

A BIO-INSPIRED HTTP-BASED ADAPTIVE STREAMING PLAYER

Yusuf Sani*, Andreas Mauthe, Christopher Edwards

Mu Mu

School of Computing and Communications
InfoLab21, Lancaster University
Lancaster LA1 4WA, UK
y.sani,a.mauthe,c.edwards @lancaster.ac.uk

Faculty of Arts, Science and Technology
The University of Northampton
Northampton NN2 6JB, UK
mu.mu@northampton.ac.uk

ABSTRACT

In order to streamline video content distribution on a myriad of platforms over heterogeneous networks, HTTP Adaptive Streaming (HAS) is being increasingly adopted. In this paper we pilot a bio-inspired HAS optimisation design with the aim of maximising the overall user experience of a video playback session. Evaluations conducted within a real-world Internet environment, using quality indicators such as convergence time, start-up delay, average video rate, stability, and fairness, demonstrate the benefits of our design.

Index Terms— HTTP Adaptive Streaming, Adaptive Bitrate Selection, Buffer-based Player

1. INTRODUCTION

HTTP-based adaptive video streaming service ensures that the quality of video each user receives is derived from context dependant information, e.g. throughput, buffer level, etc. This requires a client to continuously monitor and estimate the resource availability. It then requests a chunk with the highest video rate the estimated network and device capacity can sustain. This unique mechanism allows HAS video services to adapt to different environments and changes in transmission channel states. The main challenge of designing an adaptation scheme is to seek an optimal balance between a number of QoE factors such as video rate and stability or occurrence of re-buffering. Inspired by the study of population dynamics in biology, this paper introduces a novel HAS player that maximises user experience. Evaluations conducted within a real-world Internet environment demonstrate the benefits of our design with respect to fast convergence, low start-up delay, maximum utilisation, high average video rate, high stability, fairness to other players and video freeze avoidance.

2. BACKGROUND

HAS services provide audio-visual content in one or multiple adaptation sets of representations. Each representation is

*Thanks the Petroleum Technology Development Fund (PTDF), Nigeria for funding.

encoded using a specific encoding configuration (such as bitrate and native frame resolution). A representation of content is stored in a series of chunks (segments). Using a manifest file such as Media Presentation Description (MPD), a HAS client progressively requests chunk of a certain bitrate, from the selected representation, based on the level of available resources measured at the client side. Such a streaming method is also known as *adaptive bitrate selection* (ABR). The first generation of ABRs select a chunk with the bitrate just below the estimated throughput derived from a predefined number of previously downloaded chunks [1, 2]. When there is a change of the estimated throughput, the media playback buffer level can be used as an indicator to determine whether to increase, decrease or keep the current quality level in order to maximise the playback quality whilst avoiding buffer under-run (i.e., re-buffering) [3, 4]. Many of the current HAS services are found to suffer from unnecessary re-buffering [5], undesirable fluctuations [6] and sub-optimal video quality [6]. Instead of solely relying on heuristic, it is shown in [7] that incorporating a system model that captures the relationship between buffer state changes and video rate can improve the QoE performance of a HAS player. Based on this observation we were motivated to take a cross-disciplinary approach towards the QoE modelling of ABR algorithms.

Population dynamics is a branch of mathematical ecology that is mainly concerned with how the population of species grows and shrinks as well as the processes that influence the changes e.g. resource and competing specie [8]. This paper focuses on the most well-known population model, the *Verhulst-Pearl logistic model*. The model is very well understood and is known to be stable and converges at the maximum carrying capacity. We then propose a buffer-based ABR algorithm that uses the Verhulst-Pearl equation to drive its rate evolution map.

3. VERHULST-PEARL BASED VIDEO RATE MAP

The video quality level is assumed to be a specie whose growth we are interested in, and the playback buffer is considered to be its habitat. We assume that the rate at which the incoming video chunks arrives is analogous to birth rate and the rate at which a player consumes content from the play-

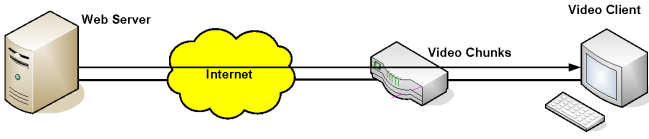


Fig. 1. Experimental Set-up

back buffer is assumed to be the death rate. It is known that the rate of chunk arrival is $\frac{c(t_i)}{Q}$ [5], where $c(t_i)$ is the throughput of a link and Q is bitrate of the chunk being downloaded. Please note that the consumption rate is constant at one second of content every wall-clock second. Therefore, buffer level changes B_t at any time t depends on these two parameter. AS in a natural habitat, the buffer size will determine a limit on the maximum video rate q_{max} a player can download. In our case, unlike in the wild, the maximum video rate is given (as defined in the MPD), therefore our task is focused on identifying the amount of the buffer required to guarantee the maximum video rate. We now modify the Verhulst-Pearl equation thus as:

$$\frac{dQ}{dB} = \alpha q(q_{max} - q) \quad (1)$$

Equation (1) defines the only valid system state-space, which is represented by the rate at which video rate changes with respect to buffer state. With α being the growth constant and q the requested video rate. To derive the rate map we solve the differential equation (1) assuming that, like in a natural context, there is a seed population that reproduces to kick-start the growth. We call this the minimum video quality level q_0 .

$$Q = \frac{q_{max}}{1 + \left[\frac{q_{max}}{q_0} - 1\right]e^{-\alpha q_{max} B_t}} \quad (2)$$

It is easy to show that equation (2) will always converge at the maximum video rate, that is, $\lim_{B \rightarrow \infty} Q = q_{max}$ [7].

3.1. Implementation

A HAS player chooses video representations from a finite and discrete set. We first restrict the player to switch between adjacent video rates only so as to prevent high amplitude variations.

Assuming q_k has been completely downloaded (please recall that we are starting with (q_{min}) , we can use equation (3) to calculate the buffer distance needed to change video rate. Once $\Delta B_{k+1} = 0$ we move up the ladder of video rates. If $\Delta B_{k-1} = 0$, the video rate is switched down, else we maintain the current video rate.

$$\Delta B_{k\pm 1} = \begin{cases} Q^{-1}(q_{k+1}) - B_t & q_{k+1} \\ B_t - Q^{-1}(q_{k-1}) & q_{k-1} \end{cases} \quad (3)$$

4. PERFORMANCE EVALUATION

The experimental set-up is shown in Figure 1. The client is connected to the Internet either via an Ethernet switch or EE's 3G networks in the UK. We use the web services located at

the Alpen-Adria-Universität, Klagenfurt, which hosts the Big Buck Bunny dataset [9]. The player is implemented in Python and uses Request package for HTTP request-response transactions, and is tested on Ubuntu 12.04.2 LTS with 3.8 kernel. The host that runs the player also hosts *dummynet*, *tcpdump*, *lsof*, and *wget* as part of the experiment.

We conducted a "blue-sky" test in both wired and wireless setting. Then varies the link capacity using *dummynet*. We set $B_{max} = 200s$, and the the growth constant $\alpha = 0.05$ is used. Each experiment was conducted 10 times and the average result is used. When more than one player is used or when a player operates in the presence of a background traffic, the experiment is conducted on the same machine for the maximum portability.

4.1. Evaluation Metrics

There is a growing number of research in the field of QoE, and no definite model has so far been established for adaptive streaming [10, 11]. To evaluate the performance of our model, we employ the following metrics which are widely recognised as the key QoE indicators:

- **Re-buffers:** this is the total number of video freeze event per streaming session.
- **Average video rate:** is calculated as $\frac{t_1 q_1 + t_2 q_2 \dots t_n q_n}{t_n - t_1}$ and measured in *kb/s* [12].
- **Stability:** measures the degree of video quality oscillation as described in [13].
- **Utilisation of available network resource:** is calculated by dividing the average video rate by the average network capacity [14].
- **Fairness:** two or more services are said to be fair if they divide the available resource among themselves equally.
- **Convergence time:** is the time taken to settle at the sustainable video rate.
- **Startup Delay:** in this paper we define startup delay as the amount of time it takes for a player to download 15 chunks before the playback starts.

5. RESULTS

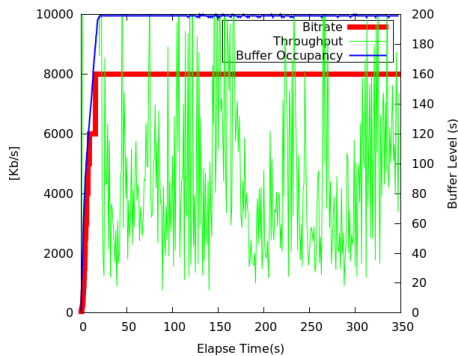
This section discusses the result of the various test-bed experiments conducted.

5.1. Startup Period and Stability

In the first set of the experimentation we stream in both the wired and wireless environments with the network link as is. As can be see from Figure 2 the player is able to converge at the maximum video rate in a very short time and without a single instance of oscillation within the wired environment. Furthermore, the start-up delay is minimal at 1.19s. A similar performance is obtained when the player is run in a wireless environment (see Table 1 the summary). It's worth noting that the player seems to be more cautious in a risky environment, delaying the video rate upgrade in order to maximises

Table 1. Adaptation for Variable Bandwidth

Environment	Startup Delay (s)	Convergence Time (s)	Maximum Rate (kb/s)
Wired	1.19	16	8000
Wireless	1.25	91	8000

**Fig. 2.** Blue Sky Test (Wired Network)

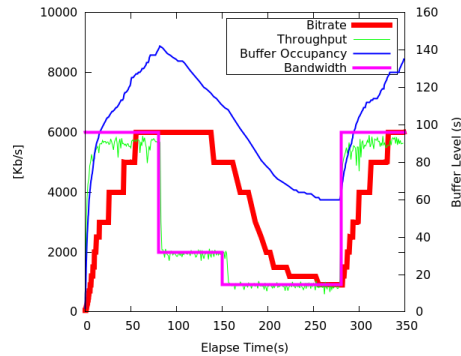
stability. As can be seen from Figure 4(b) there are only three (3) instances where the player changes its video rate, with the oscillation factor of only 2.0%, even though the network is anything but stable as can be observed from Figure 4(a).

5.2. Responsiveness and utilisation

Next, we investigate how the propose player adapts in a network with severe bandwidth fluctuation. Figure 3 clearly demonstrates that the player is able to converge at exactly the system capacity regardless of whether it is an upward or downward convergence. Furthermore, when the link capacity suddenly decreases, the player does not instantly over-react like many other HAS player designs, rather because it senses the buffer can sustain high video rate it takes a modest approach, and when it starts reducing the video rate it does that linearly. As is summarised in Table 2, the design maximises the utilisation of link capacity at 106% without causing any video freeze or resulting in buffer dropping to a risky level. This confirms the fact that with careful design, a player can download above the link capacity, provided that some content is pre-buffered, without affecting its QoE performance.

5.3. Fairness

Also investigated is how fair the player is to other HAS player and background TCP traffic. A total of four players, all using the same implementation, were run concurrently. We set the maximum bandwidth to $6Mb/s$, the hypothesis is that if the players are fair to one another they should equally share the available bandwidth (because all the players are connected to the same network and are also running on similar device). As can be seen from Figure 5(a) none of the players secures more than $1.5Mb/s$, a fair share of the available capacity. This is

**Fig. 3.** Responsive and Utilisation test**Table 2.** Adaptation for Variable Bandwidth

Rebuffers	Maximum Video rate (kb/s)	Average Video rate(kb/s) (%)	Throughput Utilisation (%)
0	6000	3827	106

achieved with a high level of stability, as none of the players observed more than three unnecessary oscillation of the video rate.

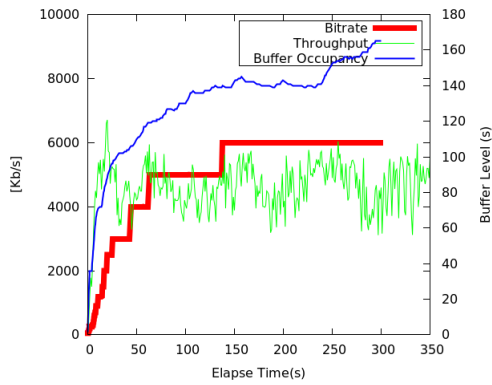
Next, the background traffic (file downloading from the same server) is started 30s after the start of the streaming session. As can be seen in Figure 5(b), as soon as the background traffic is started the video quality starts to gradually drop until an equilibrium is reached. They fairly share the available bandwidth, that is, each uses about $3Mb/s$. Furthermore, it is worth noting that the drop in video rate does not affect the stability of players, start-up delay or other QoE factors.

6. CONCLUSIONS

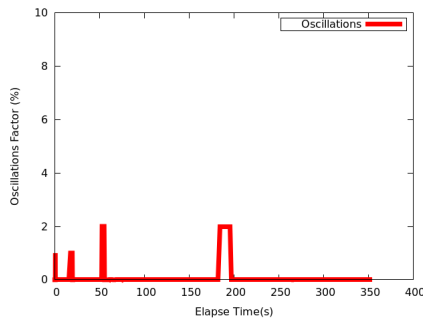
HTTP adaptive streaming provides a great foundation for online media streaming over heterogeneous networks. Recent years have seen an increasing amount of efforts in the development of adaptation algorithms with the shared objectives of maximising the quality of user experiences. This paper pilots a new approach that balances multiple key quality factors using a bio-inspired mechanism. Evaluation results demonstrate the effectiveness of the proposed design with respect to start-up delay, convergence time, stability, network efficiency, average video quality and fairness.

7. REFERENCES

- [1] Saamer Akhshabi, Ali C. Begen, and Constantine Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proc. of the MMSys*, 2011, pp. 157–168.
- [2] Truong Cong Thang, Quang-Dung Ho, Jung Won Kang, and Anh T Pham, "Adaptive streaming of audiovisual content using mpeg dash," *Consumer Electronics, IEEE Transactions on*, vol. 58, no. 1, pp. 78–85, 2012.

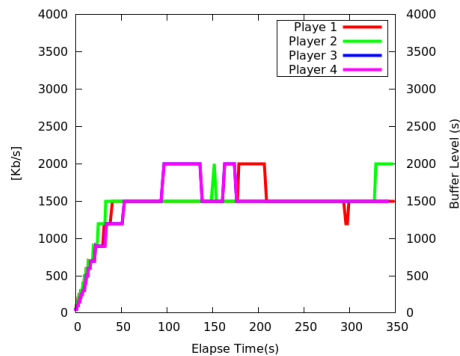


(a) Blue Sky Test (wireless)

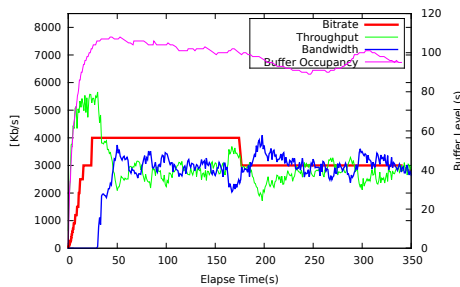


(b) Player Oscillation

Fig. 4. Four players streaming at the same time.



(a) Four Players streaming at the same time.



(b) Player and Background TCP traffic.

Fig. 5. Fairness test

- [3] Saamer Akhshabi, Lakshmi Anantakrishnan, Ali C. Begen, and Constantine Dovrolis, “What happens when http adaptive streaming players compete for bandwidth?,” in *Proc. of the NOSSDAV*, 2012, pp. 9–14.
- [4] Junchen Jiang, Vyas Sekar, and Hui Zhang, “Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive,” in *Proc of the 8th International Conference on Emerging Networking Experiments and Technologies*. ACM, 2012, pp. 97–108.
- [5] Te-Yuan Huang, Ramesh Johari, and Nick McKeown, “Downton abbey without the hiccups: Buffer-based rate adaptation for http video streaming,” in *Proc. of the FhMN*, 2013, pp. 9–14.
- [6] Te-Yuan Huang, Nikhil Handigol, Brandon Heller, Nick McKeown, and Ramesh Johari, “Confused, timid, and unstable: Picking a video streaming rate is hard,” in *Proc of the ACM IMC*, 2012, pp. 225–238.
- [7] Yusuf Sani, Andreas Mauthe, and Christopher Edwards, “Modelling video rate evolution in adaptive bitrate selection,” in *The IEEE International Symposium on Multimedia (ISM 2015)*, Dec 2015.
- [8] John Pastor, *Mathematical ecology of populations and ecosystems*, John Wiley & Sons, 2011.
- [9] Stefan Lederer, Christopher Muller, and Christian Timmerer, “Dynamic adaptive streaming over http dataset,” in *Proc. of the MMSys*, 2012, pp. 89–94.
- [10] Chao Chen, Lark Kwon Choi, G. de Veciana, C. Caramanis, R.W. Heath, and AC. Bovik, “A dynamic system model of time-varying subjective quality of video streams over http,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 3602–3606.
- [11] Lark Kwon Choi, Yiting Liao, and Alan C Bovik, “Video qoe models for the compute continuum,” *E-LETTER*, 2013.
- [12] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang, “Understanding the impact of video quality on user engagement,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 362–373, 2011.
- [13] Christopher Mueller, Stefan Lederer, Reinhard Grandl, and Christian Timmerer, “Oscillation compensating dynamic adaptive streaming over http,” in *Multimedia and Expo (ICME), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–6.
- [14] Guibin Tian and Yong Liu, “Towards agile and smooth video adaptation in dynamic http streaming,” in *Proc. of the CoNEXT*, 2012, pp. 109–120.