applied sciences

MDPI

Article

Personalized Online Live Video Streaming Using Softmax-Based Multinomial Classification

Kyeongseon Kim [1], Dohyun Kwon [1], Joongheon Kim [1,*,†] and Aziz Mohaisen [2,*,†]

[1] School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Korea; kskim.caucse@gmail.com (K.K.); kdh1102@cau.ac.kr (D.K.)

[2] Department of Computer Science, University of Central Florida, Orlando, FL 32816, USA

* Correspondences: joongheon@cau.ac.kr (J.K.); mohaisen@ucf.edu (A.M.);
Tel.: +82-2-820-5911 (J.K.); +1+407-823-1294 (A.M.)

† These authors contributed equally to this work.

Received: 4 May 2019; Accepted: 30 May 2019; Published: 4 June 2019

Abstract: As the demand for over-the-top and online streaming services exponentially increases, many techniques for Quality of Experience (QoE) provisioning have been studied. Users can take actions (e.g., skipping) while streaming a video. Therefore, we should consider the viewing pattern of users rather than the network condition or video quality. In this context, we propose a proactive content-loading algorithm for improving per-user personalized preferences using multinomial softmax classification. Based on experimental results, the proposed algorithm has a personalized per-user content waiting time that is significantly lower than that of competing algorithms.

Keywords: QoE; softmax

1. Introduction

Media content distribution today is mainly based on online live streaming services, such as Youtube, Netflix, and Hulu. The major content distribution services are based on an over-the-top (OTT) service technology, which allows for the provision of a variety of Internet-based media content by third-party operators, such as the major communication and broadcasting operators. As the demand on OTT streaming services increases constantly, provisioning the Quality of Experience (QoE) for individual users becomes more important than ever [1]. OTT services that guarantee high QoE by considering individual user requirements are primarily sensitive to service delays, which severely degrade user satisfaction [2]. Therefore, various QoE provisioning techniques have been proposed in the literature to provide continuous multimedia services. In the prior related research work, QoE-related studies have considered variables in the metrics that would be only associated with the communications environment. For OTT and online streaming services, however, one additional variable is added: the user. Because users can take multiple actions during the progression of a video stream, at the level of seconds, various complex variables are added that may affect the QoE [3,4]. For example, this characteristic has been represented in a limiting phenomenon as a result of interest degradation caused by prolonged watching, and as captured in the single stimulus continuous quality evaluation (SSCQE) methods [5]. As such, and as this limiting point is reached, a user typically passes the video over the current part of progress to another interesting part. The adaptive streaming method, one of the existing approaches to ensuring QoE, fosters its limitations under the preceding scenario. Because the quality only changes depending on the environment in which video is played, and because the order in which a video is received is sequential, it is often the case that the parts of interest to users are not yet downloaded. As a result, users are made to feel uncomfortable by the resulting delay in obtaining the next chunk corresponding to the video part of interest. Under such circumstances, the previous studies trying to address QoE through pure quality of service environment metrics optimization are not suitable to address the users' behavioral patterns and reactions, mainly because they only focus on the users' environment, rather than users' direct action (e.g., skipping) [6,7]. In particular, the existing techniques to address the problem do not guarantee a high QoE.

To this end, this work is mainly focused on a new method that provides a personalized online live video streaming using multinominal classification. Basically, the suggested method considers users' viewing patterns to reduce the loading delay. To provide a customized loading order for each user, the method collects users' viewing pattern (e.g., skipping) data generated while watching the video. The method predicts users' action probability for each chunk obtained by the collected data and the softmax algorithm. The end goal of this classification is to optimize the user experience. First, we start by describing several previous studies for streaming transport techniques to improve the quality of streaming services considering the state of the network. Then, while addressing the shortcomings set forth earlier, we also propose an optimized streaming transport technique for users. The technique is called *user-based adaptive streaming* (UAS). The proposed technique classifies users' video viewing patterns based on a softmax algorithm, and then receives the chunk of video by priorities based on the analysis results. This investigation reflects not only the changing state of the network, but also the viewing patterns of users. These findings enable the partial units (chunks) to be transmitted efficiently. This will allow us to satisfy the different QoE criteria of each user.

The rest of this paper is organized as follows. Section 2 describes the existing relevant studies. Section 3 describes the UAS algorithm proposed in this paper. Section 4 verifies the performance of the UAS method by using a simulator and discusses its limitations. Section 5 concludes.

2. Related Work

There are several approaches to conduct streaming under QoE assurance requirements depending on network conditions [8–10]. In this section, we review the previous and well-known video streaming techniques and discuss corresponding limitations. In the following, the progressive download (PD) [11,12], HTTP-based adaptive streaming (HAS) [13], Scalable Video Coding (SVC), and Dynamic Adaptive Streaming over HTTP (DASH) [14] methods are presented.

2.1. Progressive Download Method

The progressive download (PD) method is a widely used multimedia streaming technique (refer to Figure 1). The PD method places the file on the web server and notifies the user with the address of file. Therefore, users are enabled to download and use the media file from the provided URL where the file is located. In order to maintain a high QoE without interrupting the media through PD, the network speed from the server to the user must be higher than the media's data rate. For instance, suppose a user has a 1 Mbps video, the minimum speed of the network should be above 1 Mbps to ensure seamless video playback. In this context, the PD method is simple to implement, and does not incur any additional costs. However, the PD method has several security issues, mainly because users must download the media file directly. Accordingly, PD is more suitable for the Video On Demand (VOD) market, rather than the OTT market.

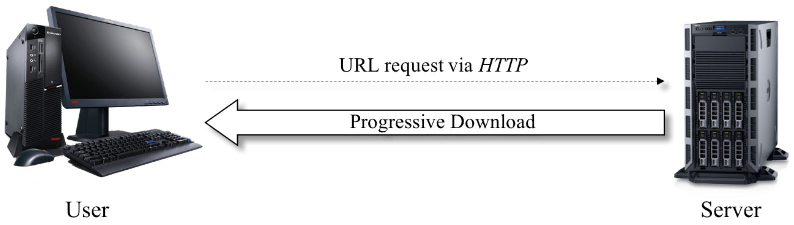


Figure 1. Structure of the progressive download (PD) method.

2.2. HTTP-Based Adaptive Streaming (HAS) Method

HTTP-based adaptive streaming (HAS) is a method that streams small pieces (chunks) of media to a server depending on the user's network status. Since the media file has various bit-rate types encoded, a bit rate can be selected depending on the network conditions. In addition, the chunks streaming method also makes it easy to select the media quality, even while watching a video [15]. However, HAS considers only the network status as a variable. That is, HAS does not consider "users" in the online streaming service, or their behavior, as variables (e.g., as in the OTT service).

2.3. Scalable Video Coding (SVC)

Scalable Video Coding (SVC) is a video coding method created by the Joint Video Team (JVT) of the ISO/IEC. The SVC method [16] is able to maintain the reconstruction quality associated with the rate of partial bit streams by enabling the transmission and decoding of such partial bit streams, as illustrated in Figure 2. In particular, the SVC method builds up multiple layers into a single video file. For example, assume that a single video file is encoded into three layers of high quality, medium quality, and low quality through the SVC method, denoted by layer 3, layer 2, and layer 1, respectively. In contrast to when the user chooses layer 1 (where only low-quality layers will be transmitted), when the user selects layer 3 all of layers 3, 2, and 1 are sent at once. This allows the user to be provided with all three quality levels of videos. For that reason, layer 3 is an enhancement layer and layer 1 is a base layer. In other words, SVC enables the transmission of video with various resolution and image quality while retaining bit rate, so that the user can be provided One Source Multi-Use (OSMU) [17] service through one encoding. Therefore, as in previous studies, the SVC method has been studied in real-time video streaming [18,19]. In particular, datasets through SVC methods have been used for Dynamic Adaptive Streaming over HTTP (DASH) technology.

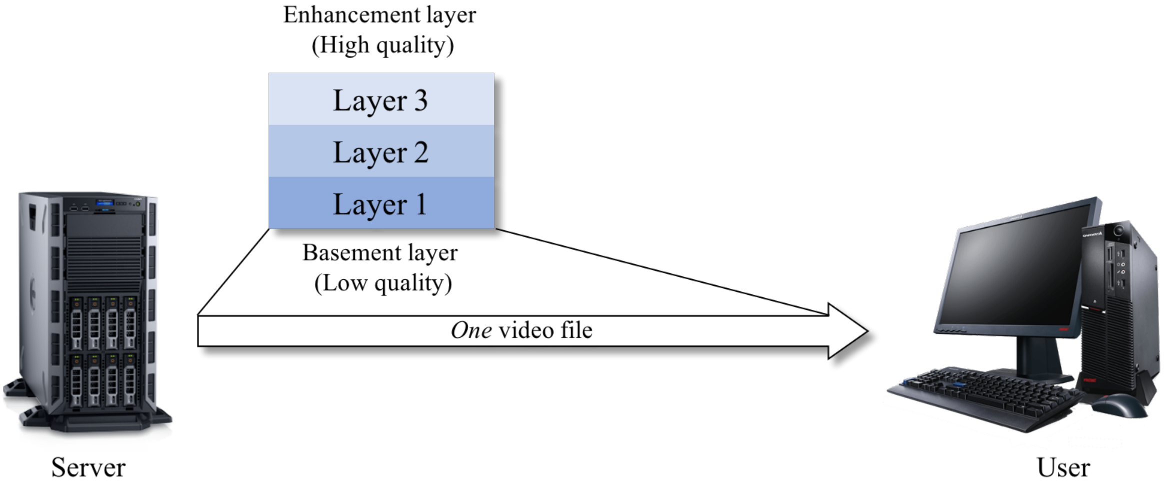


Figure 2. Structure of the scalable video coding (SVC) method.

2.4. DASH

The DASH method is able to provide video services with disparate features (e.g., adaptive bitrate switching) at different service types (e.g., on-demand, time-shifted viewing) to users, as illustrated in Figure 3 [20]. The DASH method has been studied for contributing to seamless video streaming services on one's mobile device. That is, the DASH aims to provide high-quality video not only in the case of the traditional video streaming services (which do not have any interaction with user), but also in cases with user interaction. The Real-Time Streaming Protocol (RTSP) method, one of the traditional video streaming methods, uses a stateful protocol. Upon a user connecting to the streaming server, the server keeps track of the state of the user's environment. After that, the server transfers continuous streams of packets (video) over UDP or TCP [21]. Unfortunately, the Hypertext Transfer Protocol (HTTP) is stateless, and the RTSP method operates based on HTTP. On the other hand, the DASH method has media (video) for various bitrates on the host's HTTP server. In practice, the user requests the most suitable bitrate file for user's network state based on a reference by the media presentation description (MPD) file. This feature allows customized files to be sent. In addition, the progressive download (PD) streaming method has multiple weaknesses with respect to bandwidth, bitrate adaptation, and live streaming. The DASH method compensates for these kinds of RTP/RTSP-based defects of PD through the trick mode, including rewind, fast skipping, etc. In this way, the DASH method enables live streaming considering user interaction. However, DASH falls short of considering the whole spectrum of interaction, including the viewing patterns of users. Each user has different viewing patterns depending on genre, video length, etc., which need to be taken into account when designing a protocol. In other words, previous studies of DASH considered the devices (desktop computers, smartphones, tablets, etc.) that users use to watch media. Moreover, they enable adaptive-bitrate streaming according to users' network conditions. However, these studies focus on the users' environment rather than the users' direct action (e.g., skipping). Therefore, we propose a method to overcome this problem. Toward this end, in this paper we consider the prediction of users' viewing pattern, and incorporate this pattern in the operation of DASH, namely in its trick mode to make DASH faster.

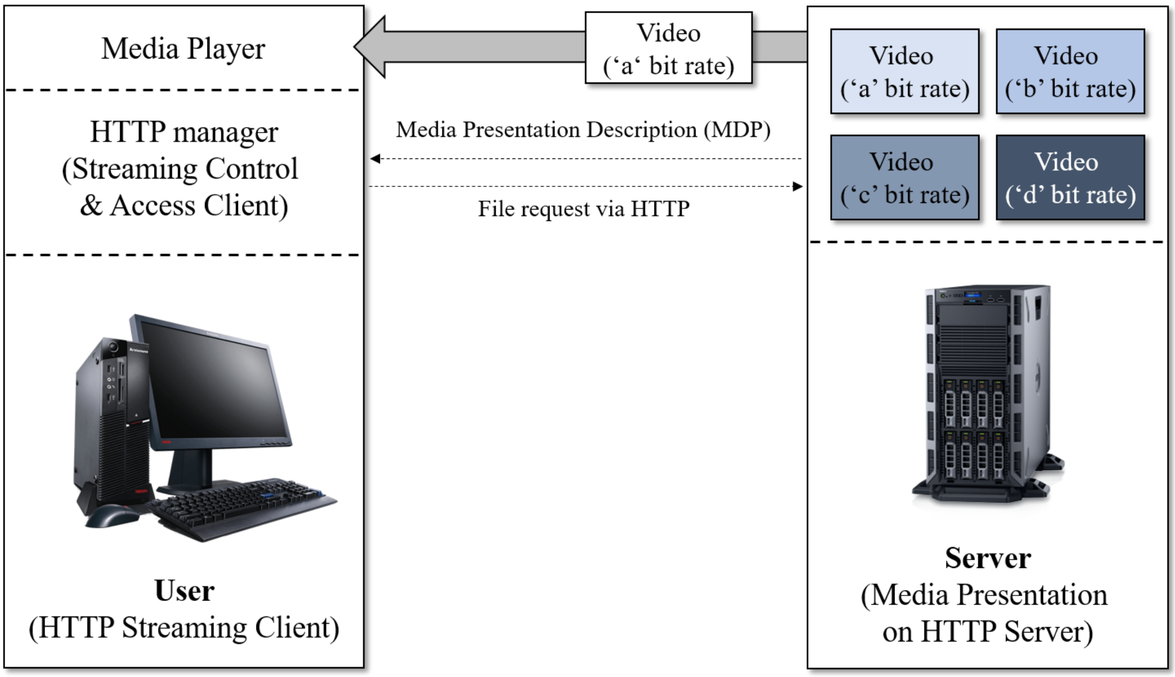

Media Player
HTTP manager (Streaming Control & Access Client)
User (HTTP Streaming Client)
Video ('a' bit rate)
Media Presentation Description (MDP)
File request via HTTP
Video ('a' bit rate)
Video ('b' bit rate)
Video ('c' bit rate)
Video ('d' bit rate)
Server (Media Presentation on HTTP Server)

Figure 3. Structure of the dynamic adaptive streaming over HTTP (DASH) method.

3. System Model and the Proposed Algorithm

Unlike the VOD market, which are one-way services, online streaming services allow users to enjoy video services interactively. The interactive actions in videos can be (i) stopping, (ii) skipping, or (iii) returning to desired parts of the video. This section describes the proposed personalized user-based adaptive streaming (UAS) method to overcome the issues with traditional adaptive streaming techniques. The UAS method aims to provide the best QoE through personalized user-optimized streaming service provisioning.

3.1. System Model

The overall process and the overall streaming system architecture using the UAS method proposed in this paper are shown in Figures 4 and 5, respectively. As illustrated in Figure 5, the basic components in this system are similar to those in DASH methods (details are in Section 2.4). However, the proposed UAS method includes more components, which are used to re-order the arriving chunks from servers.

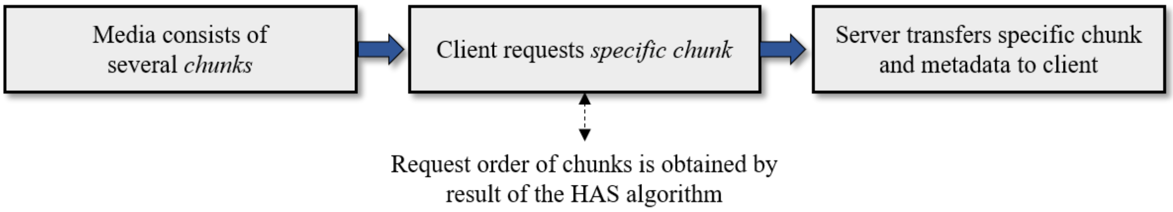

Media consists of several *chunks*
Client requests *specific chunk*
Server transfers specific chunk and metadata to client
Request order of chunks is obtained by result of the HAS algorithm

Figure 4. Overall process of user-based adaptive streaming (UAS) method. HAS: HTTP-based adaptive streaming.

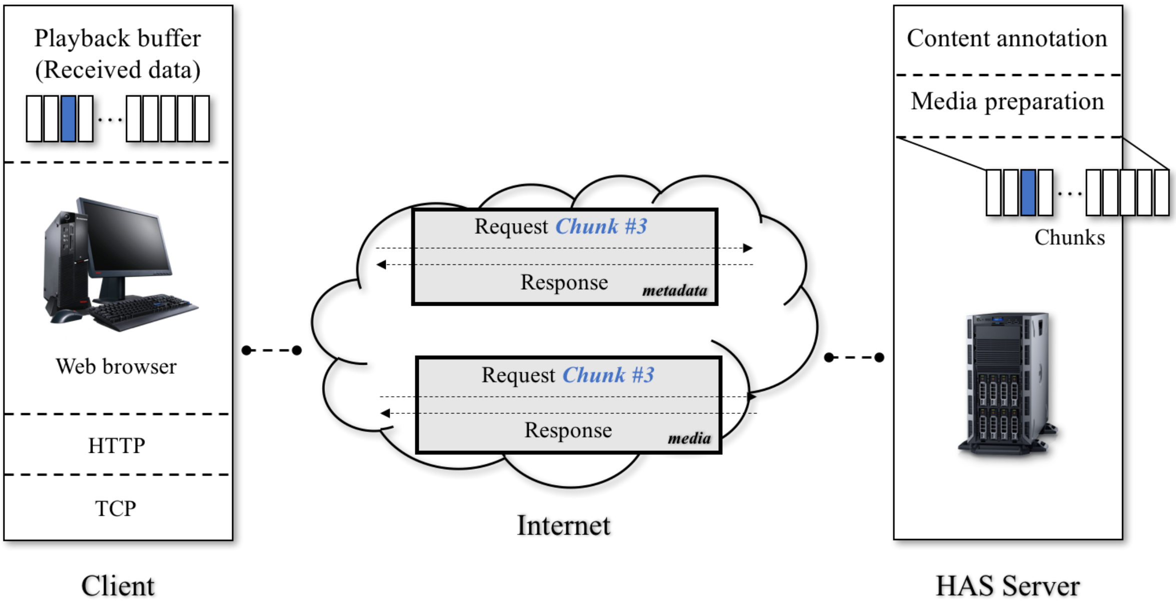


Figure 5. Structure of the personalized user-based adaptive streaming (UAS) method.

The UAS method is implemented at the server side. Thus, the basic process of UAS works in the same way as DASH. However, UAS re-organizes the received order of the units of video (i.e., chunks). In other words, the request is generally not sequential, and the chunk priority is set per chunk and then chunks are streamed according to their priority. A detailed method of prioritizing chunks by each user is described in Section 3.2.

3.2. Details of the Personalized UAS Method

The UAS method prioritizes the video chunks being transmitted by predicting user viewing patterns. That is, in order to provide a high QoE to individual users based on personalized differentiated requirements, the UAS method collects users' viewing patterns and classifies the priority of video chunks to be watched by a user based on the collected viewing patterns. This means that the chunk with the highest priority will be served before all other chunks.

The pseudo code in Algorithm 1 shows the process used in the UAS method.

The UAS method uses the `softmax` algorithm [22], which is widely used for multinomial classification in the machine learning research community [23,24]. Based on the `softmax`-based multinomial classification with N classes, it is appropriate to assign decision probability values $p_i, \forall i \in \{1, \cdots, N\}$ for classes $c_i, \forall i \in \{1, \cdots, N\}$ (note that $\sum_{\forall i} p_i = 1$). Therefore, our proposed UAS method uses the `softmax` to classify the priority of chunks based on user viewing patterns. In order to compute the `softmax`-based classification, multiple binary classification (or logistic regression) is used.

Algorithm 1 Pseudo code of the UAS method.

```
$e \leftarrow 0$
while $e = 0$ do
    for $c \leftarrow 0$ to n do
        progressive download on $c$
    end for
end while
$r \leftarrow softmax(n, t)$
for all result $r$ in $R$ do
    $r \leftarrow sort(r)$
    Streaming request of $r$ to server
end for
```

If we judge class A or not in a binomial classification, then the multinomial classification [25,26] will determine whether the class is A, B, or C. In this case, if classes A to C in multinomial classification are assumed to be the same as the video chunks, we can obtain probability values using the softmax for each chunk as shown below. Before describing the detailed softmax application process step-by-step, for the basic formula for softmax regression, we have:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}. \tag{1}$$

The probability of Equation (1) is statistically equivalented as:

$$p(y_i|x_i; W) = \frac{e^{f_{yj}}}{\sum_j e^{f_j}}, \tag{2}$$

where y_i is set as the probability of clicking chunks obtained through the action, and x_i is the action of the user who uses video streaming services. In practice, given x_i and the condition of W, (2) means a normalized probability of the y_i label. Applying Equation (2) to the UAS algorithm, an array of Y values obtained through the softmax "W" of each chunk is the probability that the user will click on each chunk. For example, if the result value "Y" is $(0.2, 0.7, 0.1)$, then the loading order of the chunks is *chunk*#2→ *chunk*#1→ *chunk*#3. We apply the above example to Equation (2), which we will explain in more detail in the following. According to Equation (2), the softmax output value of *chunk*#1, which is the probability of the user to select this chunk, is derived as shown in Equation (3):

$$p_{chunk\#1} = \frac{e^{z}_{chunk\#1}}{e^{z}_{chunk\#1} + e^{z}_{chunk\#2} + ... + e^{z}_{chunk\#n}} = \frac{e^{z}_{chunk\#1}}{\sum_{i=1}^{n} e^{z}_{chunk\#i}}, \tag{3}$$

where

$$Z = w * x + b. \tag{4}$$

Based on (3) and (4), and given three chunks, if the viewing pattern of the user of each chunk is (0.8, 0.9, 0.7), and Z is $-ln(1/p - 1)$, Z can be derived as:

$$(Z_{chunk\#1}, Z_{chunk\#2}, Z_{chunk\#3}) = (-\ln(\frac{1}{0.8} - 1), -\ln(\frac{1}{0.9} - 1), -\ln(\frac{1}{0.7} - 1)) = (1.39, 2.2, 0.85). \tag{5}$$

Applying Z to `softmax`, we obtain:

$$S_z = \texttt{argmax}\left(\frac{e^{1.39}}{e^{2.2}+e^{1.39}+e^{0.85}}, \frac{e^{2.2}}{e^{2.2}+e^{1.39}+e^{0.85}}, \frac{e^{0.85}}{e^{2.2}+e^{1.39}+e^{0.85}}\right) \tag{6}$$

$$= \texttt{argmax}(0.26, \boxed{0.59}, 0.15) = 2 \rightarrow \text{[Chunk \#2 will be selected]}. \tag{7}$$

Therefore, the UAS algorithm can set the download priority according to the viewing pattern of the user, so chunk #2 will be downloaded first. The priorities can be different in each of the categories and for each of the users. As a result, even for the same user, the order of sorting result according to the probability of each chunk is derived differently, and depends on the type of video. This priority value changes as the user takes action, and while watching the videos. The pseudo code specified in Algorithm 1 briefly shows the execution process of the proposed personalized UAS method.

4. Experiments

In this section, we show the performance of the proposed UAS method in various ways with real-world media data. We first outline the settings used for conducting the experiments, followed by the experimental results and discussion.

4.1. Settings

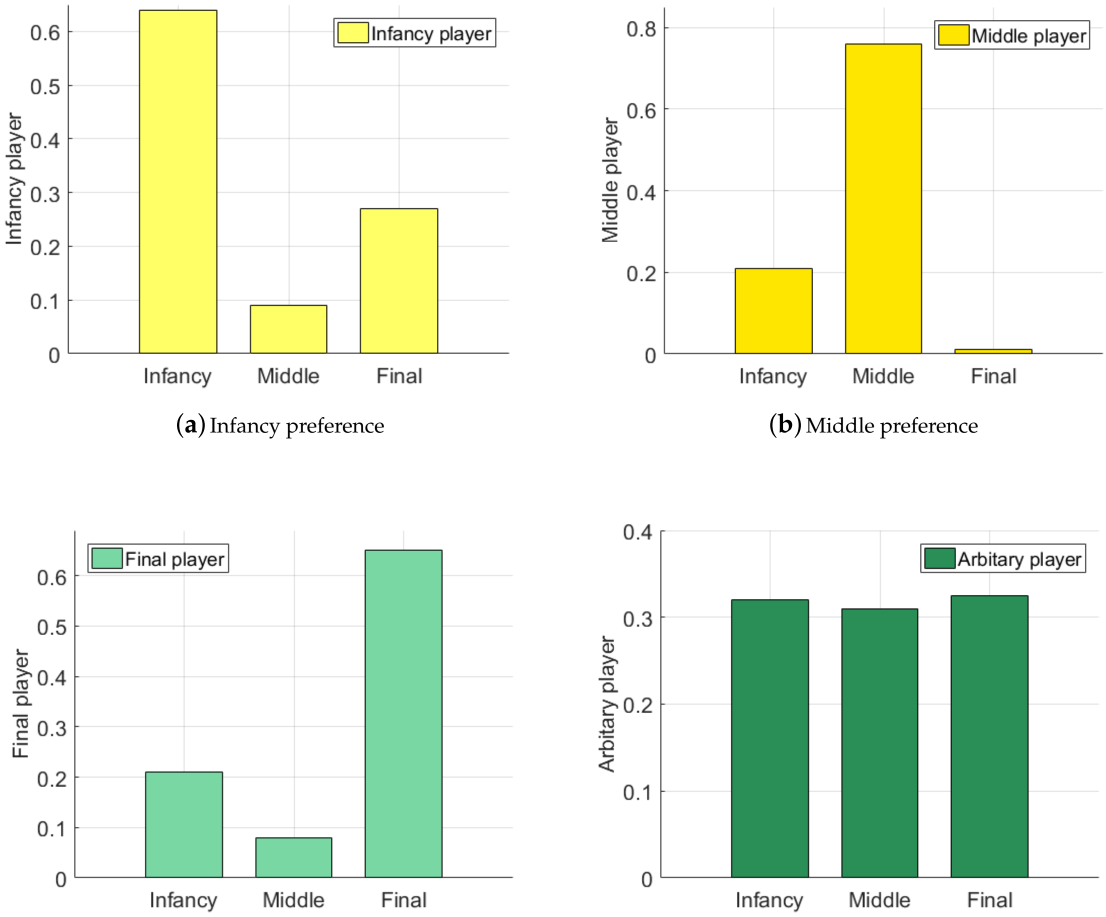
In the following we highlight the settings used for evaluating our technique. In our experiments, we assumed that each video, for simulation purposes, had at least one hour of playback time. Moreover, we note that abundant data of average playback history for each user is collected. We assumed that the given video genre of the corresponding histories was all identical. Furthermore, we assumed that each chunk had 2-s duration so that the entire time steps of video playback had 1800 evolution. Based on each user's playback history, the softmax function classifies each user's playback tendency using the calculated probability. The tendencies are composed as follows: infancy, middle, final, and arbitrary playback. For simplicity of simulation, the criteria of classifying playback tendency was set as 2:4:2 ratio from the initial part of a one-hour video, and descending weights were allocated for the corresponding video chunk *clicked* (*or hop*) by the user. Note that the weights decreased from 8 to 1 in order of *click*. After the former steps were complete, average weight values per each tendency category were calculated and utilized for input of softmax function. That is, after calculating average weights value of each playback position category, the values were utilized for the input of the softmax function. The highest value among them was assumed as the user's playback tendency. Taking the softmax property into account, trivial gaps between playback weights can be drastically differentiated. Accordingly, the expected tendency of playback per user can be classified clearly and comparatively in a precise manner. To sum up, given the abundant playback history of each user, proactive loading based on user-specific playback tendency expectation enables users to experience improved QoE. Proactive loading is enabled to prevent the *freezing phenomenon*, which halts playback of video due to the request of an unprepared video chunk. Thus, the softmax-based user tendency classification system was designed for this purpose.

Before outlining the performance evaluation, we show how the softmax classification works, as shown in Figure 6.

To measure the experimental environment, we used the video IDs "5Z6XSZcV27Q", "5zL3YJKygd4", "S91KmOLt-Fg", "HzNlrpabXw0", and "rvxGpkkjRyw" on youtube. We supposed that the users watched the same quality video. In other words, we considered users' viewing patterns, which were assumed to be more significant in this paper. As a future work, different video qualities could be considered to support the DASH method. The selection of each video was based on categories, where we selected a representative video of each category: sci-fi, romance, sports, documentary, and comedy. More detailed information of the selected videos is shown in Figure 7 and Table 1.

The initial values for learning and the table of resulting values are organized as shown in Tables 2 and 3. The priority value per category specified in Table 2 was the initial value. All initial values had the same priority. Therefore, all initial chunks had the same value of 0.1. These values were learned according to the behavioral patterns of the user, and then changed as in Table 3, which are called user-optimized values.

Table 3 shows the priority of chunks per video category obtained by learning. The proposed technique was tested under the assumption that the network environment was 24 frames droppedper second.



(**a**) Infancy preference

(**b**) Middle preference

(**c**) Final preference

(**d**) Arbitrary preference

Figure 6. (Individual) probability distribution for each user preference which determines the category of users in terms of playback tendency by using the softmax process: (**a**) infancy, (**b**) middle, (**c**) final, and (**d**) arbitrary playback.

Figure 7. List of videos used to evaluate the UAS method.

Table 1. Detailed information about the movie list.

Title	Video ID	Total Length	Loading Time	Quality	Category
Along With the Gods	5Z6XSZcV27Q	2:00	0:48	480p	SF
Love, Rosie	5zL3YJKygd4	2:01	0:15	480p	Romance
Relay of Nexen	S91KmOLt-Fg	1:45	0:35	480p	Sport
Golden Shoes	HzNlrpabXw0	2:55	1:18	480p	Documentary
Familyhood	rvxGpkkjRyw	1:24	0:32	480p	Comedy

Table 2. The initial value of the priority per category.

Category	SF	Romance	Sport	Documentary	Comedy
Chunk #1	0.1	0.1	0.1	0.1	0.1
...	...	...	...	...	...
Chunk #10	0.1	0.1	0.1	0.1	0.1

Table 3. The final priority of chunks per video category learned from user behavior.

Category	SF	Romance	Sport	Documentary	Comedy
Chunk #1	0.2	0	0	0.3	0.1
Chunk #2	0	0	0	0.1	0.2
Chunk #3	0.3	0.3	0	0.2	0.1
Chunk #4	0	0.3	0.5	0.4	0.1
Chunk #5	0.4	0	0	0	0.1
Chunk #6	0	0	0	0	0.1
Chunk #7	0.1	0.3	0	0	0.2
Chunk #8	0	0.1	0.5	0	0.1

4.2. Experimental Results

The UAS method proposed in this work controls the priority of the video unit (chunk) through user viewing patterns. Therefore, the loading time according to the adjusted priority application is an important criterion in the performance evaluation. The length of the chunk in the video for each category was designed using the reference video in Figure 7. For example, if the number of chunks was 8, the chunk length would be set as the the total length divided by 8. Thus, in the case of the video in the SF category, the total length was 2 min and the length of one partial unit ("chunk") was 15 s. Accordingly, the total time to load all eight chunks sequentially was 48 s, so the time to load a single chunk was six seconds. In this way, the length of one partial unit (chunk) of the video in the romance category was 15 s, and the time of loading one chunk was 1.875 s. In a similar way, the length of one partial unit (chunk) of the video in the sports category was 13.125 s, and the time of loading one chunk was 4.375 s. Moreover, the length of one partial unit (chunk) of the video in the documentary category was 21.875 s, and the time of loading one chunk was 9.75 s. Finally, the length of one partial unit (chunk) of the video in the comedy category was 10.5 s, and the time of loading one chunk was 4 s. This is summarized in Table 4.

Table 4. Summarized table of loading times per category.

Category	Chunk Length (s)	Chunk Loading Time (s)
SF	15	6
Romance	15	1.875
Sport	13.125	4.374
Documentary	21.875	9.75
Comedy	10.5	4

Based on Table 4, the user's video viewing patterns were applied to compare the performance. To conduct this comparison, it was assumed that the chunk of each video was skipped one time so that the user skipped the same chunk once, not several times. Moreover, we compared the performance of viewing with and without UAS. The playback of each image was started when one half of the chunk was loaded, and the system waited until the loading finished. As a result, Figure 8 shows the loading time with and without UAS applied.

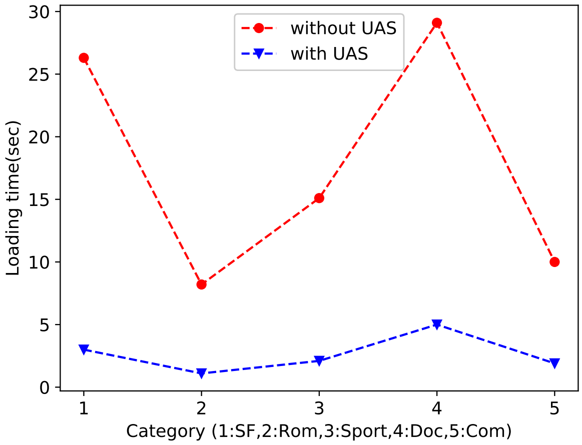


Figure 8. Comparison of experimental results.

As shown in Figure 8, the UAS method may incur some waiting time when in the sequential playback mode. On the other hand, when the user viewing pattern is applied, there is almost no waiting time. However, the sum of those two is significantly lower than when the UAS technique is not applied. To that end, we conclude that the QoE index of users who leverage this streaming service with the UAS method applied, as measured by the waiting time, will increase.

5. Conclusions and Future Work

Previous studies allowed users to select video resolution according to their network conditions by dividing a single video into several parts (chunks). Unfortunately, the user is the most important agent in maximizing the QoE, though previous studies have not considered user viewing patterns. In order to solve this problem, in this paper we propose the UAS method, which is a QoE-enhancement technique for video streaming systems. The proposed UAS technique predicts a partial video skip due to long-time viewing by the user. This prediction prevents unnecessary network usage and guarantees maximum continuity by minimizing the video interruption and delay.

It is necessary to study not only the QoE but also the network utilization by selecting a network in multiple network environments instead of a single network environment, which will be the focus of our future work. Another direction that we leave as future work is considering both video quality and user viewing patterns. Within this context, our experiments assumed all users wanted the same video quality. However, we should also consider different video qualities, which we will pursue as an additional feature of DASH in the future.

Author Contributions: K.K., J.K., and A.M. were the main researchers who initiated and organized the research reported in the paper, and all authors were responsible for analyzing the simulation results and writing the paper.

Funding: This research was supported by the Chung-Ang University Research Scholarship Grants in 2019 (for Kyeongseon Kim) and also by the IITP grant funded by the Korean government (MSIP) (No. 2017-0-00068, A Development of Driving Decision Engine for Autonomous Driving using Driving Experience Information).

Acknowledgments: J. Kim and A. Mohaisen are the corresponding authors of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee, S. Global Entertainment & Media Industry Status and Implications. Available online: http://www.kocca.kr/cop/bbs/view/B0000141/1823163.do?menuNo=200898 (accessed on 30 September 2014).
2. Kim, J.; Caire, G.; Molisch, A.F. Quality-Aware Streaming and Scheduling for Device-to-Device Video Delivery. *IEEE ACM Trans. Netw.* **2016**, 24, 2319–2331. [CrossRef]
3. Yu, H.; Zheng, D.; Zhao, B.Y.; Zheng, W. Understanding user behavior in large-scale video-on-demand systems. In Proceedings of the ACM SIGOPS Operating Systems Review, New York, NY, USA, 1–14 April 2006; pp. 333–344.
4. Mok, R.K.; Chan, E.W.; Luo, X.; Chang, R.K. Inferring the QoE of HTTP video streaming from user-viewing activities. In Proceedings of the First ACM SIGCOMM Workshop on Measurements up the Stack, Toronto, ON, Canada, 19 August 2011; pp. 31–36.
5. Yang, F.; Wan, S.; Chang, Y.; Wu, H.R. A novel objective no-reference metric for digital video quality assessment. *IEEE Signal Proc. Lett.* **2005**, *12*, 685–688. [CrossRef]
6. Ghinea, G.; Thomas, J.P. Quality of perception: User quality of service in multimedia presentations. *IEEE Trans. Multimed.* **2005**, *7*, 786–789. [CrossRef]
7. Zhu, P.; Zeng, W.; Li, C. Joint design of source rate control and QoS-aware congestion control for video streaming over the internet. *IEEE Trans. Multimed.* **2007**, 9, 366–376.
8. Georgopoulos, P.; Elkhatib, Y.; Broadbent, M.; Mu, M.; Race, N. Towards network-wide QoE fairness using openflow-assisted adaptive video streaming. In Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking, Hong Kong, China, 16 August 2013; pp. 15–20.
9. Mok, R.K.; Chan, E.W.; Chang, R.K. Measuring the quality of experience of HTTP video streaming. In Proceedings of the Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium, Dublin, Ireland, 23–27 May 2011; pp. 485–492.
10. Nam, H.; Kim, K.H.; Kim, J.Y.; Schulzrinne, H. Towards QoE-aware video streaming using SDN. In Proceedings of the Global Communications Conference (GLOBECOM), Austin, TX, USA, 9–13 December 2014; pp. 1317–1322.
11. Garcia, M.N.; Dytko, D.; Raake, A. Quality impact due to initial loading, stalling, and video bitrate in progressive download video services. In Proceedings of the IEEE Quality of Multimedia Experience (QoMEX), Berlin, Germany, 8–20 September 2014; pp. 129–134.
12. Chattopadhyay, S.; Ramaswamy, L.; Bhandarkar, S.M. A framework for encoding and caching of video for quality adaptive progressive download. In Proceedings of the 15th ACM International Conference on Multimedia, Seoul, Korea, 24–29 September 2007; pp. 775–778.
13. Kim, S.; Yun, D.; Chung, K. Video Quality Adaptation Scheme for Improving QoE in HTTP Adaptive Streaming. In Proceedings of the IEEE International Conference on Information Networking (ICOIN), Kota Kinabalu, Malaysia, 9–11 January 2016; pp. 201–205.
14. Koo, J.; Yi, J., Kim, J.; Hoque, M.A.; Choi, S. REQUEST: Seamless Dynamic Adaptive Streaming over HTTP for Multi-Homed Smartphone under Resource Constraints. In Proceedings of the ACM International Conference on Multimedia (ACMMM), Mountain View, CA, USA, 21–25 October 2017.
15. Cicalo, S.; Changuel, N.; Tralli, V.; Sayadi, B.; Faucheux, F.; Kerboeuf, S. Improving QoE and Fairness in HTTP Adaptive Streaming Over LTE Network. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 2284–2298. [CrossRef]

16. Schwarz, H.; Marpe, D.; Wiegand, T. Overview of the scalable video coding extension of the H. 264/AVC standard. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 1103–1120. [CrossRef]
17. Yoon, C.; Um, T.; Lee, H. Classification of N-Screen Services and its standardization. In Proceedings of the Advanced Communication Technology (ICACT), PyeongChang, Korea, 19–22 February 2012; pp. 597–602.
18. Kim, J.; Tian, Y.; Mangold, S.; Molisch, A.F. Joint Scalable Coding and Routing for 60 GHz Real-Time Live HD Video Streaming Applications. *IEEE Trans. Broadcast.* **2013**, *59*, 500–512.
19. Wien, M.; Cazoulat, R.; Graffunder, A.; Hutter, A.; Amon, P. Real-Time System for Adaptive Video Streaming Based on SVC. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 1227–1237. [CrossRef]
20. Stockhammer, T. Dynamic adaptive streaming over HTTP–: Standards and design principles. In Proceedings of the Second Annual ACM Conference on Multimedia Systems, San Jose, CA, USA, 23–25 February 2011; pp. 133–144.
21. Lin, R.; He, X.; Wang, S.; Luo, S.; Xiao, Y.; Zhang, X. Estimating End-to-End Available Bandwidth with Noises. *IEEE Access* **2017**, *5*, 22584–22589. [CrossRef]
22. Duan, K.; Keerthi, S.S.; Chu, W.; Shevade, S.K.; Poo, A.N. Multi-category Classification by Soft-Max Combination of Binary Classifiers. Multiple Classifier Systems. MCS 2003. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2709.
23. Tang, D.; Qin, B.; Liu, T. Learning semantic representations of users and products for document level sentiment classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 1014–1023.
24. Chen, H.; Sun, M.; Tu, C.; Lin, Y.; Liu, Z. Neural sentiment classification with user and product attention. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 21 September 2016; pp. 1650–1659.
25. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Torr, P.H. Conditional random fields as recurrent neural networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1529–1537.
26. Morris, M.R.; Huang, A.; Paepcke, A.; Winograd, T. Cooperative gestures: Multi-user gestural interactions for co-located groupware. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Gaithersburg, MD, USA, 24 April 2006; pp. 1201–1210.