Int. J. Advanced Networking and Applications
Volume: 10 Issue: 06 Pages: 4054-4060 (2019) ISSN: 0975-0290

What happens when stochastic adaptive video streaming players share a bottleneck link?

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

ABSTRACT

Competition among adaptive video streaming players severely diminishes user-QoE. When players compete at a bottleneck link many do not obtain adequate resources. This imbalance eventually causes ill effects such as screen flickering and video stalling. There have been many attempts in recent years to overcome some of these problems. This work focuses on such a situation. It evaluates current stochastic adaptive video players at a bottleneck link and when the number of players increases. Experimental setup includes the TAPAS player and emulated network conditions. The results show mDASH outperforms x-MDP, sdpDASH and the Conventional players.

Keywords— Adaptive video streaming, bottleneck, flickering, stalling, TAPAS, mDASH, x-MDP, spdDASH

Date of Submission: June 12, 2018 Date of Acceptance: June 11, 2019

I. INTRODUCTION

The player can be in two different phases: (1) Buffering phase: Segments requests are performed back-to-back to quickly fill the playout buffer, and (2) Steady-state: segment requests are spaced to keep the playout buffer level constant. This generates an ON-OFF traffic pattern. It has been experimentally shown that the ON-OFF traffic pattern causes the video flows to obtain a significantly smaller bandwidth share with respect to the fair one when competing with long-lived TCP flows [9], [23]. The downward spiral effect can lead to an even worse degradation of the perceived QoE when concurrent long-lived TCP flows share the bottleneck with the video flow.

It is very challenging to provide satisfactory quality of experience (QoE) during an entire video session [15], [18], [16]. When there are one or multiple players working in a bandwidth rich environment, adaptive streaming works very well. However, in a constrained bandwidth environment with two or more players fair sharing of bandwidth is not achieved. This is due to highly dynamic network conditions which frequently occurs in the real-world, for example, multiple players sharing a bottleneck link. Without an effectual rate adaption algorithm, a DASH client may suffer from frequent buffer underruns, significant quality switches and other QoE related degradation. For example, a buffer underruns can cause flickering, freezing, skipping of video which negatively affects the user QoE. QoE is multi-dimensional and involves the following five metrics:

1. fair share of bandwidth (equal bandwidth allocation),
2. stability (quality switches),
3. buffer over/under-runs (flickering, freezing, skipping),
4. bandwidth utilization (unused bandwidth),
5. quality (low versus high)

Adaptive video streaming in multi-player scenarios sharing bottleneck links poses many user-perceived Quality of Experience (QoE) challenges. Existing solutions still do not adequately address fair sharing of QoE metrics and this work seeks to find better solutions to the research gaps. However, achieving all of these goals simultaneously is intractable and often involves trade-offs between the QoE metrics. Bandwidth-based approaches for adaptive streaming use estimated bandwidth as input to the adaptive streaming algorithm. Buffer-based approaches for adaptive streaming use buffer levels as input to the adaptive streaming algorithm (see Figure 1). Hybrid-based approaches for adaptive streaming combines both features of both bandwidth- and buffer-based approaches. The solution presented in this paper is a bandwidth-based approach.

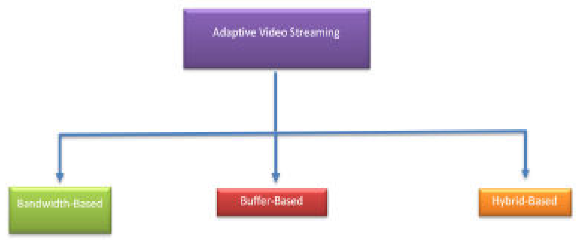

Figure 1: Adaptive streaming approaches.

In adaptive video streaming a Markov Decision Process (MDP) [19] is used to optimize future decisions based on recent history (see Figure 2). A player that has past streaming data can calculate an optimal policy using valid states and actions that move the player between states. This policy would indicate the optimal quality to request from the server for the duration of the streaming session (see Figure 3). However, in present MDP solutions the policy is global and is used by all players [28], [4], [12]. This may result in viewers receiving sub-optimal QoE characteristics. This paper explores current state-of-the-art MDP adaptive video streaming players.

Volume: 10 Issue: 06 Pages: 4054-4060 (2019) ISSN: 0975-0290

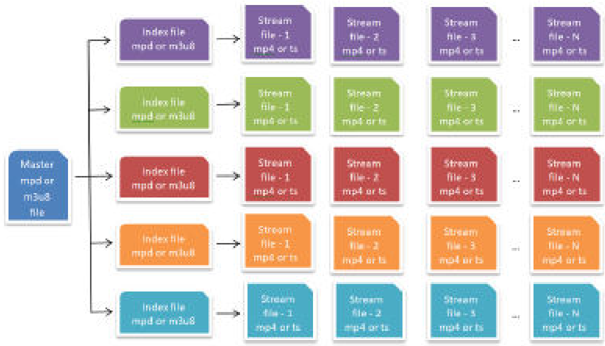

Figure 2: The MPEG-DASH Media Presentation Description

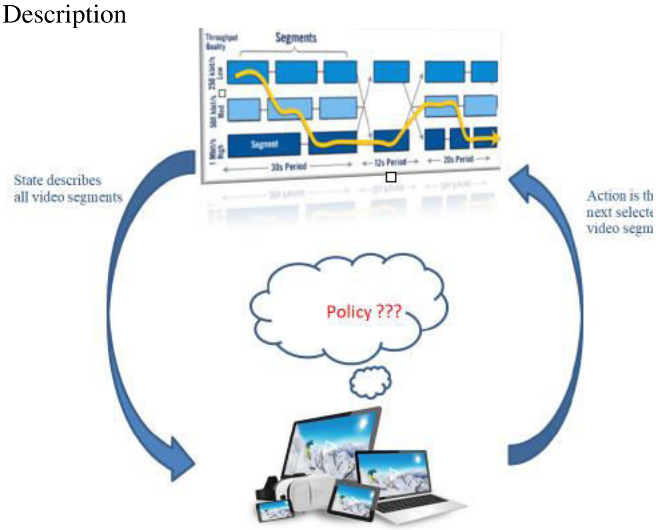

Figure 3: MDP policy determines next segment selection.

The rest of paper is organized as follows. Section 2 explores different adaptive streaming approaches including x-MDP, mDASH, sdpDASH, Conventional players. In Section 3, we look at the experimental setup of the emulations illustrated in this paper. In section 4 we give the results. Finally, we give our conclusion in section 5.

II. Literature Review

During streaming an adaptive video player selects chunks or segments of different quality (see Figure 4). There is a growing body of literature on utilizing Markov Decision Processes (MDP: see Figure 5) to optimize adaptive video streaming [17], [14]. We now outline different MDPs for video streaming. Research shows bandwidth varies severely in different locations [11] and [27]. Thus, they model the adaptive streaming quality selection problem as an MDP problem to cope with varying network conditions [13]. However, to guarantee performance, application of the strategy must be in same network environment. Researchers in [20] model the power consumption problem of video decoding as an MDP to optimize rate adaption, where network uncertainty is high, [24] and [25] model bandwidth as a Markov chain, with its own bandwidth states. The MDP model in [24] aims to find an optimal streaming strategy in terms of user-perceived QoE, such as (1) playback interruption, (2) average playback quality and (3) playback smoothness. The model obtains an optimal MDP solution using dynamic programming. Researchers in [12], [2], [21], [3] and [6] propose stochastic dynamic programming (SDP) approaches for rate adaption in DASH players, where player buffer occupancy and bandwidth conditions determine the system rate. In [22] researchers employ a two-state Markov channel model to analyze buffer underflow-delay trade-offs for adaptive playout strategies. The model with typical parameters reduces the average end-to-end delay by 1 to 2 seconds.

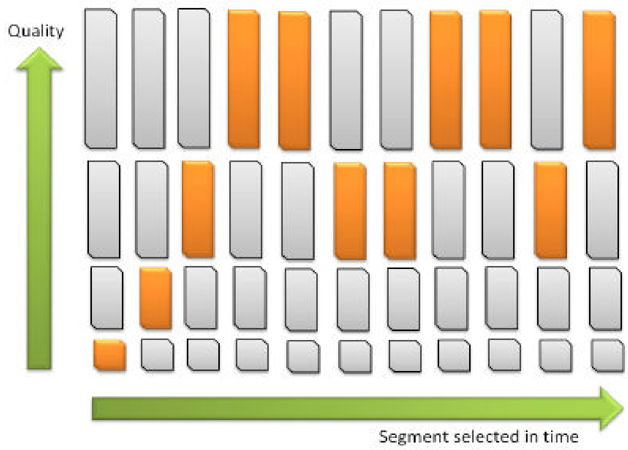

Figure 4: The structure of an adaptive HTTP stream. Rows make up a single quality level of the entire stream. A single box represents a segment of the stream (usually somewhere between 2 to 10 seconds). Segments in the same column represent exactly the same content, but in different encoding bitrates (qualities). The orange colored boxes represent the video playout, indicating the streaming quality the player selects during the streaming session.

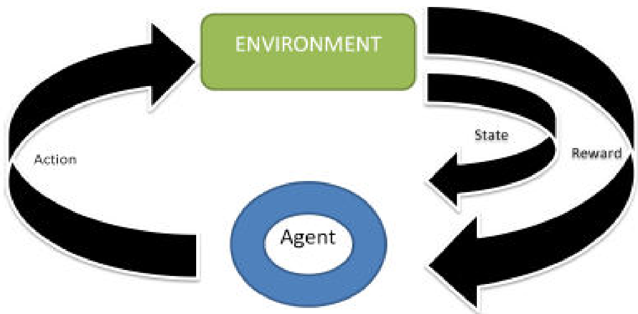

Figure 5: A Markov Decision Process. The agent accesses the state of its environment or surroundings. It observes the actions and re-wards for transitioning to various states. Then, based on the state that the agent is in and the reward for that state, the agent selects an appropriate actions.

The researchers in [26] show how to efficiently and cost-effectively utilize multiple links to improve video streaming quality in mobile networks. Their MDP model utilizes the following parameters: (1) startup latency, (2) playback fluency, (3) average playback quality, (4) playback smoothness and (5) wireless service cost. In addition, they propose an adaptive, best-action search algorithm to obtain a sub-optimal solution. mDASH, [28] aims to maximize the quality of user experience, under time-varying channel conditions. Their MDP parameters includes: (1) video playback quality, (2) video rate switching frequency and amplitude, (3) buffer overflow/underflow and (4) buffer occupancy. However, a trade-off between the overall video quality and continuous playback occurs. Researchers in [8] address the problem of streaming packetized media over a lossy packet network. Data units in a media presentation generally depend on each other according to a directed acyclic graph. However, they reduce the problem of rate-distortion optimized streaming of an entire presentation to the problem of error-cost optimized transmission of a single data unit. The researchers solve the problem in a variety of scenarios, including the important common scenario of sender-driven streaming with feedback over a best-effort network.

The goal of MDP-DASH [4] is to explore different methods of reducing decision making overhead, for DASH-based adaptive video players. The states depend on: (1) quality level of the segment download and (2) time available before segment playback deadline (current buffer occupancy as a measure in time). The action (decision) is the quality level of the next segment download. Higher rewards are given for watching a higher quality segment. There is a penalty for missing a deadline and switching quality from the present segment to the next. For a given action (segment size), calculation of state transition probabilities depends on the Cumulative Distribution Function (CDF) [5] of the network bandwidth. The CDF allows calculation of the probability of a given buffer occupancy, when the next segment download occurs. The buffer occupancy, together with the action (quality level decision), defines the next state. Transition probabilities will change with different CDF's. Different CDFs lead to different MDP strategies.

Adapting video data rate during streaming effectively reduces the risk of playback interruptions due to channel throughput fluctuations. The variations in rate, however, also introduce video quality fluctuations. This potentially affects viewer QoE. Rate adaptation and admission control improves the QoE of video users [7]. A subjective study shows viewer QoE is strongly correlates with an empirical cumulative distribution function (eCDF) of predicted video quality. Consequently, based on this observation, researchers propose a rate-adaptation algorithm incorporating QoE constraints on the empirical cumulative quality distribution per user. Also, a threshold-based admission control policy block users, whose empirical cumulative quality distribution is not likely to satisfy their QoE constraint. Research undertaken in [12] utilizes Stochastic Dynamic Programming (SDP) to model their MDP in order to aid adaptive video streaming users. Researchers use three parameters to compute the state transition matrix: (1) buffer level, (2) average channel bandwidth and (3) quality. A cost function penalizes situations leading to a reduction in QoE. This computation is done offline. The control policies map environment information to player requests. The main result is higher average quality of requests. However, there is an increase in the number of quality switches among segments.

III. EXPERIMENTAL SETUP

The Controller code was written in python. TAPAS [10], an open-source Tool for rApid Prototyping of Adaptive Streaming control algorithms. TAPAS is a flexible and extensible video streaming client written in python that allows researchers to easily design and carry out experimental performance evaluations of adaptive streaming controllers without needing to write the code to download video segments, parse manifest files, and decode the video. TAPAS have been designed to minimize the CPU and memory footprint so that experiments involving a large number of concurrent video flows can be carried out. The player logs experimental data results. The TAPAS player communicates with the video server in the form of a GET request. The Controller has access to a shared table which contains INFO data. The INFO packet is small and thus imposes very low overhead to the BEGGAR protocol. This data is used to calculate Fair share bandwidth and to then request or reduce the player's bitrate request to the server.

A virtual network is setup on the same host machine creating a custom emulation framework (see Figure 6). Our setup consists of client players, video servers, and a bottleneck link. The server resides on a Windows 10 machine. All experiments are performed on a Windows 10 client with an Intel(R) Core(TM)i7-5500U CPU 2.40GHz processor, 16.00 GB physical memory, and an Intel(R) HD Graphics processor. It serves video data to the client(s) who are on a Ubuntu operating system hosted on VMware. The virtual machine is allocated 12GB of physical memory. TAPAS is installed on Ubuntu 15.04 Linux. The TAPAS Adaptive Video Controller client makes different video segment bitrate level requests to the Apache server. TAPAS allow multiple instances of the player to be created enabling multi-client scenarios. This work involves the interaction between adaptive streaming algorithm at the controller and TAPAS players. All traffic between clients and servers go through the bottleneck, which uses VMware settings which allow bandwidth limits to be set during the experiment. TAPAS support both the HTTP Live Streaming (HLS) and Dynamic Adaptive Streaming over HTTP (DASH) format.

The ten-minute-long MPEG-DASH video sequence "Elephant's Dream"[1] is encoded at twenty different bitrates, between 46 Kbps to 4200Kbps and five different resolutions, between 320x240 to 1920x1080, is used to run the experiments (cf. Table II). The video is encoded at 24 frames per second (fps) using the AVC1 codec. Fragment duration of 2s is used and is recorded in the mpd playlist accordingly. All the DASH files (.m4s fragments and .mpd playlists) are placed on the Apache server. We implemented three client-side algorithms in the TAPAS controller. The conventional approach is present by default and is used as a baseline in which to compare against other algorithms. TAPAS is lightweight in built, thus allowing the same receiving host to run a large number of separate video player instances at the same time at different command line interfaces. Thus, it allows the multi-client scenarios which are essential to the work in this paper.

The experiment considers a bottleneck link with two total video connections. The available bandwidth is set to b = 10Mbps for the two player experiments and b = 35Mbps for the increasing players. QoE metrics are described as follows:

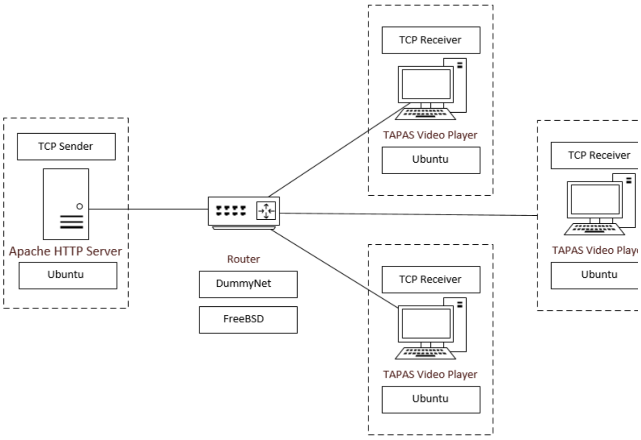

Figure 6: Network testbed setup.

i. The unfairness metric (for two players) is the average of the absolute bitrate difference between the corresponding chunks requested by each player (cf. Equation 5, where p1 and p2 are player 1 and player 2, respectfully). The bitrate is the number of bits required to encode one second of playback.

$$Unfairness = Average\left(\sum_{i=0}^{n-1} \left|r_{i,p1} - r_{i,p2}\right|\right) \quad (5)$$

ii. The utilization metric is defined as the aggregate throughput during an experiment divided by the available bandwidth in that experiment (cf. Equation 6, where tp_i is the throughput at time i and bw is the experimental available bandwidth).

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

In the experiment (E2) the instability, inefficiency, and unfairness (different formulae used for the multi-player scenario) metrics, and re-buffering ratios is used to compare the performances of the considered algorithms.

i. Instability: The instability for player i at time t is given in Equation 7, where $w(d) = k - d$ is a weight function that puts more weight on more recent samples. k is selected as 20 seconds.

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

ii. Inefficiency: The inefficiency at time t is given in Equation 8. Consider N players sharing a bottleneck link with bandwidth, w, with each player x, playing a bit rate, $b_{x,t}$, at time t. A value close to zero implies that the players in aggregate are using as high an average bitrate as possible to improve user experience.

$$Inefficiency = \left|\frac{\sum_x b_{x,t-w}}{w}\right| \quad (8)$$

iii. Unfairness: Let $JainFair_t$ be the Jain fairness index (cf. Equation 10) calculated on the average received rates **Error! Reference source not found.**, r_i, (cf. Equation 9) at time t over all players. The unfairness at time t is defined as $\sqrt{1 - JainFair_t}$. A lower value implies a fairer allocation.

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

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

iv. Re-buffering ratio: is the ratio of the time spent in re-buffering and the total playtime of the stream Equation 11.

$$Re-buffering\ ratio = \frac{total\ re-buffering\ time}{experiment\ duration} \quad (11)$$

We utilize unfairness, inefficiency and instability in our results which is presented in the upcoming section.

IV. RESULTS

We first present the level curves which represent the incoming bitrates of players, see Figures 7, 8, 9 and 10. We observe mDASH with the best level curves among the competing two players. X-MDP does the second best with sdpDASH the third and the Conventional doing the worst. mDASH rate metric which is incorporated into its MDP allows future states to be predicted accurately.

This result is also shown on Figures 11, 12 and 13 is for the increasing players' experiments. mDASH utilizes buffer overflow/underflow and buffer occupancy as adaptation parameters. This allows players to quick recover when network capacity is reduced that is when link underutilization happens.

V. CONCLUSION

Competition among adaptive video streaming players severely diminishes user-QoE. When players compete at a bottleneck link many do not obtain adequate resources. This imbalance eventually causes ill effects such as screen flickering and video stalling. There have been many attempts in recent years to overcome some of these problems. This work focuses on such a situation. It evaluates current stochastic adaptive video players at a bottleneck link and when the number of players increases. Experimental setup includes the TAPAS player and emulated network conditions. The results show mDASH outperforms x-MDP, sdpDASH and the Conventional players.

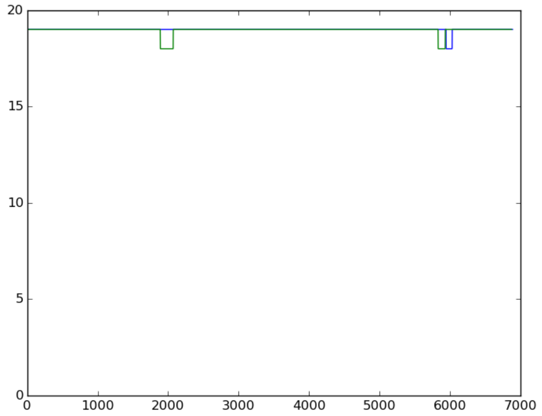

Figure 7: mDASH LEVEL CURVE: BANDWIDTH VARIATIONS

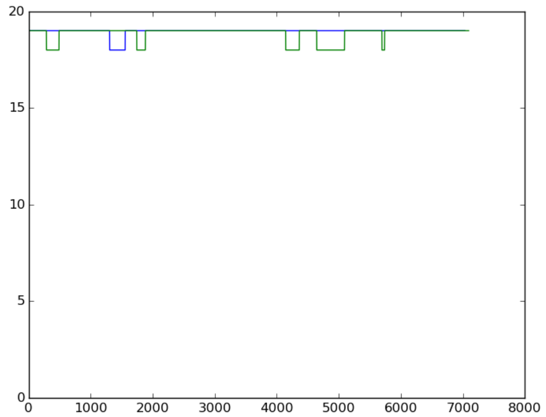

Figure 8: x-MDP LEVEL CURVE: BANDWIDTH VARIATIONS

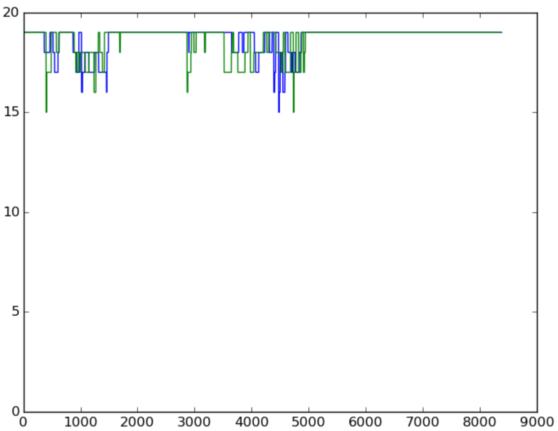

Figure 9: sdpDASH LEVEL CURVE: BANDWIDTH VARIATIONS

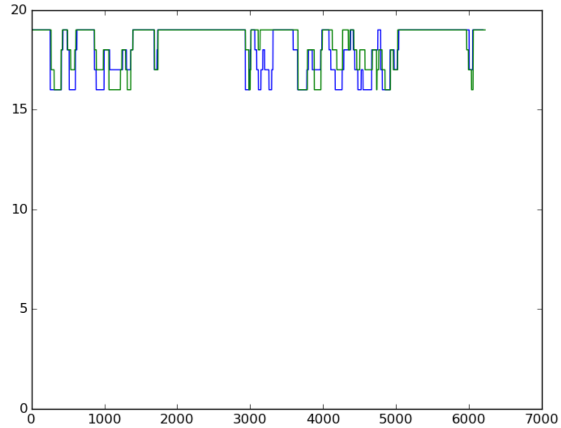

Figure 10: CONVENTIONAL LEVEL CURVE: BANDWIDTH VARIATIONS

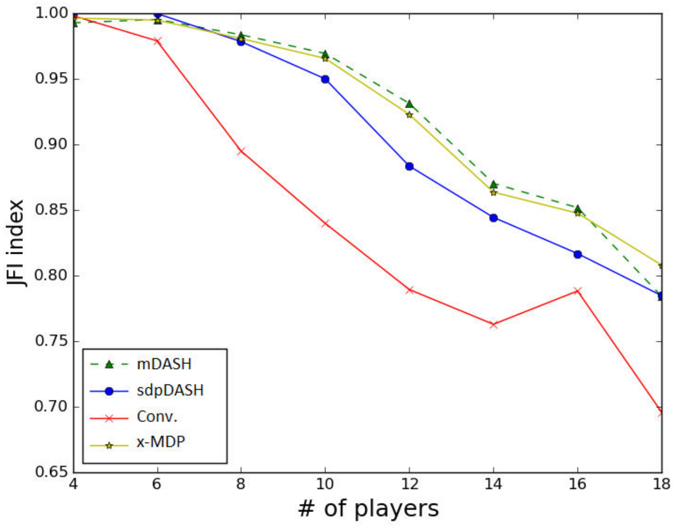

Figure 11: Jain Fairness Index (JFI) for adaptive players.

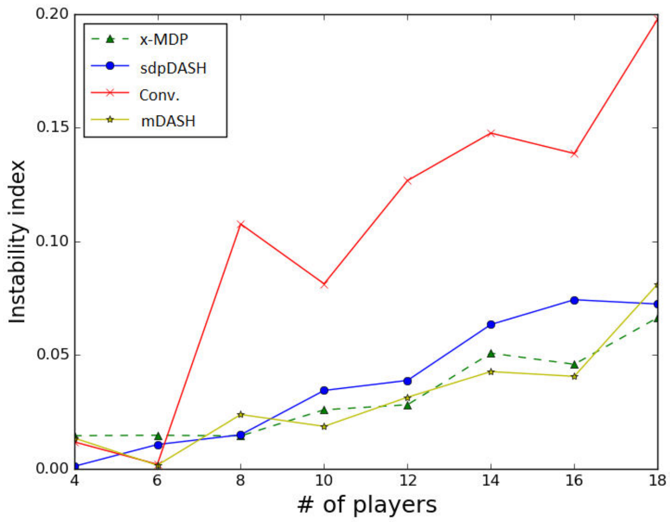

Figure 12: Instability Index for adaptive players.

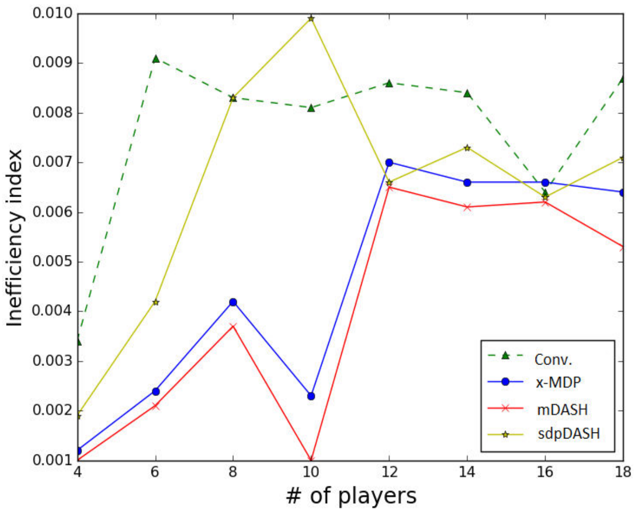

Figure 13: Inefficiency Index for players.

REFERENCES

[1] Akhshabi, Saamer, Lakshmi Anantakrishnan, Ali C. Begen, and Constantine Dovrolis. "What happens when HTTP adaptive streaming players compete for bandwidth?." In Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video, pp. 9-14. ACM, 2012.

[2] Andelin, T., V. Chetty, D. Harbaugh, S. Warnick, and D. Zappala (2012). Quality selection for dynamic adaptive streaming over http with scalable video coding. In Proceedings of the 3rd Multimedia Systems Conference, pp. 149-154. ACM.

[3] Bertsekas, D. P., D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas (1995). Dynamic programming and optimal control, Volume 1. Athena scientic Belmont, MA.

[4] Bokani, Ayub, Mahbub Hassan, and Salil Kanhere. "HTTP-based adaptive streaming for mobile clients using Markov decision process." In Packet Video Workshop (PV), 2013 20th International, pp. 1-8. IEEE, 2013.

[5] Borgonovo, E. (2017). Cdf-based sensitivity measures. In Sensitivity Analysis, pp. 181189. Springer.

[6] Boutilier, C., R. Dearden, and M. Goldszmidt (2000). Stochastic dynamic programming with factored representations. Articial intelligence 121(1), 49-107.

[7] Chen, C., X. Zhu, G. de Veciana, A. C. Bovik, and R. W. Heath (2015). Rate adaptation and admission control for video transmission with subjective quality constraints. IEEE Journal of Selected Topics in Signal Processing 9(1), 22-36.

[8] Chou, P. A. and Z. Miao (2006). Rate-distortion optimized streaming of packetized media. IEEE Transactions on Multimedia 8(2), 390404.

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

[10] De Cicco, Luca, Vito Caldaralo, 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. ACM, 2014.

[11] Deshpande, P., X. Hou, and S. R. Das (2010). Performance comparison of 3g and metro-scale wifor vehicular network access. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, pp. 301307. ACM.

[12] Garcia, Sergio, Julián Cabrera, and Narciso García. "Quality-optimization algorithm based on stochastic dynamic programming for MPEG DASH video streaming." In Consumer Electronics (ICCE), 2014 IEEE International Conference on, pp. 574-575. IEEE, 2014.

[13] Jarnikov, Dmitri, and Tanır Özçelebi. "Client intelligence for adaptive streaming solutions." Signal Processing: Image Communication 26, no. 7 (2011): 378-389.

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

[15] Khan, Koffka, and Wayne Goodridge. "QoE in DASH." International Journal of Advanced Networking and Applications 9, no. 4 (2018): 3515-3522.

[16] 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.

[17] Khan, Koffka, and Wayne Goodridge. "S-MDP: Streaming with Markov Decision Processes." IEEE Transactions on Multimedia (2019).

[18] Khan, Koffka, and Wayne Goodridge. "Variants of the Constrained Bottleneck LAN Edge Link in Household Networks." International Journal of Advanced Networking and Applications 10, no. 5 (2019): 4035-4044.

[19] Marbach, Peter, and John N. Tsitsiklis. "Simulation-based optimization of Markov reward processes." IEEE Transactions on Automatic Control 46, no. 2 (2001): 191-209.

[20] Mastronarde, N., K. Kanoun, D. Atienza, P. Frossard, and M. Van Der Schaar (2013). Markov decision process based energy-e-cient on-line scheduling for slice-parallel video decoders on multicore systems. IEEE transactions on multimedia 15(2), 268278.

[21] Ross, S. M. (2014). Introduction to stochastic dynamic programming. Academic press.

[22] Steinbach, E., N. Farber, and B. Girod (2001). Adaptive playout for low latency video streaming. In Image Processing, 2001. Proceedings. 2001 International Conference on, Volume 1, pp. 962965. IEEE.

[23] Wang, Bing, Jim Kurose, Prashant Shenoy, and Don Towsley. "Multimedia streaming via TCP: An

Volume: 10 Issue: 06 Pages: 4054-4060 (2019) ISSN: 0975-0290

analytic performance study." In Proceedings of the 12th annual ACM international conference on Multimedia, pp. 908-915. ACM, 2004.
[24] Xiang, S., L. Cai, and J. Pan (2012). Adaptive scalable video streaming in wireless networks. In Proceedings of the 3rd multimedia systems conference, pp. 167-172. ACM.
[25] Xing, M., S. Xiang, and L. Cai (2012). Rate adaptation strategy for video streaming over multiple wireless access networks. In Global Communications Conference (GLOBECOM), 2012 IEEE, pp. 5745-5750. IEEE.
[26] Xing, M., S. Xiang, and L. Cai (2014). A real-time adaptive algorithm for video streaming over multiple wireless access networks. IEEE Journal on Selected Areas in communications 32(4), 795805.
[27] Yao, J., S. S. Kanhere, and M. Hassan (2008). An empirical study of bandwidth predictability in mobile computing. In Proceedings of the third ACM inter national workshop on Wireless network testbeds, experimental evaluation and characterization, pp. 1118. ACM.
[28] Zhou, Chao, Chia-Wen Lin, and Zongming Guo. "mDASH: A Markov decision-based rate adaptation approach for dynamic HTTP streaming." IEEE Transactions on Multimedia 18, no. 4 (2016): 738-751.

AUTHOR DETAILS

Koffka Khan received the M.Sc., and M.Phil. degrees from the University of the West Indies. He is currently a PhD student and has up-to-date, 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.

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