive streaming over SDN approach." In *Advances in Computers*, vol. 123, pp. 245-269. Elsevier, 2021.

- [3] Abuteir, Rabee Mustapha, Anne Fladenmuller, and Olivier Fourmaux. "An SDN approach to adaptive video streaming in wireless home networks." In 2016 International wireless communications and mobile computing conference (IWCMC), pp. 321-326. IEEE, 2016.
- [4] Akhshabi, Saamer, Lakshmi Anantkrishnan, 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. 2012.
- [5] Akhshabi, Saamer, Lakshmi Anantkrishnan, 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. 2012.
- [6] Al Shinwan, Mohammad, Laith Abualigah, Nguyen Dinh Le, Chulsoo Kim, and Ahmad M. Khasawneh. "An intelligent long-lived TCP based on real-time traffic regulation." *Multimedia Tools and Applications* 80, no. 11 (2021): 16763-16780.
- [7] Alkhalili, Yassin, Jannis Weil, Tobias Meuser, Boris Koldehofe, Andreas Mauthe, Heinz Koeppl, and Ralf Steinmetz. "Towards QoE-Driven Optimization of Multi-Dimensional Content Streaming." *Electronic Communications of the EASST* 80 (2021).
- [8] Amirpour, Hadi, Ekrem Çetinkaya, Christian Timmerer, and Mohammad Ghanbari. "Towards optimal multirate encoding for HTTP adaptive streaming." In *International Conference on Multimedia Modeling*, pp. 469-480. Springer, Cham, 2021.
- [9] Ayad, Ibrahim, Youngbin Im, Eric Keller, and Sangtae Ha. "A practical evaluation of rate adaptation algorithms in http-based adaptive streaming." *Computer Networks* 133 (2018): 90-103.
- [10] Bağcı, Kadir Tolga, Kemal Emrehan Sahin, and A. Murat Tekalp. "Compete or collaborate: Architectures for collaborative DASH video over future networks." *IEEE Transactions on Multimedia* 19, no. 10 (2017): 2152-2165.
- [11] Bampis, Christos G., Zhi Li, Ioannis Katsavounidis, Te-Yuan Huang, Chaitanya Ekanadham, and Alan C. Bovik. "Towards Perceptually Optimized Adaptive Video Streaming-A Realistic Quality of Experience Database." *IEEE Transactions on Image Processing* (2021).
- [12] Barman, Nabajeet, and Maria G. Martini. "QoE modeling for HTTP adaptive video streaming—a survey and open challenges." *Ieee Access* 7 (2019): 30831-30859.
- [13] Borchert, Kathrin, Anika Seufert, Edwin Gamboa, Matthias Hirth, and Tobias Hoßfeld. "In vitro vs in vivo: does the study's interface design influence crowdsourced video QoE?." *Quality and User Experience* 6, no. 1 (2020): 1-16.
- [14] Bryant, Blake, and Hossein Saiedian. "An evaluation of videogame network architecture performance and security." *Computer Networks* 192 (2021): 108128.
- [15] Carlucci, Gaetano, Luca De Cicco, Stefan Holmer, and Saverio Mascolo. "Analysis and design of the google congestion control for web real-time communication (WebRTC)." In *Proceedings of the 7th International Conference on Multimedia Systems*, pp. 1-12. 2016.
- [16] Cofano, Giuseppe, Luca De Cicco, Thomas Zinner, Anh Nguyen-Ngoc, Phuoc Tran-Gia, and Saverio Mascolo. "Design and experimental evaluation of network-assisted strategies for HTTP adaptive streaming." In *Proceedings of the 7th international conference on multimedia systems*, pp. 1-12. 2016.
- [17] De Cicco, Luca, Vito Caldalaro, Vittorio Palmisano, and Saverio 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*, pp. 1-6. 2014.
- [18] Dumas, Vincent, Fabrice Guillemin, and Philippe Robert. "A Markovian analysis of additive-increase multiplicative-decrease algorithms." *Advances in Applied Probability* 34, no. 1 (2002): 85-111.
- [19] Erfurt, Johannes, Christian R. Helmrich, Sebastian Bosse, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. "A study of the perceptually weighted peak signal-to-noise ratio (WPSNR) for image compression." In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2339-2343. IEEE, 2019.
- [20] Freire, Emanuel P., Artur Ziviani, and Ronaldo M. Salles. "On metrics to distinguish skype flows from http traffic." In *2007 Latin American Network Operations and Management Symposium*, pp. 57-66. IEEE, 2007.
- [21] Go, Sharleen Joy Y., Cedric Angelo M. Festin, and Wilson M. Tan. "An SDN-based framework for improving the performance of underprovisioned IP Video Surveillance networks." *Journal of Network and Computer Applications* 132 (2019): 49-74.
- [22] Hahn, Bridger, Rishab Nithyanand, Phillipa Gill, and Rob Johnson. "Games without frontiers: Investigating video games as a covert channel." In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 63-77. IEEE, 2016.
- [23] Huang, Chun-ying, Yun-chen Cheng, Guan-zhang Huang, Ching-ling Fan, and Cheng-hsin Hsu. "On the Performance Comparisons of Native and Clientless Real-Time Screen-Sharing Technologies." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17, no. 2 (2021): 1-26.
- [24] Huang, Te-Yuan, Nikhil Handigol, Brandon Heller, Nick McKeown, and Ramesh Johari. "Confused, timid, and unstable: picking a video streaming rate is

- hard." In Proceedings of the 2012 internet measurement conference, pp. 225-238. 2012.
- [25] Huang, Xinyu, and Lijun He. "Playback experience driven cross layer optimisation of APP, transport and MAC layer for video clients over long-term evolution system." *IET Communications* 14, no. 13 (2020): 2176-2188.
- [26] Kanclirz, Jan, ed. *Netcat power tools*. Elsevier, 2008.
- [27] Khan, Koffka, and Wayne Goodridge. "B-DASH: broadcast-based dynamic adaptive streaming over HTTP." *International Journal of Autonomous and Adaptive Communications Systems* 12, no. 1 (2019): 50-74.
- [28] Khan, Koffka, and Wayne Goodridge. "QoE evaluation of dynamic adaptive streaming over HTTP (DASH) with promising transport layer protocols." *CCF Transactions on Networking* 3, no. 3 (2020): 245-260.
- [29] Khan, Koffka, and Wayne Goodridge. "QoE Evaluation of Legacy TCP Variants over DASH." *International Journal of Advanced Networking and Applications* 12, no. 5 (2021): 4656-4667.
- [30] Khan, Koffka, and Wayne Goodridge. "Server-based and network-assisted solutions for adaptive video streaming." *International Journal of Advanced Networking and Applications* 9, no. 3 (2017): 3432-3442.
- [31] Khan, Koffka, and Wayne Goodridge. "S-MDP: Streaming with markov decision processes." *IEEE Transactions on Multimedia* 21, no. 8 (2019): 2012-2025.
- [32] Kim, Jihye, and Minseong Kim. "Spectator e-sport and well-being through live streaming services." *Technology in Society* 63 (2020): 101401.
- [33] Kim, Seohyang, and Chongkwon Kim. "XMAS: An efficient mobile adaptive streaming scheme based on traffic shaping." *IEEE Transactions on Multimedia* 21, no. 2 (2018): 442-456.
- [34] Korczyński, Maciej, and Andrzej Duda. "Classifying service flows in the encrypted skype traffic." In 2012 IEEE International Conference on Communications (ICC), pp. 1064-1068. IEEE, 2012.
- [35] Lee, Zach WY, Christy MK Cheung, and Tommy KH Chan. "Understanding massively multiplayer online role-playing game addiction: A hedonic management perspective." *Information Systems Journal* 31, no. 1 (2021): 33-61.
- [36] Liu, Gaozheng, Li Chen, Weidong Wang, and Xiaohui Chen. "Hybrid beamforming based on minimum mean square error for multi-user multi-data stream system." In 2018 IEEE 4th International Conference on Computer and Communications (ICCC), pp. 124-128. IEEE, 2018.
- [37] LIU, Zhi-guo, and Ya-ming ZHANG. "On FLV Video Format." *Computer Knowledge and Technology* 2008 (2008): 20.
- [38] Mallari, Keri, Spencer Williams, and Gary Hsieh. "Understanding Analytics Needs of Video Game Streamers." In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1-12. 2021.
- [39] Nam, Hyunwoo, Kyung-Hwa Kim, Doru Calin, and Henning Schulzrinne. "Youslow: a performance analysis tool for adaptive bitrate video streaming." In Proceedings of the 2014 ACM conference on SIGCOMM, pp. 111-112. 2014.
- [40] Nechaev, Boris, Mark Allman, Vern Paxson, and Andrei V. Gurtov. "A Preliminary Analysis of TCP Performance in an Enterprise Network." *INM/WREN 10* (2010).
- [41] Postel, Jon. "RFC0768: User Datagram Protocol." (1980).
- [42] Rahe, Vanessa, Christopher Buschow, and Daniela Schlütz. "How users approach novel media products: Brand perception of Netflix and Amazon Prime video as signposts within the German subscription-based video-on-demand market." *Journal of Media Business Studies* 18, no. 1 (2021): 45-58.
- [43] Saldana, Jose. "The effect of multiplexing delay on MMORPG TCP traffic flows." In 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC), pp. 245-250. IEEE, 2014.
- [44] Sani, Yusuf, Andreas Mauthe, and Christopher Edwards. "Adaptive bitrate selection: A survey." *IEEE Communications Surveys & Tutorials* 19, no. 4 (2017): 2985-3014.
- [45] Schwarzmann, Susanna, Nick Hainke, Thomas Zinner, Christian Sieber, Werner Robitza, and Alexander Raake. "Comparing fixed and variable segment durations for adaptive video streaming: a holistic analysis." In Proceedings of the 11th ACM Multimedia Systems Conference, pp. 38-53. 2020.
- [46] Sebai, D., and E. Manai. "MPEG-DASH parametrisation for adaptive online streaming of different MOOC videos categories." *Multimedia Tools and Applications* (2021): 1-20.
- [47] Seufert, Michael. "Fundamental advantages of considering quality of experience distributions over mean opinion scores." In 2019 Eleventh international conference on quality of multimedia experience (QoMEX), pp. 1-6. IEEE, 2019.
- [48] Sodagar, Iraj. "The mpeg-dash standard for multimedia streaming over the internet." *IEEE multimedia* 18, no. 4 (2011): 62-67.
- [49] Somandepalli, Krishna, Tanaya Guha, Victor R. Martinez, Naveen Kumar, Hartwig Adam, and Shrikanth Narayanan. "Computational media intelligence: human-centered machine analysis of media." *Proceedings of the IEEE* (2021).
- [50] Taha, Miran, Aree Ali, Jaime Lloret, Paulo RL Gondim, and Alejandro Canovas. "An automated model for the assessment of QoE of adaptive video streaming over wireless networks." *Multimedia Tools and Applications* (2021): 1-22.



- [51] Udoakpan, Nokuphiwa, and Robertson Khan Tengeh. "The Impact of Over-the-Top Television Services on Pay-Television Subscription Services in South Africa." *Journal of Open Innovation: Technology, Market, and Complexity* 6, no. 4 (2020): 139.
- [52] Wei, Xuekai, Mingliang Zhou, Sam Kwong, Hui Yuan, Shiqi Wang, Guopu Zhu, and Jingchao Cao. "Reinforcement learning-based QoE-oriented dynamic adaptive streaming framework." *Information Sciences* 569 (2021): 786-803.
- [53] Wu, Tingyao, Stefano Petrangeli, Rafael Huysegems, Tom Bostoën, and Filip De Turck. "Network-based video freeze detection and prediction in HTTP adaptive streaming." *Computer Communications* 99 (2017): 37-47.
- [54] Yarnagula, Hema Kumar, Parikshit Juluri, Sheyda Kiani Mehr, Venkatesh Tamarapalli, and Deep Medhi. "QoE for mobile clients with segment-aware rate adaptation algorithm (SARA) for DASH video streaming." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, no. 2 (2019): 1-23.

#### Author's Details:



**Koffka Khan** received the M.Sc., M.Phil., and D.Phil. degrees from the University of the West Indies. He is currently Lecturer in Computer Science and has up-to-date, published numerous papers in journals & proceedings of international repute. His

research areas are computational intelligence, routing protocols, wireless communications, information security and adaptive streaming controllers.



**Leah Joseph** is a past UWI B.Sc. computing student. She worked on TCP and UDP flows in her final year project. Some of the work she did in her project are included in this

paper.



**Emilie Ramsahai** is a Lecturer in Data Science. She is currently working at the UWI. She has more than 20 years industry experience. She has worked with UWI-Roytec in programme development and course writing. She completed her PhD in Statistics and a Masters in Computer Science, both at the University of the West Indies, where she has also lectured the Big Data and Visualisation course from the Masters in Data Science, offered by the Department of Computing and Information Technology, St Augustine Campus. She lectured bioinformatics at the BSc. and MSc. programmes. She also completed her fellowship at the International Centre for Genetic Engineering and Biotechnology (ICGEB) in New Delhi, India and continues to publish and collaborate with a number of researchers in this area.