B2DASH: Bandwidth and Buffer-Based Dynamic Adaptive Streaming over HTTP

Peihan Du[1], Jian Wang[1(✉)], Xin Wang[2], and Zufeng Xu[3]

[1] Nanjing University, Nanjing, China
18523270951@163.com, wangjnju@nju.edu.cn

[2] 61428 unit of the Chinese People's Liberation Army, Beijing, China
xinwangwx@sina.com

[3] State Key Laboratory of Smart Grid Protection and Control, NARI Group Corporation, 19 ChengxinRoad, Nanjing, jiangsu 211000, China

Abstract. Currently, video streaming technology is widely used for entertainment, advertising, social networking and so on. Dynamic adaptive streaming over HTTP (DASH) has largely replaced the previous release of streaming video using UDP. Many researchers have proposed algorithms to improve DASH performance. There are two types of methods: bandwidth-based and buffer-based. Both methods have pros and cons. In this article, we propose a DASH algorithm that takes both bandwidth and buffer into account. And we imitate the method of network congestion control to adjust bitrate of the video segment. The algorithm was implemented and tested, and compared with the state-of-the-art DASH algorithm–BOLA. The results showed that B2DASH outperformed BOLA for both the average bitrate and buffer rise time.

Keywords: Adaptive algorithm
MPEG dynamic adaptive streaming over HTTP (MPEG DASH)
Bandwidth and buffer-based algorithm · Congestion control

1 Introduction

With the popularization of the Internet and the rapid development of the computer communications industry, there is an increasing demand on multimedia data. The Cisco Technical Report [5] predicts that mobile data traffic will increase significantly in the next few years, reaching 24.3 EB per month by 2019, and nearly 72% of the it will be video traffic. Therefore, the design of an efficient video streaming algorithm is critical to provide a high quality experience (QoE) to meet the growing demand for video streaming over wireless networks.

The most popular content provider such as YouTube, Netflix uses adaptive HTTP streaming technology to transmit video [16]. Adaptive transmission technologies include HTTP Live Streaming (HLS) and HTTP Dynamic Adaptive Streaming (DASH) standards [11]. The video is encoded into multiple video representations that are segmented into k-second segments. At the end of each

L. Meng and Y. Zhang (Eds.): MLICOM 2018, LNICST 251, pp. 565–574, 2018.
https://doi.org/10.1007/978-3-030-00557-3_56

segment, the client predicts the available bandwidth for the next k seconds and requests a video representation so that the segment bitrate matches the throughput.

According to past research, DASH's adaptive algorithm can be divided into two types of methods. First, the bandwidth-based method [1–4] is to select the video bitrate as close to the actual bandwidth as possible. However, it is difficult to predict instantaneous bandwidth because it depends on several factors such as delay, bandwidth change, buffer level or segment duration. Then, the buffer-based method [6,7] is to select the video bitrate based on the client's buffer level status. This method can reduce the number of interrupts, but it does not consider the bandwidth, and the video bitrate may be lower than it should be.

Since both methods have their own pros and cons, many studies have attempted to combine them in order to get better QoE. Jiang et al. [8] proposed an optimized bandwidth estimation algorithm based on the previous bandwidth average to improve the accuracy of bandwidth estimation and maintain the stability of the buffer length. Zhou et al. [9] proposed a buffer occupancy model to smooth short-term bandwidth changes. In order to maintain the smoothing of the video rate of the proportional-derivative (PD) controller, Zhou et al. [10] proposed a Markov decision model to estimate the bandwidth accurately, but it is difficult to obtain a good Markov transition matrix.

In this article, we proposed B2DASH, an adaptive algorithm combined buffer-based with bandwidth-based technique. The experimental results demonstrated the effectiveness of B2DASH adaptive algorithm in making users get higher bitrate and faster video startup speed than before.

We started this article by introducing the background of DASH. We described the algorithmic flow of the B2DASH algorithm in the third section, and then introduced our evaluation and data analysis in the fourth section. This article concluded with the fifth part of the conclusion.

2 Background and Related Work

Adaptive streaming media technology, especially DASH, can dynamically change video's quality to suit network conditions. Although DASH is gradually gaining popularity, there are still some issues that have led to inefficiencies. For example, Poor bandwidth prediction may causes bitrate fluctuations, especially in mobile networks; segmented transmission delays increase; video freezes, and may therefore have a negative effect on QoE.

Improvements in adaptive strategies can minimize the impact of changing networks on QoE. Recent work shows that client-side adaptive streaming algorithm uses two different methods: bandwidth-based and buffer-based. The representative of the bandwidth-based methods is PANDA [13] and Elastic [14]. The performance of the method may be affected by the accuracy of its bandwidth estimator. Bandwidth estimation and prediction are considered daunting tasks [15,16]. Buffer-based methods, such as the recent BOLA [17] and [18,19], avoid the inaccuracy of bandwidth estimation and use the system buffer as the main factor for bitrate switching. Most proposed buffer-based algorithms assume relatively large buffer sizes which are not suitable for short video. These work show that: one outstanding challenge was that under the two conditions of bandwidth stability and bandwidth fluctuation, the adaptive algorithm can play a relatively stable effect and achieve a higher bitrate.

This article is to develop an adaptive algorithm using both bandwidth and buffer-based algorithm to enhance the video's quality when the actual bandwidth is drastically changed, and the algorithm is combined with theory of congestion control. Furthermore, the loading speed of video playback has also been greatly optimized.

3 System Model

In this section, we proposed B2DASH, an adaptive algorithm which combines buffer-based with bandwidth-based optimization technique. B2DASH predicts the bandwidth according to the current bandwidth during the transmission of video, and will determine video bitrate by both bandwidth and buffer level. In addition, we imitate the TCP congestion control mechanism and divide the buffer into several levels. Different buffer level will lead to different adaptive method.

In this work, B2DASH consists of four parts: bandwidth prediction (Sect. 3.1), selection based on the bandwidth (Sect. 3.2), adjustment based on the congestion control (Sect. 3.3), algorithm overview (Sect. 3.4) The definitions of some variables are attached in Table 1.

Table 1. Main symbols and meaning

Symbol	Meaning
B(t)	The bandwidth at time t
t_c	The current time
S_c	The critical buffer size
S_{adq}	The adequate buffer size
S_{max}	The maximum buffer size
S_{now}	The buffer size at now
QI	Quality index
QI_n	QI of the $n-th$ transmission cycle
$bufferLevel_n$	Buffer level at the start of n-th transmission cycle

3.1 Bandwidth Prediction

In order to determine the benchmark value of the segment's bitrate, we first need to estimate the future bandwidth. Previous research [20] showed that simple prediction based on historical information has the best prediction accuracy.

Therefore, we use the measured bandwidth information of the previous segment to predict future bandwidth. Specifically, we extract the measurement information from the last transmission cycle and use the average as the future bandwidth. Assuming that the current time is t_c, the last round of transmission time we chose is T_{max}. During the period from T_{max} to t_c, we selected x sample points, $T_{max}<t_{c-x}<...<t_{c-1} <t_c$. Since the time is closer to t_c, the more referential for the next cycle of prediction, so we give it a higher weight. Select a function *w(t)* monotonically increasing with t and normalize it from T_{max} to t_c, i.e.:

$$\int_0^{T_{max}} w(t)dt = 1$$

Then, this transmission bandwidth can be predicted as:

$$B = \sum_{i=1}^{x} B(t_i) \int_{t_c-t_{i-1}}^{t_c-t_i} w(t_c - t_i)dt$$

3.2 Selection Based on the Bandwidth

After obtaining the prediction function *B(t)* of the bandwidth for a period of time in the future, a basic bitrate b is assigned to the segment during this period of time;

The test set is assumed to have a total of k bitrates for selection. These bitrates b are distributed from low to high in a set A=$\{b_1$,b_2, ...$b_k\}$. In the initial of playback, because the bandwidth is unstable and the buffer is limited, the minimum bitrate b_1 or b_2 is allocated. After that B2DASH allocate b_i which is closest to bandwidth predicted to video segment. This is the initial allocation based on bandwidth.

Figure 1 shows the result of the initial allocation based on bandwidth. In the case of fixed bandwidth, this algorithm can always stabilize the bitrate between 1–2 QIs.

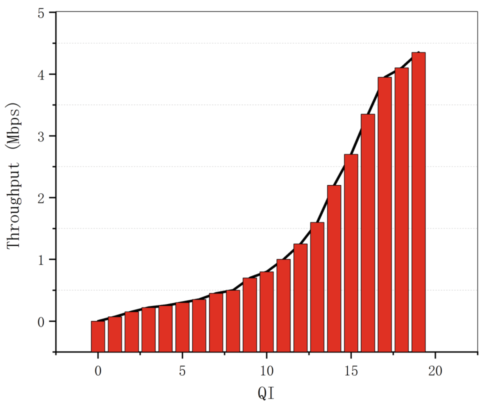

Fig. 1. Initial allocation based on bandwidth

With bandwidth stability, bandwidth prediction based on past experience can accurately and efficiently allocate bitrates to video segments without causing rebuffers. However, if the predicted bandwidth is much higher than the actual bandwidth, network congestion may occur. And for this reason, the buffer must be made to play a role in the selection of the bitrate.

3.3 Adjustment Based on the Congestion Control

The so-called congestion control is to avoid too much traffic into the network, to reduce the load of network communication links. At present, the Internet initiative standards have formulated four types of congestion control mechanisms: slow start, congestion avoidance, fast recover, and fast retransmission. We mainly use the method of slow start and congestion avoidance, and adjust bitrate according to length of buffer.

We assume that the maximum value of the buffer level is S_{max}. The goal of the algorithm is to keep the buffer within the interval [S_c, S_{adq}]. Then, B2DASH denote the current buffer level as S_{now}. When S_{now} is close to 0, use the slow start algorithm to start. In this way, after establishing the network channel connection, the congestion window value is increased gradually. At the time of initial transmission, QI=1 is set. Whenever the transmitter receives feedback from the receiver, bitrate can be doubled. The idea of the congestion avoidance algorithm is to slowly increase the video bitrate. After a transmission cycle, linear increasing bitrate instead of doubling it.

3.4 Algorithm Overview

The total algorithm that combines the bandwidth estimation and buffer congestion control is as follows:

As shown in Algorithm 1, the client first determines the range of transmission cycle numbers. Then at the begin of each cycle, the client need to get the throughput predicted (*throughput*), the length of buffer ($bufferLevel_n$), the QI of last cycle(QI_{n-1}). Then for each possible n, a suitable bitrate is selected via B2DASH. Function *estimate()* represent the process of selection based on the bandwidth. If the $bufferLevel_n$ was within [S_c,S_{adq}], B2DASH will simply use throughput to select bitrate. Otherwise slow start or congestion avoidance will adjust bitrate base on the result of *estimate(throughput)*. With such designs, the algorithm sketches the segment scheduling in a transmission cycle for B2DASH.

Algorithm 1. B2DASH

```
for n = 1 to N do
    if n = 1 then
        QI_n = 1
    else if bufferLevel_n < S_c then
        if 2 * QI_{n-1} < estimate(throughput) then
            QI_n = 2 * QI_{n-1}
        else
            QI_n = estimate(throughput/2)
        end if
    else if bufferLevel_n < S_adq then
        QI_n = estimate(throughput)
    end if
    if bufferLevel_n > S_adq then
        QI_n = max[estimate(throughput) + 1, QI_max]
    end if
end for
```

4 Evaluation

4.1 Experimental Environment

The system used in the experiment is Linux Ubuntu 14.04, equipped with a Linux kernel 3.13.0-66-generic; the server of the B2DASH system is deployed on a Jetty server. The B2DASH client uses the open source project dash.js[12] video player. This player is a DASH-compliant JavaScript player. In the network simulation, we used the Linux Traffic control (tc) to control and The bandwidth of the network was changed and the network structure was simulated using Linux netem. The data set used in the experiment was a distributed data set launched by the GPAC project.

We deployed the DASH standard dataset on the Jetty server and used tc to simulate the stable and variable network bandwidth. In order to compare with this proposed B2DASH, BOLA [17] was also simulated the same conditions, because BOLA showed good performance in other proposed works. The 20 bitrates for each video segment in the simulation are 47, 92, 135, 182, 226, 270, 353, 425, 538, 621, 808, 1.1M, 1.3M, 1.7M, 2.2M, 2.6M, 3.3 M, 3.8M, 4.2M and 4.7 Mbps. Video duration is 1 s. In this study, the performance indicator is the time of the buffer first rising to a safe level and the average bitrate.

(1) In general, the network conditions are most unstable at the beginning of video playback. Therefore, establishing a buffer as soon as possible can minimize video stuck in the initial transmission period.

(2) The average video bitrate is the average bitrate requested from the DASH server.

4.2 Buffer Rise Time Test

The test content in this section is: test the time that the video buffer reaches half of the buffer upper limit for the first time under different bandwidth conditions.

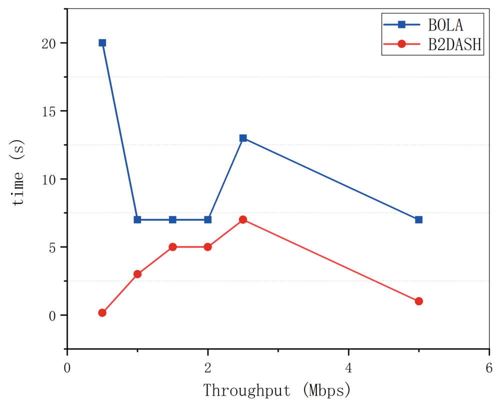

Fig. 2. Buffer rise time at different throughput

As can be seen from the Fig. 2, B2DASH benefits from its slow-start mechanism. It can quickly raise the buffer level to a very safe level at the beginning of the video's playback. B2DASH can deal with both high and low throughput. In contrast, BOLA's buffer is rise slowly and fluctuates at different throughputs. Especially when the link throughput is low, BOLA often takes a long time. Only then can the buffer be pulled up to half the maximum value.

4.3 Bitrate Comparison

There are test under stable throughput and test under variable throughput in this section.

The stable throughput test's content is: The throughput of the network is 1 Mbps in 0–100 s, then increase to 2 Mbps in 100–200 s, and then drop to 1 Mbps again in the following 100 s. The test content of variable throughput is that the throughput of the network is 1 Mbps within 0–100 s, then the code rate fluctuates between 1M and 2M every 10 s for 100 s, and then maintains a 2 Mbps for 100 s.

Table 2 shows the comparison of the two algorithms. Compared with BOLA, the average bitrate of B2DASH is higher, and the bitrate has increased by 15.72% and 7.36% respectively under the two different conditions of stable bandwidth and variable bandwidth. From the Figs. 3 and 4, it can be seen that, thanks to the slow start of B2DASH, when the network suddenly changes and the bitrate drops sharply, B2DASH can adapt to the change of the bitrate more quickly and quickly return to the appropriate bitrate. However, BOLA is not sensitive to bandwidth, and often finds the optimal bitrate after a long period of high throughput.

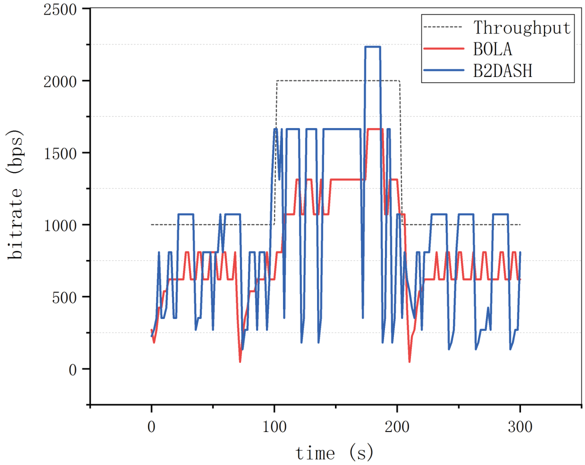

Fig. 3. Segment bitrate between BOLA and B2DASH in stable throughput

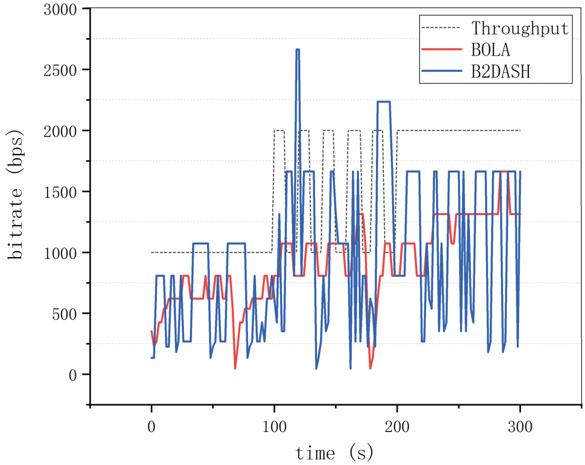

Fig. 4. Segment bitrate between BOLA and B2DASH in variable throughput

Table 2. Simulation results between BOLA and B2DASH

Algorithm	Avg.video rate(kbps)/stable	Avg.video rate(kbps)/variable
BOLA	833.23	914.25
B2DASH	967.21	981.58
Improvement	15.72%	7.36%

5 Conclusion

In this article, we propose the B2DASH algorithm combining the advantages of bandwidth and buffer-based methods to provide higher quality for video streams. And in the algorithm, the method of congestion control is applied to control the change of the bitrate. The proposed algorithm is implemented using the dash.js client, and compared with the advanced algorithm BOLA in dash.js, the performance metrics is the buffer rise time and the average transmission bitrate, both in the bandwidth stability and the bandwidth fast switching. The results show that B2DASH outperforms BOLA in both average bitrate and buffer rise time. Our future work will improve the performance of the B2DASH rate switching method.

Acknowledgements. This work was supported by State Key Laboratory of Smart Grid Protection and Control of NARI Group Corporation.

References

1. Liu, C., Bouazizi, I., Gabbouj, M.: Rate adaptation for adaptive HTTP streaming. In: ACM Conference on Multimedia Systems, pp. 169–174. ACM (2011)
2. Zhou, B., Wang, J., Zou, Z., et al.: Bandwidth estimation and rate adaptation in HTTP streaming. In: International Conference on Computing, NETWORKING and Communications, pp. 734–738. IEEE (2012)
3. Zhao, M., Gong, X., Liang, J., et al.: QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP. IEEE Trans. Circuits Syst. Video Technol. **25**(3), 451–465 (2014)
4. Fallah, Y.P., Mansour, H., Khan, S., et al.: A link adaptation scheme for efficient transmission of H.264 scalable video over multirate WLANs. IEEE Trans. Circuits Syst. Video Technol. **18**(7), 875–887 (2008)
5. White Paper: Cisco visual networking index: Global mobile data traffic- forecast update, 2014–2019. Technical report, Cisco, February 2015
6. Huang, T.Y., Johari, R., Mckeown, N.: Downton abbey without the hiccups: buffer-based rate adaptation for HTTP video streaming. In: ACM SIGCOMM Workshop on Future Human-Centric Multimedia NETWORKING, pp. 9–14. ACM (2013)
7. Huang, T.Y., Johari, R., Mckeown, N., et al.: A buffer-based approach to rate adaptation: evidence from a large video streaming service. In: ACM Conference on SIGCOMM, pp. 187–198. ACM (2014)
8. Jiang, J., Sekar, V., Zhang, H.: Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE. IEEE/ACM Trans. Netw. **22**(1), 326–340 (2014)
9. Zhou, C., Lin, C.W., Zhang, X., et al.: A control-theoretic approach to rate adaption for DASH over multiple content distribution servers. IEEE Trans. Circuits Syst. Video Technol. **24**(4), 681–694 (2014)
10. Zhou, C., Lin, C.W., Guo, Z.: mDASH: a markov decision-based rate adaptation approach for dynamic HTTP streaming. IEEE Trans. Multimed. **18**(4), 738–751 (2015)
11. ISO: Dynamic adaptive streaming over HTTP (DASH). Technical Report 23009, ISO/IEC (2012)

12. dash.js. https://github.com/Dash-Industry-Forum/dash.js/wiki
13. Li, Z.: Probe and adapt: rate adaptation for HTTP video streaming at scale. IEEE J. Sel. Areas Commun. **32**, 719–733 (2014)
14. De Cicco, L., Caldaralo, V., Palmisano, V., Mascolo, S.: ELASTIC: a client-side controller for dynamic adaptive streaming over HTTP (DASH). In: 2013 20th International Packet Video Workshop. IEEE (2013)
15. Yao, J., Kanhere, S.S., Hassan, M.: An empirical study of bandwidth predictability in mobile computing. In: Proceedings of the Third ACM International Workshop on Wireless Network Testbeds. ACM (2008)
16. Romirer-Maierhofer, P., Ricciato, F., D'Alconzo, A., Franzan, R., Karner, W.: Network-wide measurements of TCP RTT in 3G. In: Papadopouli, M., Owezarski, P., Pras, A. (eds.) TMA 2009. LNCS, vol. 5537, pp. 17–25. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01645-5_3
17. Spiteri, K., Urgaonkar, R., Sitaraman, R.K.: BOLA: near-optimal bitrate adaptation for online videos (2016)
18. Huang, T.-Y., Johari, R., McKeown, N.: Downton abbey without the hiccups: buffer-based rate adaptation for http video streaming. In: Proceedings of the 2013 ACM SIGCOMM Workshop on Future Humancentric Multimedia Networking. ACM (2013)
19. Huang, T.-Y., Johari, R., McKeown, N., Trunnell, M., Watson, M.: A buffer-based approach to rate adaptation: evidence from a large video streaming service. ACM SIGCOMM Comput. Commun. Rev. **44**, 187–198 (2015)
20. Tian, G., Liu, Y.: Towards agile and smooth video adaptation in dynamic HTTP streaming. In: Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, pp. 109–120. ACM (2012)