QoE Enhancement for Streamed Video by Reducing Quality Changes during Playout

by **Zahra Yazdanpanah**, Faculty of Engineering, IRIB University, Tehran, Iran; **Mohammad Behdadfar**, Faculty of Engineering, IRIB University, Tehran, Iran; **Mohammad Reza Noorifard**, Faculty of Engineering, IRIB University, Tehran, Iran.

Adaptive streaming, allows dynamic bit rate matching in different network conditions to reach an acceptable quality of experience level. In this paper, a rate adaptation algorithm is proposed for HAS (HTTP adaptive streaming) that determines the video bit rate based on the playback buffer level and the estimated bandwidth. The main property of this algorithm is reducing fast large changes of the media quality level and preserving the pre-defined minimal buffer length to reduce the media stall time. The proposed scheme was compared with other algorithms in single-client scenarios. The simulation results, in different network conditions, demonstrate high quality levels with smooth variations compared with existing solutions.

KEYWORDS
Quality of Experience, Video Streaming, Rate Adaptation. Playback, HAS

INTRODUCTION

Based on the statistics and predictions presented in *Forecast and Trends 2017-2022 White paper by Cisco public*, 82 percent of total internet traffic capacity will be video traffic by 2022. HTTP streaming has also become a cheap solution for multimedia content transmission based on being supported by internet infrastructure and permitted by most firewalls.

Traditional streaming solutions, utilise protocols such as real-time transport protocol (RTP) and real-time streaming protocol (RTSP) based on user datagram protocol (UDP) to control the video transmission rate. However, such protocols are difficult to deploy, because a specialised streaming server is required. In order to survive this constraint, the MPEG DASH protocol has been proposed.

Dynamic adaptive streaming over HTTP (DASH) [see "ISO/IEC IS 23009-1"] is an HTTP-based technique standardised by moving picture experts group (MPEG). For each item of multimedia content, a document is considered by MPEG-DASH called media presentation description (MPD), containing information about existing representations of multimedia content stored in the server. Each representation is divided into several segments, accessible by their Universal Resource Locator (URL). A client requests each segment from the server using a HTTP GET message. Also, for each representation, a profile is dedicated for specifying supported resolutions and encoding parameters.

In this paper, a client-side playout rate adaptation algorithm is proposed. As specified in the following sections, the basic idea is to select the video bit rates based on the playback buffer level and estimate bandwidth for each decision period.

Selecting high-quality video levels for playback, while reducing large changes in quality level of different segments and avoiding playback interruption, are other enhancements applied in the proposed solution.

Another consideration in the proposed algorithm is the reduction of stall time which may occur due to sudden bandwidth drops below the lowest acceptable bit rate. Another policy is decreasing the playback quality level smoothly to reduce large quality changes and avoiding a resultant weak quality of experience (QoE).

Existing playback and regional radio area schemes frequently modify their playback quality due to periodic network bandwidth fluctuations. However, the proposed algorithm selects a fixed quality level, if there is no possibility of interruption. This approach results in a better quality of experience.

Simulation results, show the advantages of the proposed solution compared with the existing ones.

The remainder of this paper is organised as follows: Section 2, presents related work. Section 3 describes the proposed approach. Section 4 presents experimental results. Section 5 concludes the paper.

RELATED WORK

Dongeun Suh et al [see *"QoE-enhanced Adaptation Algorithm over DASH for Multimedia Streaming"*, IEEE, ICOIN 2014] proposed a QoE-Adaptation Algorithm over DASH (QAAD) that minimises video quality changes during bandwidth fluctuations and preserves the minimum buffer level to reduce the possibility of interruption. Experimental results in a DASH testbed show that when the available network bandwidth decreases sharply, QAAD reduces the quality level gracefully, to minimise the quality change of consecutive segments, and even if the available bandwidth fluctuates periodically, QAAD provides a stabilised level of quality. Miller et al [see *"Adaptation Algorithm for adaptive streaming over HTTP"*, Proceedings of IEEE Packet Video Workshop, 2012] describe an approach that divides the buffer into predefined thresholds (*B1, B2, B3, Bmax*) where (*B1< B2< B3< Bmax*), and selects a video quality level based on the buffer level.

Rahman et al. [see *"An Efficient Rate Adaptation Algorithm for streaming over HTTP"*, IEEE, ICOIN 2018] propose an adaptation algorithm that selects the video quality level based on the estimated throughput and buffer level. Experimental results show that, when the available network bandwidth decreases sharply, the proposed algorithm suddenly reduces the quality level and if the available bandwidth fluctuates, the proposed algorithm provides a stabilised quality level. Yaqoob et al [See "A DASH-based Efficient Throughput and Buffer Occupancy-based Adaptation Algorithm for Smooth Multimedia Streaming, IEEE, 2019"] describe a DASH-based throughput and buffer occupancy-based adaptation algorithm (TBOA) that selects the video quality level based on the estimated throughput and buffer level. In this approach, the playback buffer is divided into partitions based on predefined thresholds (*B0, BL, BH*) where (*B0 < BL < BH*). To select the bit rate in each segment, the current level is compared to these three mentioned levels.

PROPOSED ALGORITHM

In this section, the bandwidth estimation approach and the bitrate selection algorithm are described.

3.1 System model

First, it is worth mentioning some important notations in ***Table 1***. Assume that a video file can be stored with at most *n* different quality levels in the streaming server. Consider these *n* video quality levels in the server are one of the members of $L = \{L1, L2,...,Ln\}$. Also consider that the video stream is divided into m segments, each containing T_s seconds of playback. To reach an acceptable QoE, the proposed Algorithm selects one quality level from the set *L* for each playing segment to adapt the video bitrate to the estimated throughput and playback buffer. *L1* is the lowest quality level related to lowest bit rate and *Ln* is the highest quality level, related to highest bit rate in the set *L* based on the property of $b(La+1) > b(La)$. The adaptation Algorithm in DASH client consists of two parts:
1) Bandwidth Estimation Part; *and*
2) Bitrate Selection Part

TABLE 1 Summary of notation

Notation	Definition
B_h	Pre-defined marginal buffer length.
B_{min}	Pre-defined minimal buffer length.
Bufferth	Pre-defined safe buffer length.
Safeth	Safe threshold
B_{avg}	Pre-defined safe buffer length
$B(t)$	Buffer length in video second at time t.
L	Set of the video quality levels.
L_{best}	Video quality level not exceeding the available bandwidth.
L_{pre}	Video quality level of the previously requested segment.
L_{next}	Video quality level of the next segment to be downloaded.
$b(L)$	Bitrate of the quality level *L*.
$T_{L,\ Bmin}$	Elapsed time to deplete the segments in the buffer while preserving the minimum buffer length B_{min} for the quality level *L*.
$T_{L,\ Bavg}$	Elapsed time to deplete the segments in the buffer while preserving B_{avg} level in buffer for the quality level *L*.
nL	The number of segments which can be downloaded without interruption for the quality level *L*.
L_{pb}	Pre-defined safe distance between quality levels.
T_s	Segment duration.
$BW_{estimated}$	Estimated bandwidth.

3.1.1 Bandwidth estimation

A client estimates the bandwidth of an upcoming segment based on the available bandwidth observed during downloading previous segments. The segment bandwidth BWrealtime is sampled in every T seconds and calculated as

$$BW_{realtime} = N/T \quad (1)$$

where *N* is the size of data downloaded in *T* seconds. The estimated bandwidth is smoothed by means of a weighted moving average scheme as

$$BW_{estimated} = w \times BW_{estimated} + (1-w) \times BW_{realtime} \quad (2)$$

where $BW_{estimated}$ is the previously estimated bandwidth and *w* is the weight factor for sampled bandwidth ($0 < w < 1$).

3.1.2 | Bitrate selection

After loading each segment, the bitrate selection module, selects the video quality level of the next segment, denoted by Lnext, based on available bandwidth BWestimated and the current buffer level B(t). On the other hand, Lbest represents the highest video quality level not exceeding the estimated network bandwidth and calculated as

$$L_{best} = argmax\ (\ b(L_a) \leqslant BW_{estimated}) \quad (3)$$

Where $L_a \in L$.

The main objective of the bitrate selection scheme in the proposed approach is to minimise sudden decrease in the chosen bitrates for sustainable QoE. ***Fig.1*** shows the flow chart of the proposed algorithm. To determine the quality level of each segment, the current best quality level, L_{best} is compared to the previous quality level, L_{pre}. Decision is made based on comparing L_{best} with L_{pre} with the following possible results:
1) Mode 1: L_{best} is equal to L_{pre}.
2) Mode 2: L_{best} is higher than L_{pre}.
3) Mode 3: L_{best} is lower than L_{pre}.

1) Mode 1: ($L_{best} = L_{pre}$), in order to avoid reducing the buffer level and changing the quality level, L_{next} is set to the previous video quality level L_{pre}. However, to avoid interruption, the next quality level will be set to the lowest quality level L_1 if the current buffer level is less than B_{min}.

2) Mode 2: ($L_{best} > L_{pre}$), the current buffer level should be considered to reduce the effects of the bitrate fluctuation and unpredictable buffer level reduction. Therefore, L_{next} is incremented as L_{pre}+1 if the current buffer length is larger than the pre-defined marginal buffer length B_h. Otherwise L_{next} is set to minimum quality level L_1 only if the current buffer length is lower than the B_{min}. If none of the above conditions are met, L_{next} is set to the previous video quality level L_{pre}. Therefore, the buffer level will be higher than the marginal level and reduce the possibility of interruption when the current network bandwidth suddenly decreases.

3) Mode 3: ($L_{best} < L_{pre}$), the current available network bandwidth cannot keep the previous video quality level L_{pre}, unless the bandwidth drop is transient or low. Therefore, two modes should be considered:
Mode 3(a): If the distance between Lbest and the previous level is less than a certain threshold called Lpb, ($L_{pre} - L_{best} < L_{pb}$), the buffer level is greater than the Buffer threshold level (*Bufferth*), that is (*B(t) > Bufferth*) and fluctuate factor is lower than Safe threshold (*Safeth*), that is (*fluctuate factor* ⩽ *Safeth*), L_{next} is considered as previous level ($L_{next} = L_{pre}$).

The system enters this mode if the bandwidth is slightly reduced, instantaneous or fluctuating. As long as the bitrate selection part remains in this mode, *fluctuate factor* starts to increase in each run. However, after exiting this mode, the *fluctuate factor* will be reset.

In other words, each time the system enters this mode, the factor increases by one unit. and by exiting this mode and entering mode 1 or mode 2, the factor becomes zero.

L_{best} =max {L, $B(L)$<= $BW_{estimated}$}
Mode1: if $L_{best} = L_{prev}$:
$L_{next} = L_{prev}$
F = 0 , U = 0

Mode2: if $L_{best} > L_{prev}$:
if $B(t) > B_h$:
$L_{next} = L_{prev}$ + l
else
$L_{next} = L_{prev}$
F = 0, U = 0

Mode3: if $L_{best} < L_{prev}$:
3(a) if $B(t) \geqslant Bufferth$, $L_{prev} - L_{best} \leqslant L_{pb}$, F $\leqslant$ *Safeth* :
$L_{next} = L_{prev}$, F = F + 1
3(b) else
1) if U = 0: $B_{avg} = (B(t) + B_{min})$ /2, l = B(t)
2) K = 0
3) if l~= B_{min}, k=2 :
k = k-1
if l = B(t) : l = B_{avg}
else : l = B_{min}
if $n_{L,l} > 1$:
$L_{next} = L_{prev}$ - k
else
k=k+1, return to line 3 of 3(b)

FIGURE 1*: Proposed Algorithm (fluctuate factor and updatefactor are represented by F and U, respectively)*

The bitrate selection part can stay in this mode until fluctuate factor becomes lower than *Safeth*. An increase of fluctuate factor from *Safeth* indicates a successive decrease in bandwidth, therefore the system cannot remain in this mode and quality level reduction is required. On the other hand, during fixed small bandwidth fluctuations, the bandwidth is regularly decreasing and increasing. Therefore, the bitrate selection part regularly fluctuates between Mode 3(a) and mode 1 or 2 and mode 3(a) preserves the previous quality level in oscillating bandwidth condition.

Mode 3(b) : If the conditions of mode 3(a) are not met, a bitrate reduction is inevitable. However, lowering more than two levels at a time would result in significant QoE degradation. Therefore, by considering the buffer level, the bitrate selection part tries to keep the next quality level close to the previous quality level. To this end, the expected number of segments downloaded before reaching a certain buffer level should be calculated.

In multimedia streaming, the playback buffer pushes out one segment to decoder in T_S seconds. On the contrary, for the video quality level l, the playback buffer receives one segment from the network in $T_S \times b(l) / BW_{estimated}$ seconds.

Therefore, the changes in the playback buffer level due to the downloading of each segment, can be obtained from

$$Diff = T_S \times b(l) / BW_{estimated} - T_S \qquad (4)$$

On the other hand, $T_{L,\,l} = B(t) - l$ is the time available before the buffer reaches a certain level of buffer l.

Let $n_{L,\,l}$ be the expected number of segments downloaded during $T_{L,\,l}$. Then, $n_{L,\,l}$ can be computed as

$$n_{L,\,l} = T_{L,\,l} / Diff \qquad (5)$$

Where Y is a floor function that returns the largest integer smaller than Y.

If the expected number of segments downloaded during $T_{L,\,l}$ is greater than or equal to 1,that is $n_{L,\,l} > 1$, the corresponding quality level l is defined as a feasible quality level.

If the buffer level is greater than B_{avg}, that is ($B(t) > B_{avg}$), by considering $l = B_{avg}$, the quality level that can be downloaded before reaching the B_{avg} buffer level is selected to reduce sudden quality change. Otherwise, by considering $l = B_{min}$, the quality level that can be downloaded before reaching the B_{min} buffer level is selected to prevent interruption.

In mode3(b), the distance between the current buffer level, $B(t)$, and the minimum buffer levels, B_{min}, is divided in two parts, if the updatefactor is reset.

The first part is considered from current buffer level to Bavg and the second part is considered from B_{avg} to B_{min}. The value of Bavg can be computed as

$$B_{avg} = (B(t) - B_{min}) / 2 \qquad (6)$$

The maximum feasible quality level lower than L_{pre} is searched by iterating the index k. To prevent lowering more than one level in quality, the previous quality level L_{pre} is first reduced by one level. If this level, $L_{pre} - 1$, can be downloaded during T_L, B_{avg} seconds, the next level, L_{next} is set to this level ($L_{next} = L_{pre} - 1$). Otherwise, this level will be reduced again by one level. If this level can be downloaded during the T_L, B_{min}, L_{next}, the selected level will be L_{pre}-2 (Lnext = $L_{pre} - 2$). Otherwise, this process continues while the quality level is reduced to ($L_{pre} - $ k) which can be downloaded during T_L, B_{min}, where nL_{pre},-k , $B_{min} > 1$.

Therefore, from quality levels lower than the previous level, the maximum quality level that can be downloaded before reaching the B_{avg} buffer level is selected. When the buffer reaches the B_{avg} level, the quality decreases by one level to prevent sudden changes in the quality level and the maximum quality that can be downloaded before reaching the B_{min} buffer level is selected.

As long as the system remains in this mode, the *updatefactor* will increase and B_{avg} will remain unchanged. Therefore, if system exits this mode and enters another mode, the *updatefactor* will reset.

By considering the T_L, B_{avg} time, system can smoothly reduce the bit rate and prevent instantaneous changes in quality level during times of sudden bandwidth fluctuations.

EXPERIMENTAL RESULT

The proposed algorithm is simulated in MATLAB as follows. The first video segment is fetched with the lowest bit rate. Fifteen different quality versions of the big buck bunny [see http://www.bigbuckbunny.org/] movie is then created. After that, Ffmpeg [see http://ffmpeg.org] is applied to encode the original video file with encoding rates of 45 kbps (quality 1), 89 kbps (quality 2), 131 kbps (quality 3), 178 kbps (quality 4), 221 kbps (quality 5), 263 kbps (quality 6), 334 kbps (quality 7), 396 kbps (quality 8), 522 kbps (quality 9), 595 kbps (quality 10), 791 kbps (quality 11), 1033 kbps (quality 12), 1245 kbps (quality 13), 1547 kbps (quality 14) and 2134 kbps (quality 15).

Each video stream is multiplexed with a common 128 kbps audio file to form a corresponding single MPEG-2 TS stream. Each stream is divided into 2-second segments. We set the weight factor w and the estimation period T to 0.8 and 0.3 seconds, respectively. Also, the pre-defined marginal buffer length, the minimum buffer length and buffer length are set to 10 seconds, 3*T_S seconds and 30 seconds, respectively. The

Bufferth and *Safeth* are set to 15 seconds as half of buffer length, and One-fifth of number of quality levels 3. The safe distance L_{pb} is also set to 4 levels.

The performance of the proposed approach is evaluated in comparison with that of QAAD, URrahman and TBOA Algorithms. The performance is evaluated in terms of four video quality metrics: average video bitrate, number of video quality switches, weighted changes in quality levels shown with a parameter called Reward [see *"Optimizing HTTP-Based Adaptive Streaming in Vehicular Environment Using Markov Decision Process"* IEEE TRANSACTIONS ON MULTIMEDIA, 2015] and playback interruptions. The reward parameter represents the amount of large and sudden changes. With higher quality level changes, greater weight will be assigned to it. Therefore, the greater reward parameter, indicates smaller QoE.

Table 2 shows the weights assigned to each incremental change in quality level. ***Table 3*** shows the weights assigned to each reduction in quality level. As shown in ***Tables 2*** and ***3***: The larger the change, the more weight is assigned to it.

TABLE 2 Weights assigned to each incremental change

Level change	0	1	2	3	4	5	6	7
Weight	0	1	5	10	25	50	100	150

Level change	8	9	10	11	12	13	14	
Weight	250	350	500	750	900	1100	1300	

TABLE 3 Weights assigned to each decreasing change

Level change	0	1	2	3	4	5	6	7
Weight	0	10	50	250	500	1000	1500	2200

Level change	8	9	10	11	12	13	14	
Weight	3000	3900	5500	6300	7900	9800	11800	

In the experiment scenario, four types of bandwidth stress test are conducted: 1) step-down test (Bw1), 2) stable and constant change test (Bw2), 3) fluctuation test (Bw3) and 4) mobile bandwidth test (Bw4).

In the step-down test, the available bandwidth is reduced from 1600 Kbps to 45 Kbps. On the other hand, in the fluctuation test, the available bandwidth is fluctuated repeatedly every 4 seconds from 1800 Kbps to 1000 Kbps and again 1000 Kbps to 1800 Kbps. In a stable and constant test, bandwidth changes from 2000 Kbps to 1000 Kbps after the first 100s and then switches repeatedly between 1000 Kbps and 500 kbps every 50s until the end of simulation at 300 seconds. In a mobile test, bandwidth changes randomly from 3000 Kbps to 300 Kbps.

4.1 Single client scenario

Quality of experience parameters for the proposed approach and three other approaches are shown in ***Table 4***.

First, the step-down bandwidth is considered. The video bitrates for the proposed approach, TBOA, URrahman and QAAD are shown in ***Fig. 2***. TBOA requests higher bitrates than other approaches at the beginning of the experiment. The reason is that other approaches try to support the confident marginal buffer length. Therefore, TBOA leads to the maximum stall time.

Second, the adaptive behaviour of the bitrate selection part, in case of stable and constant bandwidth change, is shown in ***Fig. 3***. Proposed approach achieves a high average video bitrate, least quality change and the least reward in comparison to other schemes. QAAD has lowest video bitrate and URrahman has the largest number of video quality switches.

Third, a bandwidth fluctuation is considered. The video bitrates for the proposed approach, TBOA, URrahman and QAAD are shown in ***Fig. 4***. The proposed approach and URrahman have a stable response to short-term periodic bandwidth variations with the least amount of video quality switches and reward.

Fourth, a mobile bandwidth is considered. The video bitrates for proposed Approach, TBOA, URrahman and QAAD are shown in ***Fig. 5***. Proposed approach and TBOA achieve a high average video bitrate. As shown in ***Fig. 5***. By decreasing the bandwidth in 226 to 280 seconds, the proposed approach slowly reduces the quality level to prevent large changes, while other approaches constantly change the quality level.

TABLE 4 *Statistic of different adaptation Algorithm*

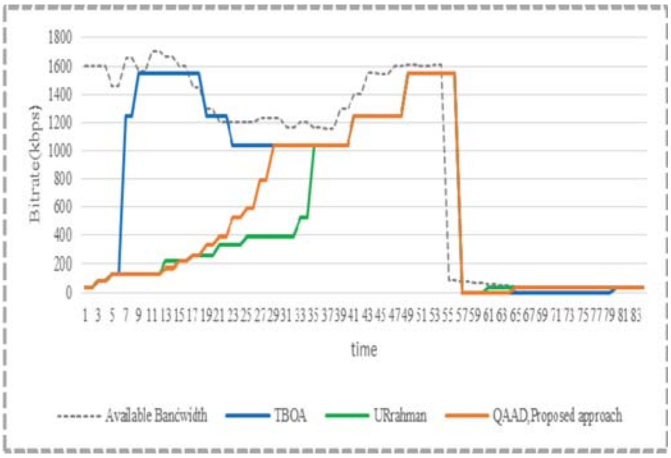
approach	Proposed approach			
BW	Bw1	Bw2	Bw3	Bw4
Video rate avg (kbps)	373	745	1014	1400
Number of change	24	21	13	62
Time of stall (seconds)	8.3	0	0	0
Reward	9811	104	33	297

approach	QAAD			
BW	Bw1	Bw2	Bw3	Bw4
Video rate avg (kbps)	373	595	858	1360
Number of change	24	23	86	103
Time of stall (seconds)	8.3	0	0	91
Reward	9811	104	264	528

approach	TBOA			
BW	Bw1	Bw2	Bw3	Bw4
Video rate avg (kbps)	584	765	1140	1400
Number of change	28	23	80	210
Time of stall (seconds)	23.1	0	0	2.9
Reward	10323	1313	2419	35281

approach	URrahman approach			
BW	Bw1	Bw2	Bw3	Bw4
Video rate avg (kbps)	334	705	935	1360
Number of change	24	72	13	38
Time of stall (seconds)	4	0	0	2.6
Reward	9821	21669	21	12111

FIGURE 2 *Step-down test*



FEATURE ARTICLE 1

FIGURE 3 *Stable and constant change test*

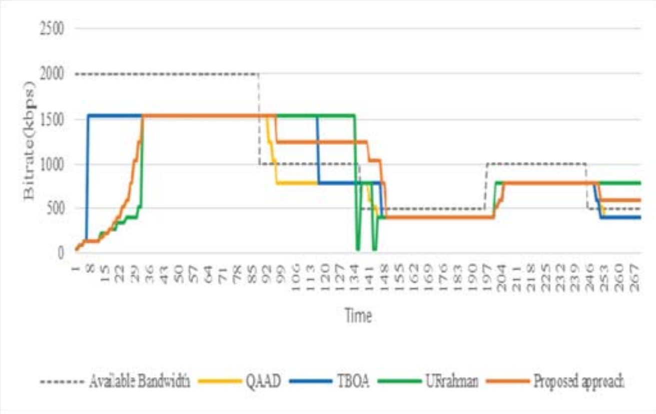


FIGURE 4 *Fluctuation test*

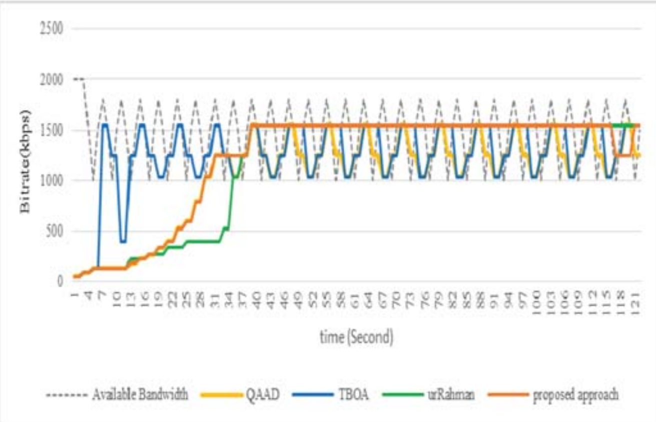


FIGURE 5 *Mobile bandwidth test*

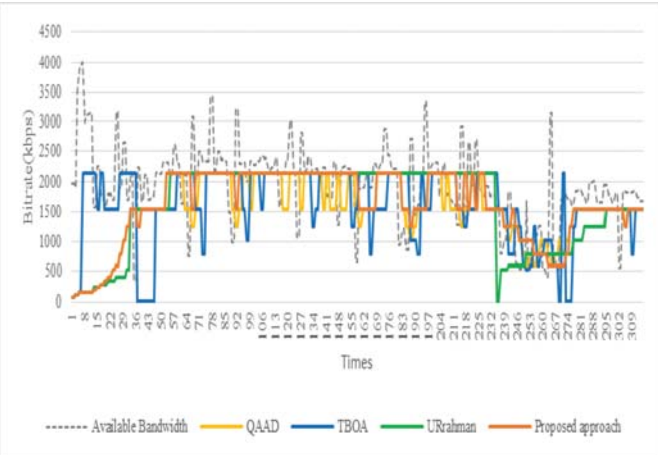


Table 5 shows the difference of the quality of experience parameters in the proposed method compared to other approaches.

In general, the proposed approach has better performance in terms of quality level changes and reward than other approaches and has better performance in terms of average quality level than URrahman and QAAD approaches.

The TBOA has better performance in terms of average quality level than other approaches, because it selects high quality levels at the beginning of the playback. However, it has the worst performance in terms of stall time and reward than other approaches.

TABLE 5 *Statistic of different adaptation Algorithm*

approach	Proposed approach – QAAD			
BW	Bw1	Bw2	Bw3	Bw4
Video rate avg. (kbps)	0	50	156	40
Number of changes	0	-2	-73	-41
Time of stall (seconds)	0	0	0	0
Reward	0	0	-231	-231

approach	Proposed approach – TBOA			
BW	Bw1	Bw2	Bw3	Bw4
Video rate avg. (kbps)	-211	-20	-126	0
Number of changes	-4	-2	-67	-148
Time of stall (seconds)	-14.8	0	0	-2.9
Reward	-512	-1206	-2386	-34984

approach	Proposed approach – URrahman			
BW	Bw1	Bw2	Bw3	Bw4
Video rate avg. (kbps)	39	40	79	40
Number of changes	0	-51	0	24
Time of stall (seconds)	4	0	0	-2.6
Reward	-10	-21565	12	-11814

CONCLUSION

In this paper, a rate adaptation algorithm for HTTP adaptive streaming was proposed that improves QoE by preventing playback interruption and reducing sudden and large changes in quality level during playback.

The quality assessment of the proposed algorithm was done in comparison with existing solutions, in terms of four quality metrics, i.e., video quality level, number of video quality switches, number of sudden video quality switches, and the stall time.

The Proposed Algorithm outperformed the other approaches under various network bandwidth conditions, by attaining a high video bitrate, reducing video bitrate change, smoother changes in quality level and reducing stall time, all resulting in better QoE.

AUTHORS

Mohammad Reza Noorifard was born in in 1965 in Iran. Since 1994, he has been a member of the faculty at IRIB University. His research interests include software-defined networking, information-centric networking, and the assessment of QoS/QoE in multimedia networks.

Mohammad Behdadfar was born in Iran in September 1977. He received his B.Sc., M.Sc., and Ph.D. degrees in Electrical Engineering from Isfahan University of Technology (IUT) in 1999, 2002, and 2010, respectively. He is currently an Assistant Professor in the Faculty of Engineering at IRIB University. His current research interests include high-speed networking, switch/router design, SDN/NFV, audio and video networking, networking algorithms, QoS, and QoE.

Zahra Yazdanpanah was born in October 1994 in Iran. She earned her B.Sc. in Electrical Engineering from Yazd University in 2017 and her M.Sc. in Telecommunication Engineering from IRIB University in 2021. Her research interests include video networking, QoS, and QoE. ■