

HTTP Rate Adaptive Algorithm with High Bandwidth Utilization

M. Sajid Mushtaq, Brice Augustin and Abdelhamid Mellouk

LiSSi/TincNET Laboratory, Department of Networks and Telecoms, University of Paris-Est Créteil (UPEC) VdM, France

Abstract—Video streaming over Hypertext Transfer Protocol (HTTP) is highly dominant due to the availability of Internet support on many devices. The multimedia applications that generate IP traffic should be conducive with efficient utilization of network resources. Adaptive video streaming over HTTP becomes attractive for content service providers, as it not only uses the existing infrastructure of Web downloading (thus saving an extra cost), but it also provides the ability to change the video quality (bitrate) according to dynamic network conditions for increasing the user's perceived Quality of Experience (QoE). Video streaming over HTTP is easier and cheaper to move data closer to network users, and the video file is just like a normal Web object. In this paper, we have proposed a novel rate adaptive streaming algorithm that enhances the user's perceived quality with high bandwidth utilization for on-demand video. The proposed algorithm considers the following metrics in order to adapt the video quality, which are; player buffer, dropped of excess video frames per second (fps), and availability of network bandwidth. The algorithm is evaluated in dynamic real time Internet environment by using the wired and wireless network at the client side.

Keywords: QoE; QoS; Adaptive Streaming; Video; HTTP Streaming; Rate-Adaptation Algorithm; Multimedia.

I. INTRODUCTION

Nowadays, watching of online video contents is easily possible, thanks to the availability of a large variety of video-enabled devices and faster Internet connection. Cisco forecast report predicts that all forms of video (TV, Video on demand [VoD], Internet and P2P) will be 80% to 90% of global consumer traffic by 2017 [4].

In spite of powerful electronics devices and faster Internet connection, the end users still confront a problem of varying networks' condition; as the Internet is a collection of diverse networks all over the world. Indeed, to improve the user perceived Quality of Experience (QoE) (defined in [10]), despite of different network technologies and varying networks' condition, it requires new adaptive method that considers the varying behaviour of networks, and adopts the multimedia contents for delivering the highest quality video streaming to the end users. One such method is adaptive video streaming. The continue monitoring of network Quality of Service (QoS) parameters play an important role in selecting the appropriate video segment during video playback. The leading groups and companies, e.g., Microsoft, Apple, Adobe and MPEG/3GPP have introduced their own standard for adaptive video streaming (on-demand or live) over HTTP. The famous online Content Distribution Networks (CDNs) e.g. Netflix, Vudu, YouTube, Akamai, and Hulu, are using the

adaptive video streaming methods to cope the fluctuation in network bandwidth [3] [2], and maximize the user's QoE. The adaptive video streaming method has the advantage that it can efficiently share the network resources (bandwidth) among the users [5] when available network resources are limited, and to some extent it also lay off the burden of network resources management.

Earlier, researchers focused on the sender-driven based rate adaptation method, where sender or server estimated the client side parameters, and adapted the video streaming according to current situation. In [6], an adaptive method proposed that estimate the buffer occupancy of client at the server side, and adapted the video quality in order to maintain the client's buffer level above certain threshold value.

Now-a-days, the rate adaptive approaches are deviated from sender-driven based towards receiver-driven, where a client decides to adopt the video streaming quality by monitoring its parameters, and network conditions. In [7], the authors proposed a receiver-driven rate adaptation algorithm for video streaming over the HTTP. The proposed method was evaluated by simulator with the exponential and constant bit-rate background traffic. The proposed algorithm unable to select the appropriate video quality, as results clearly show the fluctuation. In [1] authors high lighted the behavior of different adaptive players for HTTP video streaming in order to check their stability in different scenarios. In [5], authors observed the HTTP based adaptive streaming method in term of fairness, efficient, and stability.

In [8], the adaptive method selects a video quality based on the estimated network bandwidth, and considered the client buffer level (20 to 50 seconds). The number of video quality shifts minimize when more video is buffered, because it will be less affected with instantaneous variation in network conditions, and also it did not consider the impact of frame drops rate. QoE-aware algorithm based on Dynamic Adaptive Streaming over HTTP (DASH) is presented in [9]. The authors presented that frequent change of video rate significantly degrade the user' QoE, and it was proposed to change the step by step video rate based on available bandwidth. In [12], a bandwidth estimation method is proposed; and based on past transmission history, the algorithm predicted the amount of data that client could download during a certain interval in the future. The algorithm was evaluated in terms of stalling frequency with Constant Bitrate (CBR), and did not consider impact of real time internet network and dropped video frame metric.

II. PROPOSED ALGORITHM

The pseudo-code of proposed rate adaptive algorithm is given in Algorithm 1. All symbols or abbreviations use in the proposed algorithm are provided in Table I. The algorithm dynamically selects an appropriate set of video representation R_s based on user's device properties (e.g. screen, resolution). In order to minimize the initial playback time, the algorithm selects the lowest video quality. It starts playing video as soon as the initial segments are downloaded, and buffer length (in seconds) reached to the start buffer length B_s . In case of quick start, B_s must set to low value, but it is necessary to set its value to be high enough, so it will be easy to compute the maximum bandwidth available for the stream. When a stream begins to play then algorithm considers the preferred buffer length B_p , instead of B_s . The B_p is the length of buffer (in seconds), after a stream begins playing. The value of B_p should be higher than B_s . The value of B_p represents the preferable buffer length, and it does not illustrate the current buffer length B while playing the video streaming.

In adaptive video streaming method, it is required that during the video playback period; available bandwidth, buffer, and dropped video frames should be monitored continuously in order to adapt the video quality according to time varying parameters for the next period. The playing duration of each period can be divided into n number of discrete time instants that is used to calculate the average value (bandwidth, buffer, and drop video frame) at the client side. The period length has a significant role in estimating the QoS parameter (e.g. bandwidth) [11]. The general expression for calculating the average buffer length B for the specific time period is given in Equation 1.

$$B_j = \frac{\sum_{i=1}^{n_j} b_{i,j}}{(T_{n,j} - T_{1,j})}, i = 1, 2, \dots, n \quad (1)$$

where $b_{i,j}$ is the measurement of instantaneous buffer for *Period* j at time instance i . In the proposed adaptive algorithm, we set the instantaneous time to 100 milliseconds. In the same manner, the average maximum bandwidth (BW_{max}) from Equation 2 and dropped video rate ($adfps$) from Equation 3 can be calculated as follows

$$BW_{max,j} = \frac{\sum_{i=1}^{n_j} bw_{i,j}}{(T_{n,j} - T_{1,j})}, i = 1, 2, \dots, n \quad (2)$$

$$adfps_j = \frac{\sum_{i=1}^{n_j} dfps_{i,j}}{(T_{n,j} - T_{1,j})}, i = 1, 2, \dots, n \quad (3)$$

where $dfps$ represent the video dropped frame per second, and it is calculated from Equation 4.

The maximum bandwidth capacity available for video stream is represented by BW_{max} . It represents a client bandwidth, not a server bandwidth and its value changes according to network conditions where client is currently exposed. The currently playing video stream is identified by $cSID$ that denotes any r_i ($i = 1, 2, \dots, n$) representation belongs to R_s , similarly the symbol $nSID$ denotes the possible next video stream identity that represents the r_{i+1} (possible one step high quality) or r_{i-1} (possible one step low quality) representation belongs to R_s .

The proposed algorithm also monitors the video stream in terms of a number of frame per second (fps). In such a circumstance when a video dropped frame per second ($dfps$) is higher (more than 10%) then it becomes necessary to make a decision in order to adopt lower video quality, as it influences the end user perceived video quality. In [13], the authors study the impact of video frame rate and resolution on the QoE by using the full-reference measurement method. The $dfps$ can be calculated from Equation 4

$$dfps = \frac{(df - pdfps)}{ct - tpdfps} \quad (4)$$

where df is the number of video frames dropped in the current playback session, and $pdfps$ is a number of video frames dropped in the previous playback session. The current time is denoted by ct , while $tpdfps$ represents the time when $pdfps$ occurred. In situation, when a recorded streaming is downloading, and video is a high-quality or high-resolution, but the decoder lag behind in decoding the required number of frames per second, because it does not have adequate system CPU resources that cause the frames dropped df . In live streaming, the buffer drops video frames if the latency is too high. This property df specifies the number of frames that were dropped and not presented to the user for viewing.

There are two more buffers i.e. current buffer time B_c and buffer time B_t . Initially, B_c is equal to B_s , but later it contains the same value as B_p . On the other hand, B_t specifies how long to buffer a video data before starting to display the stream. The proposed algorithm also takes into account the worst case scenario when the buffer is underflow condition. In order to avoid buffer underflow condition that causes the video streaming interruption in form of stalling or pausing, an aggressive buffer length B_a is introduced. In a case, when user buffer length B is less than B_a then a video stream switches to lowest possible bitrate in order to avoid the buffer from emptying, because an empty buffer can cause a pause or stutter in video streaming. However, shifting to lower possible video quality, it is necessary to check the QoS parameters more frequently for maximizing the user QoE.

The proposed algorithm considers three main parameters, i.e. B , BW_{max} , and fps in order to switch for lower or higher video quality. The proposed algorithm adapts the video streaming by taking into account the following conditions.

a) *Switch down to lower video:*

- When available maximum bandwidth BW_{max} is lower than the current video stream bitrate $cSRB$.
- When client buffer length B is less than current buffer time B_c .
- Dropped frame per second $adfps$ is greater than 10%.
- Aggressive mode, when client buffer length B is less than aggressive buffer length B_a .

b) *Switch-up to high video bitrate :*

- When BW_{max} is higher than the current video stream bitrate $cSBR$, but only if find a good buffer level.

III. RESULTS

The proposed rate adaptive algorithm is evaluated under the real time Internet environment, where available network

Algorithm 1: Rate Adaptive Algorithm Switch down

Input: A finite set $R_s = \{r_1, r_2, \dots, r_n\}$ of client specific video
Output: Select appropriate video ($nSID$) for end user
Result: Video quality switched down or up

```

1 Conditions to switch down video quality
2 if  $B < B_p$  or  $BW_{max} < cSBR$  or  $fps > 0$  and  $adfps > 0.10$  then
3   if  $B < B_p$  or  $BW_{max} < cSBR$  then
4      $i \leftarrow$  length of  $R_s$ 
5     while  $i \geq 0$  do
6       if  $BW_{max} > R_s(i)$  then
7          $nSID \leftarrow i$ 
8         break
9        $i \leftarrow i - 1$ 
10    if  $nSID < cSID$  then
11      if  $BW_{max} < cSBR$  then
12        Switch down due to less bandwidth
13      else
14        if  $B < B_c$  then
15          Switch down due to buffer
16    if  $B > B_c$  and  $B_c \neq B_p$  then
17       $B_c \leftarrow B_p$ 
18       $B_t \leftarrow B_c$ 
19  else
20    Switching down as  $adfps$  is greater than 10%
21    if  $adfps \geq 10\%$  and  $adfps < 14\%$  then
22       $nSID \leftarrow cSID - 1$ 
23    if  $adfps \geq 14\%$  and  $adfps \leq 20\%$  then
24       $nSID \leftarrow cSID - 2$ 
25    if  $adfps > 20\%$  then
26       $nSID \leftarrow 0$ 
27  if  $B < B_a$  then
28    Switch down to lowest quality to avoid interruption
29     $nSID \leftarrow 0$ 
30    check QoS more frequently
31 else
32  Switch Up on Maximum Bandwidth
33   $nSID \leftarrow 0$ 
34   $i \leftarrow$  length of  $R_s$ 
35  while  $i \geq 0$  do
36    if  $BW_{max} > R_s(i)$  then
37       $nSID \leftarrow i$ 
38      break
39     $i \leftarrow i - 1$ 
40  if  $nSID < cSID$  then
41     $nSID \leftarrow cSID$ 
42  else
43    if  $nSID > cSID$  then
44      switch-up only if find good buffer level
45      if  $B < B_c$  then
46         $nSID \leftarrow cSID$ 

```

Table I: Algorithm Abbreviation

| Words | Abbreviations |
|--------------------------------------|---------------|
| Next Stream ID | nSID |
| Current Stream ID | cSID |
| Average Maximum Bandwidth | BW_{max} |
| Client Specific Video Representation | R_s |
| Average Buffer Length | B |
| Start Buffer Length | B_s |
| Preferred Buffer Length | B_p |
| Aggressive Buffer Length | B_a |
| Current Buffer Time | B_c |
| Current Stream Bit-rate | cSBR |
| Buffer Time | B_t |
| Current Time | c_t |
| Current Frame Per Second | fps |
| Dropped Frame | df |
| Average Dropped Frame Per Second | adfps |
| Dropped Frame Per Second | dfps |
| Previous Dropped Frame Per Second | pdfps |
| Time Previous Dropped Frame | tpdfps |

bandwidth and user's buffer fluctuates; and their impact on the end user perceived quality is observed while watching the video streaming. The evaluation is done by using the wired Local Area Network (LAN), and also Wireless LAN (WLAN) networks available in the university laboratory. The client player is developed based on the proposed algorithm.

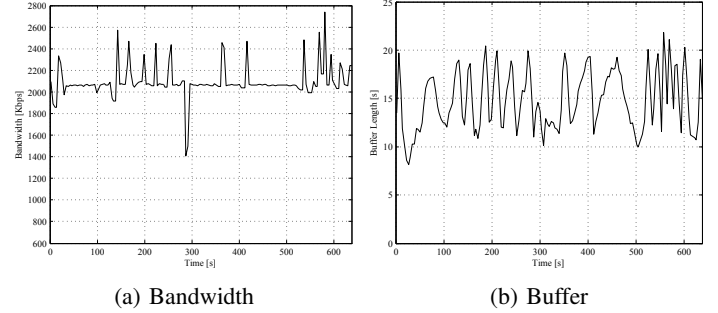


Figure 1: Client Bandwidth and Buffer in LAN

Figure 1a shows the behaviour of client's bandwidth that continuously fluctuates, and its average value is around 2.1 Mbps. It is clearly shown that there is one large dips in the bandwidth graph that causes the client player to switch down to lower possible video quality, i.e. 1300 bitrates, as it is observed in Figure 2b. This is a event that only causes the client's player to change the video quality due to bandwidth. Figure 1b illustrates the buffer status that oscillate, but mostly it is above than the preferred buffer length, i.e. 10 seconds. It is clearly shown that client buffer length never reached at aggressive buffer length (i.e. 4 seconds) that can cause the video quality must be switched down at lowest bitrates (i.e. 400 bitrates).

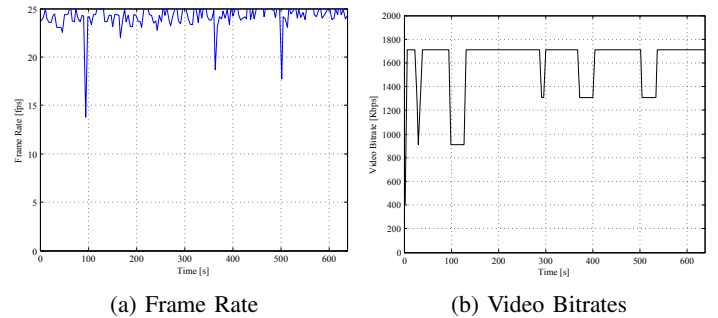


Figure 2: Client Frame Rate and Video Bitrates in LAN

The client player successfully decodes the video frames, and results in Figure 2a shows that there are three large dips (at 99, 368 and 500 seconds) as the decoder is unable to present the video frames to the user. Generally, it is occurred when the client machine does not has sufficient resources to process the video frames. The lost of video frames cause the video quality to shift down lower bitrates as it illustrates in Figure 2b. The impact of bandwidth, dropped frame rate, and buffer status cause the change of video quality as show in Figure 2b. The first decline in video quality is observed at 28 seconds that is occurred due to buffer, because the algorithm tries to keep the buffer level above than the preferable buffer length (10

seconds). The dropped frames rate forces the video quality to switch lower bitrates as noticed at 99, 368, and 500 seconds. The video quality also changes due to available bandwidth as it is observable at 300 seconds.

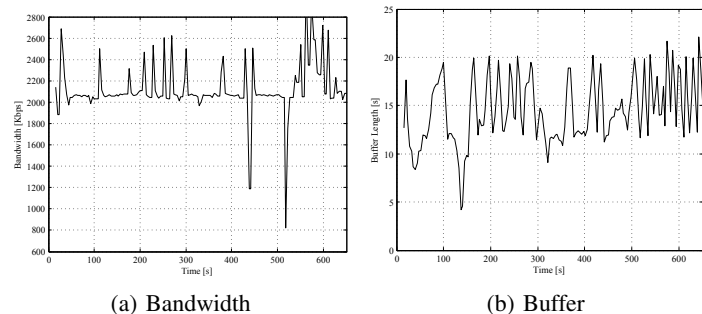


Figure 3: Client Bandwidth and Buffer in WLAN

The proposed algorithm is also evaluated in WLAN network, and mostly its maximum available bandwidth is more than 2 Mbps as shown in Figure 3a. The result shows that there are two times when large decrease in bandwidth are observed that result the switching of video quality to the next lowest possible quality, as illustrates in Figure 4b. The graph in Figure 3b depicts that client's buffer continuously fluctuates, and mostly it has value above than the preferable buffer length. The proposed method tries to keep the buffer length above than the preferable buffer length in order to avoid the stalling in playback. The result depicts that three times client's buffer has its values lower than the preferable buffer length, and decrease of buffer length at 149 seconds represents the aggressive buffer mode that shift the video quality to lowest quality level.

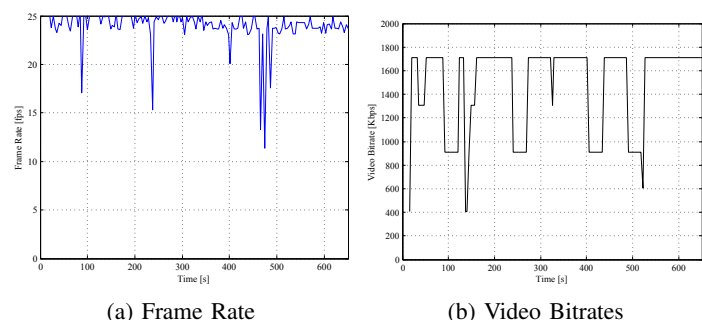


Figure 4: Client Frame Rate and Video Bitrates in WLAN

In case of WLAN, when video dropped frame rate exceed 10% as shown in Figure 4a then client player switches to the lower video quality. The three times dropped of video frames rate result the decrease in video quality as clearly shown by Figure 4b. Initially, the proposed algorithm, selects the lower video quality in order to quickly start the video streaming, but later it mostly plays high quality of available video (i.e. 1700 kbps) as shown in Figure 4b. Whenever, the client experiences the drops in buffer length, subsequently it switches to next lower video bitrates. However, when buffer length decreases lower than the aggressive buffer length then it shifts down to lowest video quality (at 149 seconds) in order to avoid the pausing in the video playback and maximize the user's

QoE. Similarly, decrease in maximum available bandwidth, and dropped video frames cause the decrease in video quality.

IV. CONCLUSION

In this paper, we have proposed a rate adaptive algorithm that adapts the video quality based on dynamic network bandwidth, dropped video frame rate, and user's buffer status. We have evaluated the proposed algorithm in real time dynamic Internet environment with two different client side networks (LAN and WLAN). The proposed algorithm can successfully adapt the video quality by considering the maximum available bandwidth, dropped video frame rate, and buffer length at the client side. The algorithm maximizes the user's QoE by avoiding the stalling during the video playback with efficient bandwidth utilization. In the future, we shall evaluate the proposed algorithm by considering the different client devices (e.g. smart phone, tablet, HD Screen), and observe its influence on user's QoE.

ACKNOWLEDGMENT

LISSI Laboratory thanks the CELTIC European Project 'NOTTS' for financial support.

REFERENCES

- [1] Saamer Akhshabi, 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, NOSSDAV '12*, pages 9–14, 2012.
- [2] L. De Cicco and S. Mascolo. An adaptive video streaming control system: Modeling, validation, and performance evaluation. *IEEE/ACM Transactions on Networking*, 22(2):526–539, April 2014.
- [3] Te-Yuan Huang, Nikhil Handigol, Brandon Heller, Nick McKeown, and Ramesh Johari. Confused, timid, and unstable: Picking a video streaming rate is hard. In *Proceedings of the ACM Conference on Internet Measurement Conference, IMC '12*, pages 225–238, 2012.
- [4] Cisco System Inc. In *Cisco Visual Networking Index: Forecast and Methodology, 2013-2017*, 2013.
- [5] Junchen Jiang, V. Sekar, and Hui Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking*, 22:326–340, Feb 2014.
- [6] L. S. Lam, J.Y.B. Lee, S.C. Liew, and W. Wang. A transparent rate adaptation algorithm for streaming video over the internet. In *18th International Conference on Advanced Information Networking and Applications (AINA)*, pages 346–351 Vol.1, 2004.
- [7] C. Liu, I. Bouazizi, and M. Gabbouj. Rate adaptation for adaptive HTTP streaming. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems, MMSys*, pages 169–174, 2011.
- [8] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz. Adaptation algorithm for adaptive streaming over http. In *19th International Packet Video Workshop (PV)*, pages 173–178, May 2012.
- [9] Ricky K. P. Mok, Xiapu Luo, Edmond W. W. Chan, and Rocky K. C. Chang. Qdash: A QoE-aware DASH system. In *Proceedings of the 3rd Multimedia Systems Conference, MMSys '12*, pages 11–22, 2012.
- [10] International Telecommunication Union-Telecommunication (ITU-T); Amendment 1: Definition of Quality of Experience. In *Recommendation P.10/G.100*, Jan. 2007.
- [11] Truong Cong Thang, H.T. Le, H.X. Nguyen, A.T. Pham, Jung Won Kang, and Yong Man Ro. Adaptive video streaming over http with dynamic resource estimation. *Journal of Communications and Networks*, 15(6):635–644, Dec 2013.
- [12] Bing Zhou, Jingyuan Wang, Zixuan Zou, and Jiangtao Wen. Bandwidth estimation and rate adaptation in HTTP streaming. In *International Conference on Computing, Networking and Communications (ICNC)*, pages 734–738, Jan 2012.
- [13] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld. Impact of frame rate and resolution on objective qoe metrics. In *Second International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 29–34, June 2010.