2021CS3570 Multimedia Final Project Report

Group 5 : Challenge1 Live Video Streaming

Szuyu Lee
Department of Computer Science
National Tsing Hua University
Hsinchu Taiwan
s106062328@m106.nthu.edu.tw

Songen Chen
Department of Computer Science
National Tsing Hua University
Hsinchu Taiwan
s106062128@m106.nthu.edu.tw

YuChi Su
Department of Computer Science
National Tsing Hua University
Hsinchu Taiwan
s106000228@m106.nthu.edu.tw

ABSTRACT

In this final project of Introduction to Multimedia Class 2021 spring, National Tsing Hua University, we chose the topic of the Live Video Streaming Problem by adapting and designing Adaptive Bitrate Algorithm.

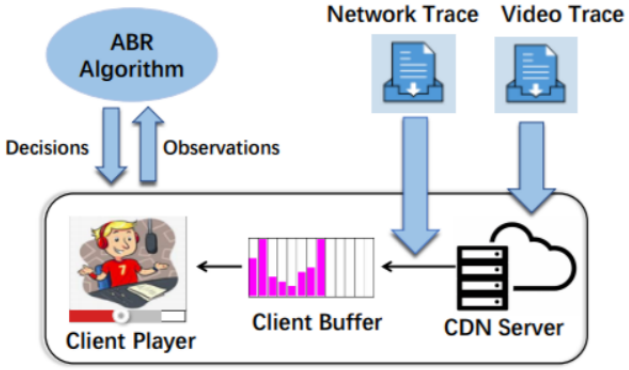
Our project is based on the testbed provided by The ACM Multimedia 2019 Live Video Streaming Grand Challenge[1], which was further modified by Teaching Assistants of this class. Our task is to obtain high QoE(Quality of Experience) value while maintaining low penalties.

We referenced and implemented BBA[2], BOLA[3], and HYSA[4] algorithms to find out a better hybrid solution. HYSA is generally better with it's prediction schemes. Some of the modules are found having a negative impact on total performance, thus we remove and modify the HYSA structure. The QoE improvement compared to the original BBA algorithm and original HYSA is significant, respectively 100% and 15%.

1 Introduction

Live Video Streaming has become increasingly popular in the past 10 years. Viewers are raising higher demands on the quality of real-time video. Engineers are trying to provide better video transmission methods with support of improved networks and development of new algorithms.

Figure 1: **model of the video streaming simulator**



1.1 Simulator with Metrics and Parameters

As Figure 1 shows, the code structure from Github mainly models a simulator, simulating interactions between client video players and CDN servers. There are 2 bunches of dataset offered. Network Traces and Video Traces. They describe the last-mile network condition and the transmission of each video frame respectively. The experiences of viewing quality of the client player can be measured by parameters from run.py such as buffer_size, data_size, play_time, skip_time, chunk_size, latency.

QoE is the most important value in our project

$$QoE = \sum_{n=1}^{N}\sum_{m=1}^{M}(\bar{\beta}R_{n,m} - \gamma T_{n,m} - \delta L_{n,m} - \theta S_{n,m}) - \sum_{n=1}^{N-1}\bar{\alpha}|R_{n+1} - R_n|$$

Our task is to obtain low QoE after running the video streaming simulator. As the formula above from [1], QoE contains, another paper [4] explains the QoE better according to the Contest code, formula listed as following:

$$\begin{aligned} QoE &= QoE_{quality} + QoE_{rebuf} + QoE_{latency} + QoE_{skip} \\ &\quad + QoE_{switch} \\ &= \sum_{k=1}^{K}(p_q V_k d_f - p_r t_k^r - p_l l_k - p_s t_k^s - p_w |V_k - V_{k-1}|) \end{aligned}$$

There are one positive contributor QoE_quality and four negative contributors(also known as penalties):

Table 1: **penalty list**

penalty_name	penalty_constant
rebuffering	1.85
latency	0.005 or 0.01
skip_frame	0.5

switch_rate	0.02

Aside from bitrate(positive contributor), rebuffering is considered the most important to deal with since the penalty is comparatively high. Another important penalty to be noted is latency since latency happens very often, nearly in every video frame. Besides, latency is affected by playback rate so the penalty constant is not stable. As a result, Bitrate Control and Latency Control (PlayBack Speed Control and Frame Skipping Control) are core ideas when following algorithms are designed.

1.2 BBA & BOLA

1.2.1 BBA0

BBA0 is an algorithm proposed by [2], the main idea of BBA0 is increasing the bitrate only when the buffer has enough content to play. Considering the bitrate is discrete, and also the variation of network capacity, BBA0 suggested the use of reservoir and cushion as in Figure 2.

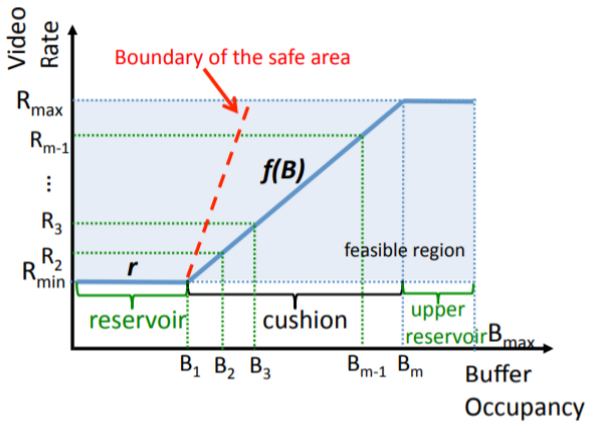


Figure 2 : From [2], shows the concept of reservoir and cushion.

When buffer occupancy is in the reservoir, BBA0 only requests the minimal bitrate. When buffer occupancy is in cushion, we take higher and higher bitrates. Reservoir provides a safe space to prevent rebuffering. Cushion provides the chance of reaching the maximal bitrate before the buffer is full.

1.2.2 BBA1

[2] also introduced BBA1, which means to deal with the variable bitrate. Compared to BBA0, the main difference is variable reservoir and chunk map. The variable reservoir is computed with how much buffer the incoming data will consume or resupply. The chunk map using actual chunk size incoming instead of bitrate we selected.

1.2.3 BBA2

BBA2 is an enhanced BBA1, it takes the risk to raise the bitrate faster in the reservoir. It also utilizes network capacity estimation to prevent draining the buffer too fast. Only when the video accumulates is faster than video consumed in the buffer, BBA2 will choose higher bitrate faster.

1.2.4 BOLA

BOLA is an algorithm proposed by [3]. It formulate bitrate adaptation as a utility maximization problem that incorporates both key components of QoE. Since BOLA is buffer-based, it avoids the overheads of more complex bandwidth prediction present in current video player implementations and is more stable under bandwidth fluctuations.

The main idea of BOLA is to maximize the formula: **(V*υm + V*γ*p − Q)/Sm.** According to the formula, we can see that there are some variables we need to know.

1. Chunk length: p
2. Utility(user's experience): vm
3. Number of chunk in current buffer: Q
4. Buffer level:Q*p
5. How strongly we want to avoid rebuffering:r
6. Sm = current bitrate*chunk length

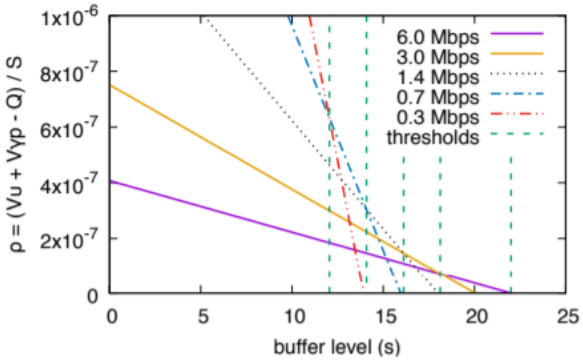


Figure 3: From [3], The value of **(V υm + V γp − Q)/Sm** for different bitrates depends on buffer level. (γp = 5 and V = 0.93.) Note that the buffer level is Qp seconds.

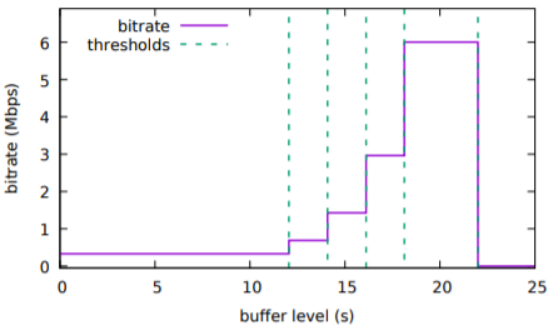
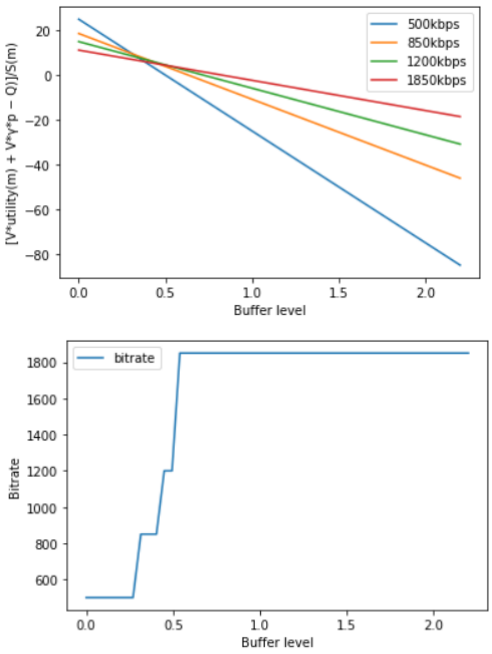


Figure 4: From [3], BOLA's bitrate choice as function of buffer level. (γp = 5,V = 0.93.) Note that the buffer level is Qp seconds.

As you can see in Figure 3, when the buffer level is about 20, the result is higher when the bitrate is 6.0Mbps. Similarly, when the buffer level is about 11, the result is higher when the bitrate is 0.3Mbps. That is to say, when the buffer size is bigger, we will choose the higher bitrate.



2. Implementation

2.1 BBA

We implemented the pseudo code in [2]. Choosing bitrate according to the current buffer occupancy, reservoir and cushion. The reservoir, cushion and latency limit are set as constants, which are determined by experiments. The selection of target buffer is based on [4].

2.2 BOLA

Bitrate control

The table is the information according to this project.
v(utility) = ln(S(m)/min(S))

m	0	1	2	3
bitrate (Mbps)	1.85	1.2	0.85	0.5
S(Mb)	0.074	0.048	0.034	0.02
v	1.3	0.87	0.53	0

The figure is the result based on this project. We can see that when the buffer level is 2.0, we choose the bitrate 1850kbps can maximize the formula. Similarly, when the buffer level is about 0.2, the bitrate 500kbps can maximize the formula. The result is the same as [3].

Latency limit

From [3], it did not mention about how to set the latency limit. Thus, we use the fixed value to implement.

Target Buffer

From [3], it did not mention about how to set the target buffer. However, We have an idea. We think that if we choose the high bitrate to encode, it represents that the current buffer size is high. We tend to choose target buffer[0] because its quick play bound is smaller than target buffer[1]. In this way, it's easier to increase the play rate.

2.3 HYSA

The spirit of HYSA is estimating the latency and choose the bitrate with lowest latency subjected to estimated buffer occupancy is higher than a threshold B_{th}. The estimation starts from throughput, same as [4], we take weighted average of past N_{WA} throughput as future throughput. [4] also adopt Kaufman's Adaptive Moving Average (KAMA) as their estimation of next segment chunk size. Then utilizing these two information to calculate downloading time of the next segment. Afterward, from the above information we can estimate buffer occupancy, and frame accumulation speed in CDN. Finally, we estimate latency using previous estimation above.
For target buffer selection, after considering the boundary of playback rate for target buffer = (0, 1). Using target buffer =

1 when buffer size between B^0_{min} and B^0_{max}. Using target buffer = 0 in other conditions.
For latency limit, we take into consideration the benefit of skipping frames(reduce latency penalty) and drawback (skip penalty and bitrate penalty). Also, a parameter λ to adjust the importance of benefits and drawbacks.

3. Experiment

(Use video_trace: game fixed as example)

(1)Different Latency Limit Prediction vs Constant Limit

	Prediction	Constant 1.8	Constant 1.6	Constant 1.4
QoE	2785	2787	**2795(peak)**	2791

Thus, we decide to remove the frame_dropping part which described in hybrid scheme to calculate latency_limit

(2)Different WMA_len (use game_high for comparison)

	WMA_len=1	WMA_len =5	WMA_len = 10
QoE	**4361**	4245	4168

(3)Different WMA_len (use game_low for comparison)

	WMA_len=1	WMA_len =5	WMA_len = 10
QoE	**1689**	1589	1559

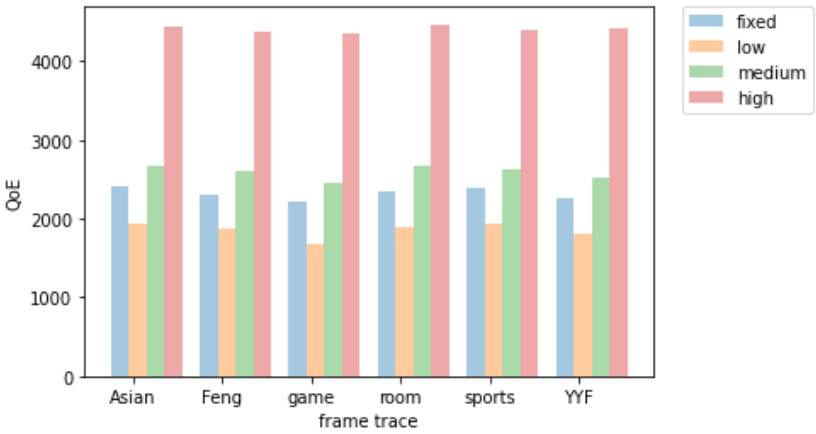
Since WMA_len is best when equals to 1, the actual contribution of weight moving average is considered negative in our case.

4. Overall Result

Table 2 : result of different algorithm

Algorithm	QoE	Run time / Tracecount	Improvement
BBA (Original)	1.41261182e+03	5.06769108e-06	0%
BBA0	1.85089028e+03	6.46227804e-06	31%
BOLA	1.48272377e+03	8.41729989e-06	5%
HYSA (Paper)	2.42404e+03	N.A.	N.A.
HYSA (Modified)	**2.79505720e+03**	3.75153645e-05	**(ToBBA) 98% (ToHYSA) 26%**

Figure 6 : QoE of modified HYSA for every frame traces



Therefore, our code over performs BBA and HYSA in any cases significantly. Numerically outperform BBA 98% and HYSA 26%. Therefore, we adopt the modified HYSA as the final result.

APPENDIX

Questions from classmates

1.target buffer跟latency limit的部分呢?
The target buffer selection scheme we used takes [4] as reference. The determination of latency is also based on [4]. The detailed implementation is in section 2.

2.Target buffer 的選擇理論上根據 bitrate 對 target buffer 進行動態調整，但是實際上的數據在這樣動態選擇跟固定 target buffer 為 0 做比較下，真的對 QOE 有改善嗎?
In our experiment, when we choose a target buffer according to buffer occupancy, the QoE score is higher than always set to 0.

3.在選擇target buffer時，你們目前是根據bitrate的大小去做選擇，那麼你們是如何確定這樣子的分法是適合的呢？(ex，根據實驗結果or根據paper方法實作)
Our reference didn't mention how to set the target buffer. Thus, we did some experience to get the result.

4.想問你們在BOLA演算法裡面的p是什麼?
p: Chunk length

5.想請問utility這個參數是如何定義以及如此定義的原因
The definition of utility is user's viewing experience. We take a paper for reference and it said that utility=ln((current bitrate*p)/(min(bitrate)*p)). I think the idea of this formula is using the bitrate to formulate user's viewing experience. If the bitrate is higher, the utility will be higher.

6.有提到 高bitrate就用target buffer[0], 反之用target buffer[1]。但是targetbuffer住要是用來控制播放速率，但是bitrate似乎跟這裡沒有太大關係? 當buffer很空時候，如果又對到播放速度加快，似乎會成反效果?
We think that if we choose the high bitrate to encode, it represents that the current buffer size is high. We tend to choose target buffer[0] because its quick play bound is smaller than target buffer[1]. In this way, it's easier to increase the play rate.

7.想問有甚麼可改進地方嗎?
At the moment of our presentation, we set the latency limit as a constant. But we can use dynamic determined latency limit for better QoE.

8.因為在時間上的限制，好像BBA1跟BBA2並沒有太多介紹，希望可以在report上補充，也想請問為何後面是使用BBA0跟BBA1去跟BOLA比較?
See section 1.2.2 and 1.2.3. We compared BBA with BOLA because they are both buffer-based ABR algorithms.

REFERENCES

[1] Yi, G., Yang, D., Bentaleb, A., Li, W., Li, Y., Zheng, K., ... & Cui, Y. (2019, October). The acm multimedia 2019 live video streaming grand challenge. In Proceedings of the 27th ACM International Conference on Multimedia (pp. 2622-2626).

[2] Huang, T. Y., Johari, R., McKeown, N., Trunnell, M., & Watson, M. (2014, August). A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In Proceedings of the 2014 ACM conference on SIGCOMM (pp. 187-198).

[3] Spiteri, K., Urgaonkar, R., & Sitaraman, R. K. (2020). BOLA: Near-optimal bitrate adaptation for online videos. IEEE/ACM Transactions on Networking, 28(4), 1698-1711.

[4] Peng, H., Zhang, Y., Yang, Y., & Yan, J. (2019, October). A hybrid control scheme for adaptive live streaming. In Proceedings of the 27th ACM International Conference on Multimedia (pp. 2627-2631).