ABR in Video Streaming

Hao-Wei Chung, Jing-Hao Chiu, Jung-Cheng Chen
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

***Abstract*—Compared with on-demand streaming, live streaming is more challenging since the need for low latency, high video quality, and the influence of playback. Many adaptation systems have been introduced to make a balance between different metrics while maximizing the QoE score. With the help of some algorithms that take latency under consideration, for instance, frame dropping and adaptive playback rate control, bitrate adaptation becomes the significant resolution for video streaming under fluctuating network situations.**

***Index Terms*—Multimedia streaming, ABR algorithm, Latancy control.**

I. Introduction

Currently, there has been a great increase in live streaming applications. Live streaming has stricter latency limitations, compared with on-demand streaming. Therefore, it is a great challenge to lower the latency while keeping the video quality high. Many people harness ABR algorithms such as BOLA and BBA to improve the QoE with a fluctuating network condition. However, most ABR algorithms do not consider latency. Frame dropping and playback rate control are deemed as the silver bullet to lower latency in live streaming. As a result, many previous works employed frame dropping and playback rate adaptation to low the latency. Though so many methods were proposed to lower the latency, we still need to take both latency and video quality under consideration while keeping an eye on QoE for the whole picture. In our term project, we reimplement the algorithm in the previous work, which combines playback rate control, bitrate control, and frame dropping control. Firstly, we adjust the playback rate according to the buffer. Second, we use a latency-constraint algorithm to determine the bitrate. The result shows that this hybrid scheme defeats many previous designs' QoE.

II. Method

A. HYSA

1) Playback Rate Control(target buffer): To find out the target buffer and the latency limit for the higher QoE score, we used the HYSA from the "A Hybrid Control Scheme for Adaptive Live Streaming", which is an approach that it can find out suitable target buffer with its "Playback Rate Control" module, and the latency limit with its "Frame dropping Control" module. In playback rate control module, it uses target buffer for the purpose of helping control playback rate. The target buffer is marked as B_{target}, it represents the value of target buffer occupancy, this value will resume playback if interrupt occur, $[B_{min}, B_{max}]$ is a buffer interval, represents player can use normal playback rate to play buffered video. So, these terminals can be marked as $[B_{min}, B_{target}, B_{max}]$, which means $B_{min} < B_{target} < B_{max}$. In this challenge, the target buffer is set to 0 or 1, satisfying $B^0_{min} < B^1_{min} < B^0_{target} < B^1_{target} < B^0_{max} < B^1_{max}$. By the conclusion in the paper, we set the next target buffer as 1 if current buffer occupancy B_n is in the range $[B^0_{min}, B^0_{max})$, otherwise it would be set as 0. In our experiment settings, we set B^0_{min} as $B_0/2$, and round it, which is 0.2 and the B^0_{max} as $B_0 * 2$, which is 1. We also did some experiments and changed the value of the bound, finally, we noticed that when $B^0_{min} = 0.4$, $B^0_{max} = 0.55$ we can get the best result in our experiments.

2) Frame Dropping Control(latency limit): As for latency limit, we reference the Frame Dropping Control module in the paper. Base on the paper's equation(19), we can find out the latency limit of next iteration. In the below figure we show the equation of it, and the term $p_q V_{n+1,quality_{n+1}}$ is selected bitrate in this iteration, p_s is skip penalty, d_f is the time length of each frame, p_d is latency penalty and λ is the variable which defined by us.

$$latency\_limit_{n+1} = \frac{(p_q V_{n+1,quality_{n+1}} + p_s) d_f}{p_d \lambda}$$

Fig. 1. The equation(19) of the paper

B. Bit rate experiment

We compare the leaderboard in ACM Multimedia 2019 Grand Challenge website with the overall QoE of "A Hybrid Control Scheme for Adaptive Live Streaming" paper. We found the QoE of HYSA is 2.424K, but all first half in the leaderboard are higher than 2.6K. Besides, We think there is a problem with the bit rate part of HYSA. In this paper, they use too much prediction to calculate bit rate, such as segment, throughput. Therefore, it sometimes doesn't choose the most proper bit rate.

At first, we try to use buffer-based methods, such as BBA, Bola to combine with our target buffer and latency parts. However, the QoE doesn't reach our expectations. In BBA part, the QoE is 1.503K; In Bola part, the QoE is 1.420K. QoE of both methods are lower than origin bit rate method. By the above methods, we get the conclusion that the lack of throughput factor will lead to terrible results. Therefore, we ought to take throughput into considering.

To avoid the predictive situation, we use the last chunk to calculate throughput. Throughput is equal to send data time of the last chunk divided by the total data size of the last chunk. Then, We try to get the relation between bitrate and throughput. There is a concept easy to understand when the throughput value is higher, bitrate value can be higher. On the other hand, when the throughput value is lower, the bitrate value should be lower. We use this concept to fulfill bitrate prediction.

We compare the remaining buffer size with the value of a constant K multiples bitrate and divides throughput. Then, Use the result of the comparison to decide the bitrate index. About the constant K, after several round experiments, we found we will get the best QoE(about 2.69K)when K = 1070. The algorithm is as follows.

Algorithm 1 Bit rate algorithm

$throughput \leftarrow datasize/SendData$
$flag \leftarrow false$
$K \leftarrow 1070$
$i \leftarrow 2$
while $i \geq 0$ **do**
  if $RemainBuffer \geq K \times Bitrate[i]/throughput$ **then**
    $bitrateIndex \leftarrow i + 1$
    $flag \leftarrow true$
    $break$
  else
    $i \leftarrow i - 1$
  end if
end while
if $flag = false$ **then**
  $bitrateIndex \leftarrow 0$
end if

III. Experiments & Results

In our term project, there are many hyperparameters to tune in our model. However, there is no direct relationship between each parameter. As a result, we decide to try every possible combination exhaustively while keeping our eyes on the trend of the output score. Since it is impossible to test every variety; Therefore, we decide to use grid-search to find the suitable setting of our algorithm. There are three parameters: buffer-min, buffer-max, and lambda to tune in our method. In conclusion, the optimal point we found for the buffer-min, the buffer-max, and lambda are 0.35, 0.55, and 5, respectively. The highest QoE we scored is 2.68213118e+03. Figure 2 shows that if we fix lambda at 5, the local maximum point we found is at (0.35 0.55, 2.68213118e+03) for min-buffer, max-buffer, and QoE score, respectively. Figure 3 shows the result of the QoE score when we fix the min-buffer and the max-buffer at 0.35 and 0.55, respectively, and vary the value of lambda. Moreover, we also did some experiments on varying the triplet (min-buffer, max-buffer, lambda). The optimal point we can find is still at (0.35 0.55, 2.68213118e+03). Table 1. shows

TABLE I
THE FOLLOWING TABLE SHOWS THE RESULT OF THE EXPERIMENT

Network trace	QoE	Average time
Fixed	2.31206868e+03	8.90394403e-05
Low	1.72636876e+03	9.70890437e-05
Medium	2.39837507e+03	9.63125877e-05
High	4.35918749e+03	8.17377483e-05

the QoE and the average time of our algorithm under different network trace.

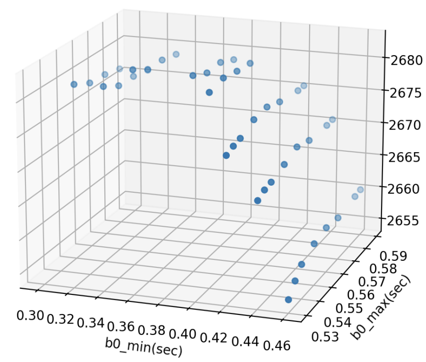

Fig. 2. The grid search result when lambda is fixed at 5

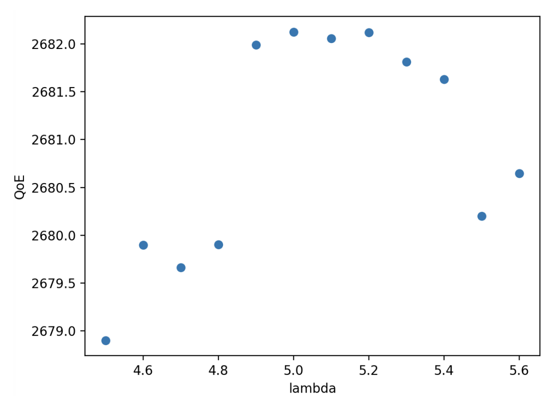

Fig. 3. The grid search result when min-buffer and max-buffer are fixed at 0.35 and 0.55 respectively

IV. Conclusion

In this term project, we proposed a hybrid scheme to take both latency and bitrate under consideration. We leverage the BBA algorithm for bitrate control, a hybrid control scheme for playback control, and an adaptive frame dropping method to achieve a high score. Also, we find a desirable combination of parameters by grid-search to reach a pleasing result. Lastly, the simulator shows that our method is comparable with other state-of-the-arts; However, our algorithm is lightweight, straightforward, and easy to understand.

REFERENCES

[1] Huan Peng, Yuan Zhang, Yongbei Yang, and Jinyao Yan. 2019. A Hybrid Control Scheme for Adaptive Live Streaming. In Proceedings of the 27th ACM International Conference on Multimedia (MM '19). Association for Computing Machinery, New York, NY, USA, 2627–2631. DOI:https://doi.org/10.1145/3343031.3356049

[2] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2014. A buffer-based approach to rate adaptation: evidence from a large video streaming service. SIGCOMM Comput. Commun. Rev. 44, 4 (October 2014), 187–198. DOI:https://doi.org/10.1145/2740070.2626296

[3] K. Spiteri, R. Urgaonkar and R. K. Sitaraman, "BOLA: Near-Optimal Bitrate Adaptation for Online Videos," in IEEE/ACM Transactions on Networking, vol. 28, no. 4, pp. 1698-1711, Aug. 2020, doi: 10.1109/TNET.2020.2996964.

APPENDIX A
Q AND A

Q : How do you determine the range of b0 and b1? Is it possible that the range you choose is only locally optimal?

A : We tried to determine the value by the relationship of the inequality of the paper. We used grid search to find it, so maybe it would be the local optimal value. However we thought it was good enough, so we didn't try other values.

Q : 請問你們認為Bitrate model的效能不好會影響，只採用frame drop 和playback control，是有實際implement 出來經過哪些測試嗎？因為我們組的情況認為$latency_{limit}$反而有負面的影響，謝謝。

A :We mainly refer to the paper of HYSA. It has many predictive variables. Hence, we focus on this direction to do experiments. I think a better QoE we get than HYSA can explain the origin bitrate model has some negative effects. About latency limit, It is possible that we have different models. Hence, Our testing result may be different.

Q : 想知道你們數據的結果，因為我們當初做單純Buffer的algo，成績都不太理想呢。

A :We considered not only the buffer based algorithm, but also the latency limit and target buffer, so we could get a better result. We had already provided our result in our paper and presentation.

Q : 想問K值怎麼定義及調整的?

A :K is a constant value to adjust the comparison result between remaining buffer and the value throughput divided by bit rate; We tried a lot of values of k, and found out that the QoE is maximized when k equals to 1070.

Q : play back control 裡面B_{min} 和B_{max} 的值是根據什麼界定的？

A :By Heuristic; however, the value of B_{min} and B_{max} should satisfy the relationship that the paper mentioned. Also, we did grid search, and hoped that the optimal point we found was the global maximum.

Q : 你們throughput是自己想出來的嗎，還是仍有用其他Paper的部分?

A : We refer to the code someone put in github, and make some adjustions like the constant value K.

Q :Why do you need to fix two parameters?

A :Because we want to find the possible value of lambda that maximizes the QoE score; As a result, we fixed B_{min} and B_{max}, and varying lambda to find the optimal point. Moreover, we did some experiments that vary three parameters, and the triplet (0.35, 0.55, 5) maximize the QoE score.