entropy

MDPI

Article

A Fuzzy-Based Adaptive Streaming Algorithm for Reducing Entropy Rate of DASH Bitrate Fluctuation to Improve Mobile Quality of Service

Linh Van Ma [1], Jaehyung Park [1], Jiseung Nam [1], Hoyong Ryu [2] and Jinsul Kim [1,*]

[1] School of Electronics and Computer Engineering, Chonnam National University, Gwangju 500-757, Korea; linh.mavan@gmail.com (L.V.M.); hyeoung@chonnam.ac.kr (J.P.); jsnam@jnu.ac.kr (J.N.)

[2] Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea; hyryu@etri.re.kr

* Correspondence: jsworld@chonnam.ac.kr; Tel.: +82-62-530-1808

Received: 27 July 2017; Accepted: 4 September 2017; Published: 7 September 2017

Abstract: Dynamic adaptive streaming over Hypertext Transfer Protocol (HTTP) is an advanced technology in video streaming to deal with the uncertainty of network states. However, this technology has one drawback as the network states frequently and continuously change. The quality of a video streaming fluctuates along with the network changes, and it might reduce the quality of service. In recent years, many researchers have proposed several adaptive streaming algorithms to reduce such changes. However, these algorithms only consider the current state of a network. Thus, these algorithms might result in inaccurate estimates of a video quality in the near term. Therefore, in this paper, we propose a method using fuzzy logic and a mathematics moving average technique, in order to reduce mobile video quality fluctuation in Dynamic Adaptive Streaming over HTTP (DASH). First, we calculate the moving average of the bandwidth and buffer values for a given period. On the basis of differences between real and average values, we propose a fuzzy logic system to deduce the value of the video quality representation for the next request. In addition, we use the entropy rate of a bandwidth measurement sequence to measure the predictable/stabilization of our method. The experiment results show that our proposed method reduces video quality fluctuation as well as improves 40% of bandwidth utilization compared to existing methods.

Keywords: dynamic adaptive streaming; fuzzy inference; moving average; bandwidth fluctuation; mobile video quality; bandwidth usage

1. Introduction

Users recently have had access to increased network bandwidth such as 5G technology. On the one hand, users have many advantages in streaming services. On the contrary, they might have very rough experiences if network throughput fluctuates rapidly and frequently, especially mobile users. For example, a mobile user signs up to a 5G service/80 Mbps, which is high network bandwidth. It brings the user to the huge gap of fluctuation between a low and high throughput. This problem degrades the quality of mobile services such as video streaming because the streaming is sensitive to network changes.

In the past, streaming media servers used Real Time Streaming Protocol (RTSP) [1] for media streaming delivery. RTSP is a stateful protocol that maintains a connection through the RTP Control Protocol (RTCP) which cooperates with Real-time Transport Protocol (RTP). However, nowadays, most servers support Hypertext Transfer Protocol (HTTP)-based streaming because it inherits the benefits of HTTP. HTTP is a stateless protocol and requires a single port. It can also traverse firewalls.

The principle of adaptive bitrate HTTP-based [2,3] is to produce multiple versions of video content, such as various bitrates and different spatial resolutions on a server. The bitrates vary from low quality to high quality to serve a user in all conditions of the network bandwidth. The technique then divides each bitrate into multiple chunks. Continually, each chunk divides into small parts or segments with a duration length that varies from two to ten seconds. All of the information of the segments and chunks are included in a manifest file called Media Presentation Description (MPD). The information contains chunk location, segment duration, type of bitrate, etc. On the client side, a player searches for segment information in the manifest file MPD. On the basis of the manifest file, the player first requests the lowest-quality representation and then increases the quality of representation depending on network conditions. During playback, the video quality changes to match network conditions. To do that, the player does two main tasks. First, the player periodically measures server-to-client bandwidth. Second, the player chooses the maximum coding rate for sustainability given current network conditions.

The advantage of the technique is that it can switch the streaming among the different bitrates automatically without awareness of a user. If the network speed is greater than a certain bitrate, then it switches downloading the next higher adjacent bitrate. Many organizations involved in the project of building such Adaptive Bitrate Streaming (ABS) applications as well as improving delivering technique in the real business applications. We currently have several types of the streaming that are based on the technique of ABS. Moving Picture Experts Group (MPEG) created Dynamic Adaptive Streaming over HTTP (MPEG-DASH or DASH) [4–6]. Adobe HTTP Dynamic Streaming [5] supports both Real-Time Messaging Protocol (RTMP) and HTTP protocol. Apple HTTP Live Streaming (Apple Technology Company, Cupertino, CA, USA) [3] is a standard streaming service on the iPhone, and Microsoft Smooth Streaming (Microsoft Corporation Technology Company, Albuquerque, NM, USA) [5] is commonly used for Windows, etc.

The problem of frequent network changes has been solved by the adaptive streaming as discussed above. However, the problem has not been resolved thoroughly because the adaptive streaming frequently changes bitrates if network metric values such as bandwidth continuously change from low to high. Researchers have invested the problem and proposed many adaptive algorithms to improve network utilization and reduce the changes. An adaptive algorithm does select representations of video streaming that match to the current network state. However, most studies only concerned about the state of the network at present to provide alternative representations. In addition, modeling a problem in a network is hard due to the uncertainty of the network as users continuously join and leave the network. Even though we can model network problems, sometimes we must solve nonlinear optimization problems.

In DASH currently, if the segment downloading time takes less than the segmentation duration, the player must wait until the segment request timeout. This problem leads to unstable generated traffic, which can trap other players in the unfair distribution [7]. As addressed in [8], the problem is challenging because adaptive streaming-based always has an inevitable mismatch between the actual network throughput and the selected video bitrate. Therefore, in this paper, we will present a method based on DASH [6] to solve the emerged problem. In this article, we take advantage of a fuzzy logic application such as a fuzzy research in [9] to handle the uncertainty of the network system. In addition, we use a cumulative moving average to average a set of values because this average technique can express related information between near-term past values and current values. When we have a new value, the technique updates the average of data to present. More specifically, we extend our previous research [10] to improve the mobile quality of service. We improve our previously proposed method for mobile devices to maximize bandwidth usage and minimize the fluctuation of mobile video quality streaming. First, we find the average of previous bandwidth/buffer variation values of a client by using a moving average calculation. Second, we trace-back at a certain number of the previous fluctuation values to infer the degree of current network fluctuation. We then predict the next request representation for the client by our proposed algorithm.

In comparison with our previous contribution [10], we reformulate fuzzy input equations to reduce computation resources. In addition, we dig deeper into the determination of trace-back step in our proposed algorithm. In the experiment, we test with more resolutions up to 4K and compare our proposed method with another method that used the same fuzzy approach. Finally, we show that our approach leads to lower entropy rate compared to other methods.

We organize the rest of the paper as the following. Section 2 provides related research. Section 3 states the problem and challenges of high-quality mobile video streaming over the Internet and the future trend of high-definition video using DASH for mobile devices. Furthermore, Section 4 presents our proposed method that discusses system working flow and major components. Then, Section 5 includes an experiment where we build a server on Node.js [11], and a mobile client runs on Android. Finally, we conclude our work with the future work of the research.

2. Related Works

Nowadays, popular service providers such as YouTube, Netflix or Apple TV, Skype, WhatsApp are using Over-The-Top (OTT) services [12]. Video delivery with the adaptive streaming has gained a lot of popularity. By encoding/storing each of these at various quality levels and then dividing a source video into multiple segments with a duration around two to ten seconds, it allows clients to dynamically adapt the requested quality during the playback along with network changes. However, the method of quality selection is unpredictable and heuristics. Thus, it is unable to deal with a vast range of highly dynamic network fluctuation.

Packet loss, end-to-end delay, and unpredictable bandwidth are all problems that can affect the quality of video streaming over the Internet. To deal with such problems, Xiong et al. [13] presented an idea to adapt the video bitrate with the fluctuation of network transmission rate. Similar to our approach, they utilized fuzzy control to propose a feedback mechanism that can make a decision on adoption bit-rate. Additionally, they also addressed a future issue of fairness in the presence of multiple concurrent users. So far, this method focuses on video layer while our proposed method concerns an algorithm to select video representation to reduce the fluctuation of the bandwidth, though their video compression efficiency is still much lower than traditional encoders.

While our proposed method is HTTP-based, research in [14] is Transmission Control Protocol (TCP)-based with the same idea of adapting rate along with network conditions. Kuschnig et al. [14] used H.264/SVC (MPEG-4 Part 10, Advanced Video Coding/Scalable Video Coding) to encode/decode a multimedia file at the sender and receiver. They set up a rate-control algorithm to cope with bandwidth limitation as well as congestion. The encoding/decoding rate changes depending on TCP throughput. Their research currently is still limited in the performance of the request–response streams by their fixed parameter settings.

An application of interval type-2 fuzzy logic [15] overcomes a major problem of bandwidth fluctuation while sending out packages. As a result, their method improved delivery of video quality significantly compared with existing traditional controllers such as TCP flow control and a type-1 fuzzy logic. Their work is on the network transport layer as opposed to our method that performs on the application layer of the network transport protocol. Even though they had a dramatic improvement, their method still limited with video coding techniques that might depend on encoding/decoding performance.

Low network metric, such as low latency streaming and low bandwidth, is a great challenge in mobile video streaming, especially in live services deployment. HTTP streaming has recently become a commonly discussed topic in multimedia communications with ongoing standardization activities. Bouzakaria et al. [16] addressed a problem of low and latency in MPEG-DASH. The latency in DASH is affected by the asynchronous fetch of media segments; the segmentation delay; the buffering at the client side; and the time to download the segments. In the first version of the DASH standard, the MPD only indicates the *availabilityStartTime* value. The client having no knowledge of how segments are produced. This problem leads to a latency of one segment duration. Therefore, they measured the overhead of a frame sent and analyzed a behavior of such overhead. In such a way, their method reduced the initial segment delay and allowed faster switching between bitrates. Meanwhile, our research concentrates on overall network utilization without considering the initial season of DASH.

While most of the previous research focused on a client side, Toni et al. [17] formed a problem of finding a set of representations in an adaptive streaming server because no scientific study supported the problem. They addressed that the issue of network changes can be solved if we have forecasting algorithms. The crystal-ball algorithm [18] considered buffer occupancy and bitrates to predict bandwidth. They also analyzed when the predictions are useful and the effect of their prediction alogorithms on adaptive streaming. Their experiment results showed that prediction-based algorithms improve the quality of service (QoS). However, their research did not pay attention to the value of bandwidth and buffer before the present time. It might result in incorrectness prediction results. On the another hand, geo-predictive crowdsourcing [19] has an approach on clients' previous download experience. Their crowdsourcing-based algorithms outperformed all other methods regarding QoS. They used the buffer level and the current representation of a client to predict bandwidth without considering past bandwidth values. If they could combine the earlier bandwidth values and the buffer level, the results might significantly improve. These reasons motivate us to come up with the idea of network state prediction based on some factors such as the buffer level and bandwidth.

As pointed out by Sobhani et al. [8], adaptation methods in ABS can be categorized into two different categories: (1) bandwidth-base [13,20–26]; (2) buffer-based [27,28]. The bandwidth-based method relies on the current bandwidth change to select the best quality video quality for the user. It also ensures the continuity and real-time video streaming. The buffer–base method examines the state of the buffer to provide the choice of video quality representations so that it will not cause the buffer overflow or underflow. Logically, the bandwidth-based method focuses on bandwidth. This method can not determine the state of the buffer. In addition, this method also raises some issues in the process of playing out streaming video at the client such as the player ignores frames due to overflowing buffer. In contrast, the player has to wait for frame download due to buffer underflow. This problem also happens to buffer-based methods without considering bandwidth.

On the basis of the change of buffer and throughput, Sobhani et al. [8] introduced a fuzzy-based adaptation algorithm to adjust the quality of video as the rnetwork changes. However, this method only considers the value of bandwidth and buffer at present. They do not yet research about the past states of the network. This method works efficiently when bandwidth and buffer are stable. If these two parameters frequently change, sometimes that change causes the buffer underflow, which might degrade the method efficiency because it causes the player sometimes to stop streaming. Therefore, in this research, we rely on their methods to make improvements to the problem raised above.

As mentioned in the introduction section, the main goal of a client is to reach the highest quality of experience (QoE) [29]. Therefore, the fluctuation of video streaming in unstable network condition, which may negatively affect the user's QoE. Thus, the experiments in [30] consider two-second segments to overcome being sluggish to bandwidth variations. Moreover, considering small segment duration does not reflect how this method prevents buffer under-run in the case of large segment durations such as ten seconds and large bandwidth variations [27].

Using entropy to characterize per-application network traffic, Petkov et al. [31] estimated the entropy of difference streaming scenarios and divided them into categories of entropy measurement from the lowest to the highest. The entropy rate of a sequence is a measure of how predictable a sequence is based on past observations. Invariant streaming sequences probably have entropy rates equal to zero. Meanwhile, an independent sequence of streaming has an entropy rate with a value one as an example of coin tossing. On the basis of the above logic, they observed the past sequence of these streaming sequences and deducted new information of the outcome in the next streaming sequence.

3. Problem Statement and Background

Ultra High Definition TV [32] will become a standard of broadcasting and the internet streaming in the next few decades as we have a constantly growing number of subscribers as well as providers. It requires an internet speed of 25 Mbps or higher with compatible hardware. As a 4K video content provider, YouTube's 4K uses the VP9 compression codec instead of the commonly-used HEVC (High Efficiency Video Coding or H.265) [33] codec, which supports the platform and serves up high-quality. Even though, a user satisfies all the requirements of Ultra HD. It still suffers from the inherent network bandwidth fluctuation as internet users frequently join and leave an Internet Service Provider (ISP). In addition, a mobile user hardly watches the 4K contents because most of the mobile devices do not support 4K display. Thus, we must reduce the video resolution to capable video rates, which suit users' current network states and users' hardware configurations.

The above description is the principle of ABS. It is an approach of distributing multimedia content from a server to a client, an end-user. The approach is HTTP-based, which uses HTTP as the transmitting protocol to deliver the content over the networks. Thus, it receives many benefits from the HTTP-based technology such as packets that can traverse firewalls without having many difficulties in a Network Associate Translation (NAT). In addition, a server and a client do not require a persistent connection between them, and the server does not need to retain session information on the client.

If network bandwidth frequently fluctuates, the current implementation of the adaptive streaming at a client can handle such bandwidth fluctuation by setting the interval time between segment requests from a client. However, the method sometimes cannot choose representations appropriately. For example, in Figure 1, after a client requests a segment with a high-quality representation at t_1, bandwidth dramatically drops down to a very low bandwidth at t_2. Thus, the downloading duration of the segment t_h gets long. In contrast, after the client requests a segment with a low-quality representation, bandwidth spontaneously increases to a high value. Hence, the downloading duration of the segment t_l gets short. We realize that if a client can estimate the average of network bandwidth, then the request segment is in the medium bitrate.

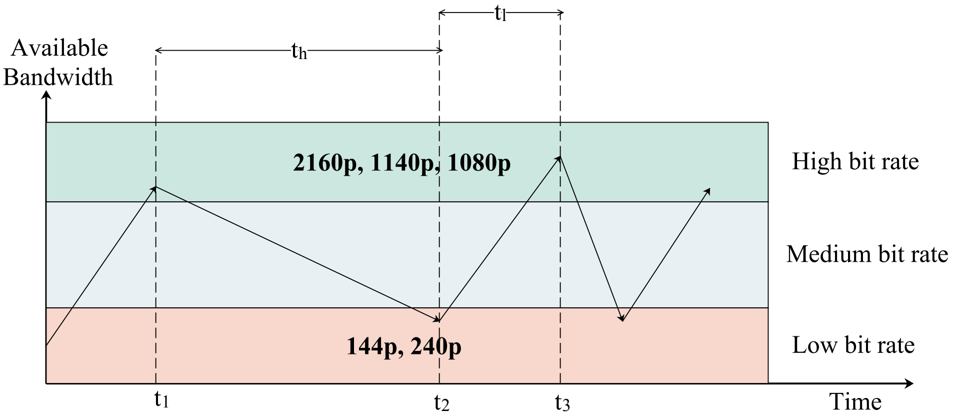

Figure 1. Change of bitrates over bandwidth fluctuation.

More specifically, in the high streaming video, such as 4K, we have a huge gap between two adjacent high-quality video representations. For example, the pixels per image of the video mode 2160p (4K) is twice the pixels per image of the video mode 2000p. Hence, the bandwidth requirement to stream the 4K video is multiple times of the division between the two rates, and twice in the example. Therefore, in this paper, we focus on maximizing the use of network bandwidth by reducing video resolution fluctuation, and we also investigate in order to infer the next optimal representations based on network past states.

4. Mathematics Modeling and Proposed Algorithm

Before modeling the problem of bandwidth fluctuations, we introduce two mathematic techniques. First, we use a moving average [34] to average bandwidth and buffer values of a client over a period. Secondly, we use fuzzy logic [35–38] to infer whether bandwidth increases or decreases compared to the current value. We then propose a fuzzy system to anticipate the next optimal representation for a client.

In network communication, we must consider many parameters, such as throughput, end-to-end delay, to ensure the stability of the network since the network is susceptible to fluctuations. Thus, we need to analyze these parameters. For example, within a certain amount of time, we collect a set of the parameters, such as throughput and a number of transmitted packages, in order to analyze data trend. On the basis of the trend, this analysis can predict the network conditions in advance. In this research, we use the moving average technique to analyze the series of bandwidth values because the method can reduce the fluctuations of data and emphasize the data trend as shown in Figure 2. It works by creating the series of average points of a data set. More specifically, it first averages the initial data set to get the first element. It then calculates the next element based on the last averaged data. The process continuously repeats over the time until we terminate the calculation. Several forms of the moving average are the simple average, and cumulative average, or weighted average forms. In this paper, we combine the cumulative moving average with a fuzzy logic method, which is described in the following.

$$\mu_q(x) = \begin{cases} 0, & x \leq q_1, \\ \frac{x-q_1}{q_2-q_1}, & q_1 \leq x \leq q_2, \\ \frac{q_3-x}{q_3-q_2}, & q_2 \leq x \leq q_3, \\ 0 & q_3 \leq x. \end{cases} \tag{1}$$

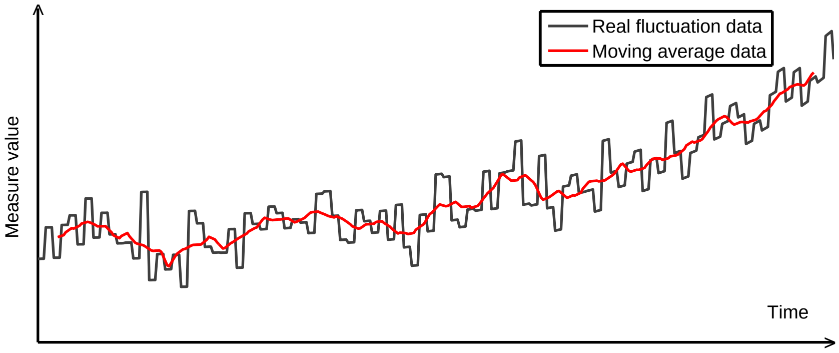

Figure 2. Moving average and time series data trend.

In mathematics, fuzzy sets [37] are sets whose elements have degrees of membership. In the application of the theory, fuzzy logic is a form of logic values that has the true values of variables are a real number between 0 and 1. The variables are managed by particular membership functions that are defined by experts or users. We employ the fuzzy to handle the concept of partial truth, where the truth value may range between completely true and completely false. In user-defined membership functions, each function has a range of value instead of a defined value. Fuzzy logic has been applied to many fields, from control theory to artificial intelligence.

The fuzzy system has three main calculation processes. First, we form the real input values of the system. We refer this process as Fuzzification. Then, we use user's or expert's experiential knowledge as if and then rules to calculate the output. We refer this process as Fuzzy Inference. Finally, the system defuses the input values to obtain the output system, which is the decision of the response segment, and this process is also known as Defuzzification. Figure 3 depicts the detail of the system.

In this research, we use the Mamdani fuzzy model [39]. A triangular fuzzy number represents a fuzzy linguistic value. It is triangle-shaped and can be viewed as a probability distribution. Suppose that we have three points $q = (q_1, q_2, q_3), q_1 \leq q_2 \leq q_3$; then, q is a triangular fuzzy number with membership function $\mu_q(x) \in [0,1]$ as shown in Equation (1).

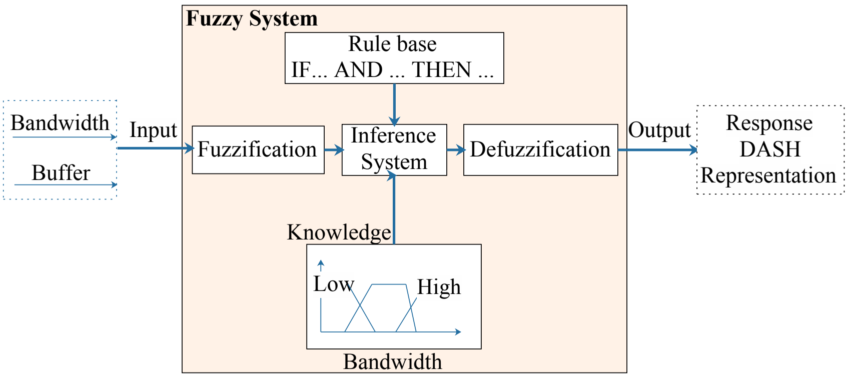

Figure 3. Overview of the proposed fuzzy system.

4.1. Fuzzification

In the adaptive streaming, we consider two important factors at a client: (1) bandwidth estimation; (2) buffer level. The quality of a video streaming reflects the quality of the network as presented by the bandwidth and the buffer level. For example, at a given time t, the bandwidth measurement of the client is b_t. A video has three resolution options for streaming, such as high, medium and low-resolution streaming, and their bandwidth requirements are b_h, b_m, b_l, respectively. Apparently, if $b_t > b_h$ and the buffer level is significantly high, then the client can retrieve the high resolution streaming and increase the buffer level.

Given buffer values at n different times $v_0, v_1, v_2, \ldots, v_n$, bandwidth values at those respective times are $b_0, b_1, b_2, \ldots, b_n$, and responses' representations are $r_0, r_1, r_2, \ldots, r_n$. After getting values at $n+1$, we calculate the bandwidth average W_{n+1}, the buffer average V_{n+1}, and the response representation average R_{n+1} by Equations (2)–(4):

$$W_{n+1} = W_n + \frac{b_{n+1} - W_n}{n+1}, \tag{2}$$

$$V_{n+1} = V_n + \frac{t_{n+1} - V_n}{n+1}, \tag{3}$$

$$R_{n+1} = R_n + \frac{r_{n+1} - R_n}{n+1}. \tag{4}$$

In the fuzzy logic modeling, we define seven linguistic sets as the following: (1) Negative High Difference—NHD; (2) Negative Medium Difference—NMD; (3) Negative Low Difference—NLD; (4) Difference—D; (5) Positive Low Difference—PLD; (6) Positive Medium Difference—PMD; (7) Positive High Difference—PHD. We might have more or less than the seven linguistics. It represents human perception based on each particular circumstance evaluation. In our research, we define the fuzzy center set (difference) and three fuzzy sets which are symmetrical on both sides of increasing and decreasing order of the center set. Throughout experience with video streaming, we recognized that the seven linguistics are enough to describe network state. To specify the degree of difference, we compare between average values and original values (measurement values) in a period. We then normalize those values to form input values of the fuzzy system.

Our logical base is network change status, such as bandwidth, fluctuates around average values of the network, which may be higher or smaller as shown in Figure 4. Network changes in the near past affect changes at some points in the future. On the basis of those changes, we can predict the next network metric values at a predetermined time. In addition, the trace-back is based on the logic that the current state of an event, which relates to its previous states, or previous states of a network cause the occurrence of an event.

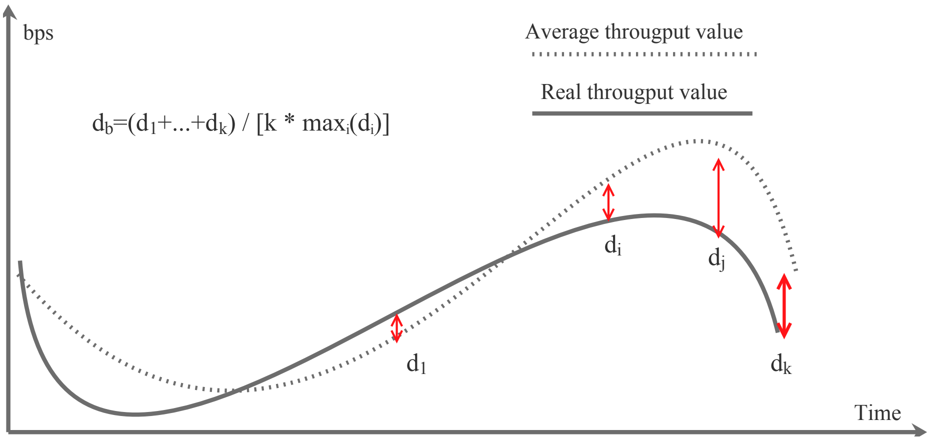

Figure 4. Base idea of estimating network change state.

We measure bandwidth, buffer level, and video quality representation each time a client makes a request to get a DASH segment from a server. We call each measurement a step. Suppose that we trace back within k steps $(k \in \mathbb{N}^+)$ from the current states of a network at n. We compare a subtraction between the moving average and original bandwidth/buffer value at the present time with the previous subtractions. We then summarize these quotients to form fuzzy input values. For example, we compare a subtraction at n with subtractions at $\{n-1,\ n-2, ...\ n-k\}$. The sum results in a rate that can be greater or less than one. Therefore, we normalize the quotients by comparing with the maximum different bandwidth/buffer. We obtain the result in a range between minus one and one. We define the degree of difference in Equations (5) and (6). Figure 5 depicts the triangular membership function of the linguistic values. We form the membership functions based on the calculated results from Equations (5) and (6):

$$d_b = \frac{1}{k} \sum_{i=n+1-k}^{n} \frac{b_i - W_i}{\max_{l=n+1-k,n}(|b_l - W_l|)}, \tag{5}$$

$$d_t = \frac{1}{k} \sum_{i=n+1-k}^{n} \frac{t_i - V_i}{\max_{l=n+1-k,n}(|t_l - V_l|)}. \tag{6}$$

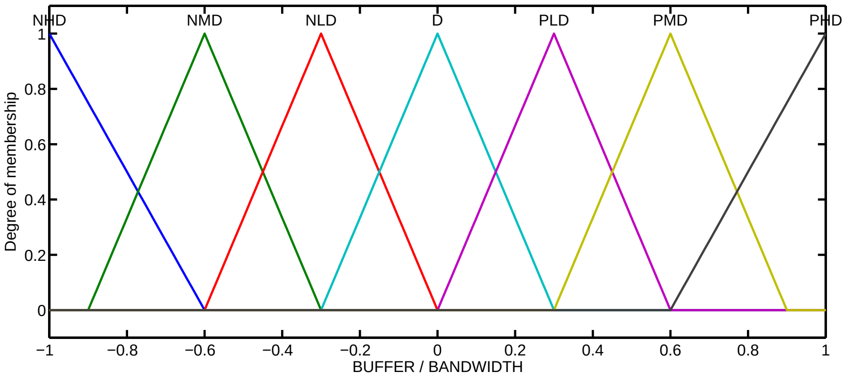

Figure 5. Input membership functions.

4.2. Fuzzy Inference

The fuzzy system has two input parameters that are bandwidth measurements and buffer values within k steps of a client. The output is a video quality representation, which aims to reduce video quality fluctuations and optimizes bandwidth utilization. In the system, we have total forty-nine rules to infer output from given inputs as shown in Table 1, where Bandwidth/Buffer represents fuzzy bandwidth and buffer numbers. As introduced in the fuzzy logic above, the inference rules are the expert knowledge or user experience. At the point of view of our experience at different video streaming scenarios, we use the experience to judge output inference based on input values. For example, we basically think that if network bandwidth is 50 Mbps, so that the network is adequate to have 4K video streaming.

Table 1. System fuzzy inference rules.

Bandwidth/Buffer	NHD	NMD	NLD	D	PLD	PMD	PHD
NHD	NHD	NHD	NMD	NMD	NMD	NLD	D
NMD	NHD	NMD	NMD	NMD	NLD	D	D
NLD	NMD	NLD	NLD	NLD	D	D	D
D	NLD	NLD	D	D	PLD	PLD	PLD
PLD	NLD	D	D	D	PLD	PLD	PLD
PMD	D	D	D	PLD	PMD	PMD	PMD
PHD	D	D	PLD	PLD	PMD	PMD	PHD

In more detail, if the bandwidth is Positive High Difference (PHD) and the buffer level is Positive Low Difference (PLD), then the response segment is Positive Medium Difference (PMD), or a client requests by "positively medium different", increasing a video resolution streaming compared to the previous values. More specifically, PHD means the bandwidth/buffer at present highly increases when we compare the increasing amount with the previous values. On the contrary, NHD means the bandwidth/buffer at the moment dramatically decreases compared to the decreasing amount of the previous values.

4.3. Fuzzy Defuzzification

In the above fuzzification process, we formed an input parameter by comparing the present difference value with the previous difference values within k steps. Therefore, the defuzzification process must inverse the input formulations. Suppose that d_r is a fuzzy output of the system at present. It can receive value from the set of the linguistic value PHD, PMD, PLD, D, NLD, NMD, and NHD. Figure 6 depicts output membership functions of the proposed system.

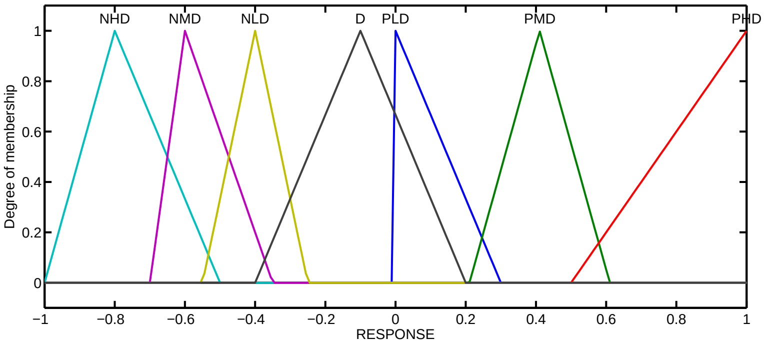

Figure 6. Output membership functions.

In the defuzzification process, the fuzzy output is calculated by the max-min inference method which is formed as Equation (7):

$$\mu_O = \max_i \left[\min_j \left[\mu_B^j, \mu_V^j \right] \right], \tag{7}$$

where μ_B, μ_V are membership functions of bandwidth and buffer level, respectively, and $i, j \in \{1, ..., 49\}$, $(i, j \in \mathbb{N}^+)$. We express the formulation as the following. We firstly calculate the minimum for both input values for each rule. We then maximize a total of forty-nine results and use a centroid triangle method to get output from the fuzzy system. Figure 7 depicts the correlation between input and output of the fuzzy system. It displays an increasing gradually from left to right. r_{n+1} is a real representation output after the defuzzification process. We calculate the representation by Equation (8), where R_{n+1} is calculated by Equation (4). In Equation (8), μ_O might greater or less than zero. Figure 8 depicts the above described defuzzification process:

$$r_{n+1} = r_n + \mu_O \frac{\sum_{i=n+1-k}^{n} (r_i - R_i)}{k}. \tag{8}$$

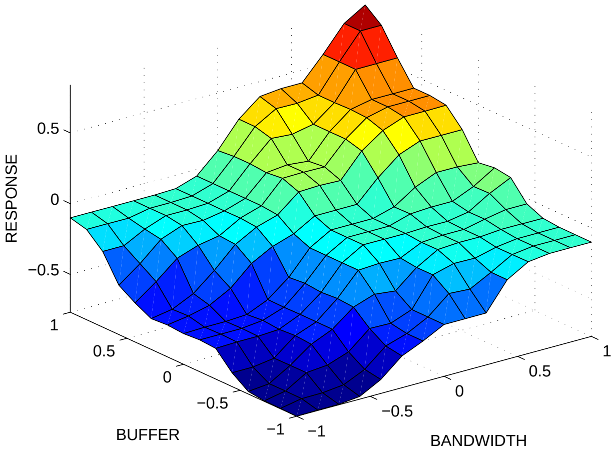

Figure 7. Correlation graph between input and output of the fuzzy system.

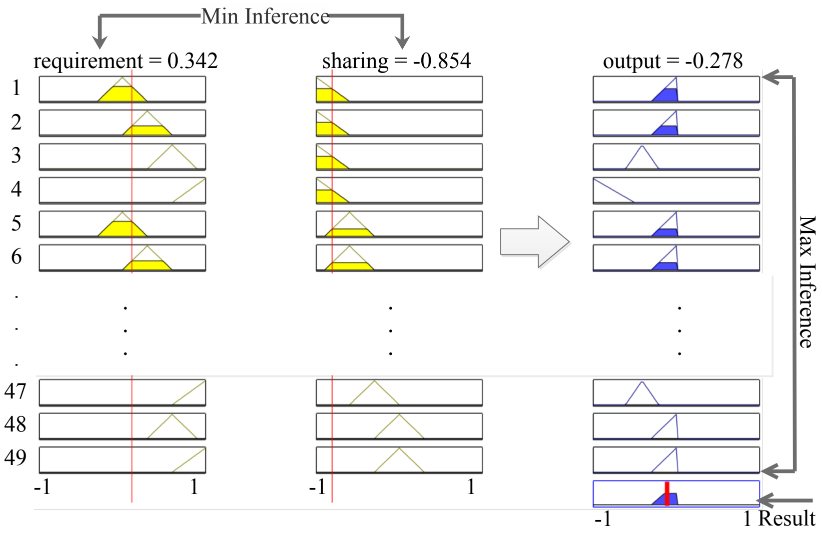

Figure 8. Defuzzification with forty-nine rules, two inputs and one output.

A list of representations is integers, but Equation (8) results in a real number. Thus, in the system implementation, we will take the value of a resolution that has a video width that is the nearest and smaller than the result of Equation (8).

Given a stochastic process $X = (X_1,\ X_2, ...,\ X_n)$, where $X_i, (i,\ n \in \mathbb{R})$ are discrete values in a domain $\mathcal{D}$. Its entropy rate $H(X)$ [40] is defined as:

$$H(X) = \lim_{n \to \infty} \frac{1}{n} H(X_1, X_2, ..., X_n), \tag{9}$$

where $H(X_1,\ X_2, ...\ X_n)$ is the entropy of random variables set, the limit exists, and $X = X_1,\ X_2, ...,\ X_n$ with joint probability $P(X_1,\ X_2, ...,\ X_n)$. It can be described as:

$$H(X_1,\ X_2, ...\ X_n) = -\sum_{X \in \mathcal{D}^n} P(X_1,\ X_2, ...,\ X_n) \log P(X_1,\ X_2, ...,\ X_n). \tag{10}$$

We use two Equations (9) and (10) to show that our proposed method leads to a lower entropy rate compared to other methods. In this research, we determine the parameter k so that it reduces the entropy rate, which describes as fluctuation values, such as bandwidth, representations, and buffer. Increasing k means that we reduce the variation of values and vice versa, and we make the average value that accurately depicts the trend of value change by reducing k. As mentioned in the last paragraph of Section 2, entropy rate of an invariant sequence has zero value. Therefore, we capture the changing trend of the network state by using moving average for entropy rate of a sequence because the entropy rate sometimes approaches zero as a result of a stable network. Then, we use this average entropy rate to control trace-back step with the above-mentioned reasoning for parameter k.

In conclusion of this section, we summarize the proposed method by an Algorithm 1 named the fuzzy moving DASH algorithm (FMDASH) as the following:

Algorithm 1: Fuzzy Representation Inference (Fuzzy Moving DASH Algorithm—FMDASH)

```
Data: bandwidth, buffer at present
Result: DASH - Representations
GetListOfResolution() in MPD file;
$k$ = a natural number //trace-back step;
while not end of video streaming do
    Calculate moving average of buffer and bandwidth value by using Equations (2)–(4);
    Fuzzification using Equations (5) and (6);
    Fuzzy Inference using Table 1;
    ouput = Defuzzification using Equation (7);
    $E_i$ = CalcualteEntropyRate() using Equation (10);
    $E_{average}$ = AverageEntropyRate() (using $(\sum_{j=0,i} E_j)/i$);
    if $E_i > E_{average}$ then
        $k = k - k/2$;
    else
        $k = k + k/2$;
    end
    for Resol in ResolutionList do
        if segment width at Equation (8) > width of Resol then
            next representation is Resol;
            break;
        end
    end
end
```

5. Experimentation and Evaluation

In this paper, we only focus the research on a mobile client. We implemented all the traffic measurements and calculations on the client. As addressed in Section 1, the proposed system is the DASH-based streaming system. The client detects bandwidth and CPU capacity in real time and accordingly adjusts the quality of a video stream along with bandwidth variations. In the experiment, we set up a system that has four components—first, a playback mobile client on Android-LibDash-based [41] where we implemented our proposed algorithm. Second, FFmpeg software [42] is used to encode a video source file into different video qualities. Third, MP4Box software [43] is to divide the encoded files into small segments and create a manifest file that links to the segment files. Fourth, a Node.js server stores uploaded files from a user. The server is responsible for executing the two software programs and providing video streaming for mobile clients.

We ran the implemented network system in two scenarios. First, we connect the Android mobile with a WiFi 5G to have a video streaming with the server. We then moved away and moved closer to the WiFi to have a bandwidth fluctuation. Second, the Android mobile has subscribed to 4G service in South Korea. We also moved the mobile around urban areas and remote areas for having bandwidth changes. In the MPD file, we encoded a source video with different video bitrates as follows: (video-frame-width video-frame-height birate (bps)): (80, 144, 196214), (134, 240, 295335), (202, 360, 394255), (270, 480, 493965), (404, 720, 1478490), (606, 1080, 2934159), and (810, 1440, 5843682). We set trace-back step is a natural random number in experiment comparisons.

First, we compared the proposed method with a method without fuzzy logic, CrystalBall [18]. In Figure 9, the blue line depicts buffer length of the proposed method while playing back video streaming. The buffer length values of the proposed method are higher than the red line, which is the statistic buffer level of the compared method. The red line reaches the zero value several times, which is more than the blue line. Thus, the compared method increased playing back the duration of the video streaming because the buffer level is depleted. Hence, the play back duration of our proposed method is shorter than the compared method. In addition, Figure 10 shows a variation of video representations in the experiment. The variation of our method does not vary much compared to the nonlinear method, which has streaming resolutions jumping from the maximum resolution (1440p) to the minimum resolution (144p) frequently. On the one hand, such frequent change in the resolution disturbs users because they can distinctly recognize the difference. On the other hand, reducing such change improves the stable of streaming and decreases the stream stalling, as the streaming time of our method is shorter than the nonlinear method.

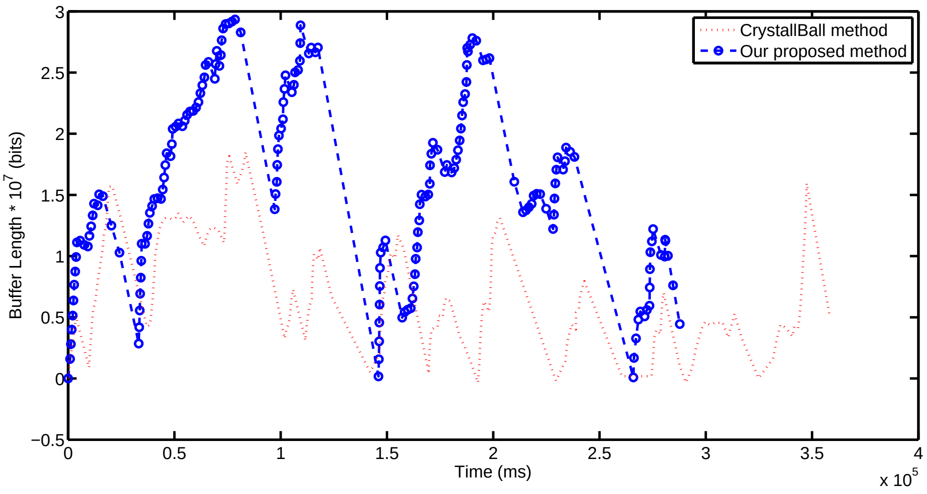

Figure 9. Variation of buffer length.

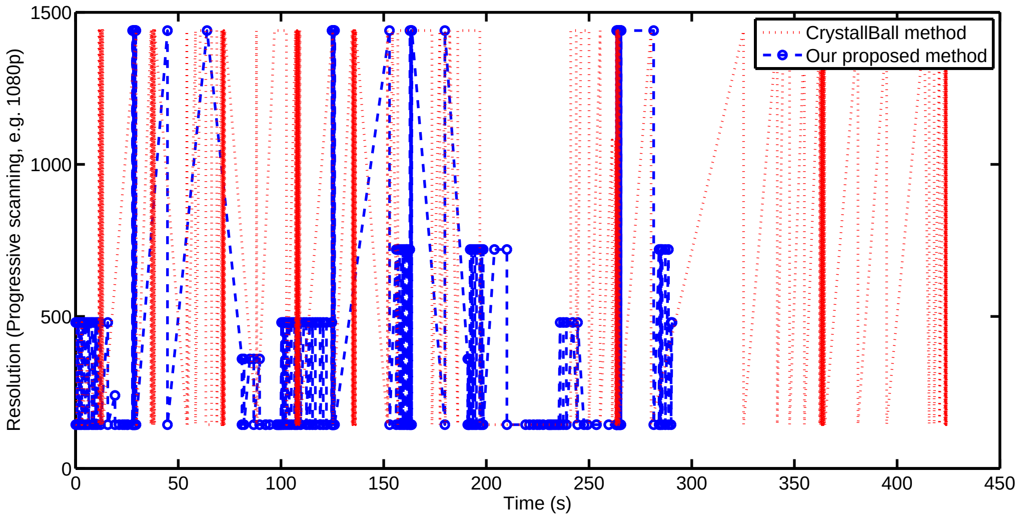

Figure 10. Variation of video representations in adaptive streaming.

In addition, we also compared our approach with another fuzzy-based method, FDASH (Fuzzy-Based MPEG/DASH Adaptation Algorithm) [8,44]. We re-used their simulation on NS3 and also implemented our proposal method on the source code. As a result, our proposed method reduced the variations of video frame width in Figure 11 and variation of video frame height in Figure 12. In addition, for simplicity, we define bandwidth utilization as a product between width and height of a video. The product represents a number of downloaded bits of a video streaming session. As a result, we got the utilization of our proposed method higher than the compared method at about 40%, though the buffer level of both the compared and proposed method is not much different, as shown in Figure 13. FDASH buffer variation is similar to our proposed method buffer variation as well as time for using a buffer because both methods employ the fuzzy logic to control the network uncertainty such as network fluctuation.

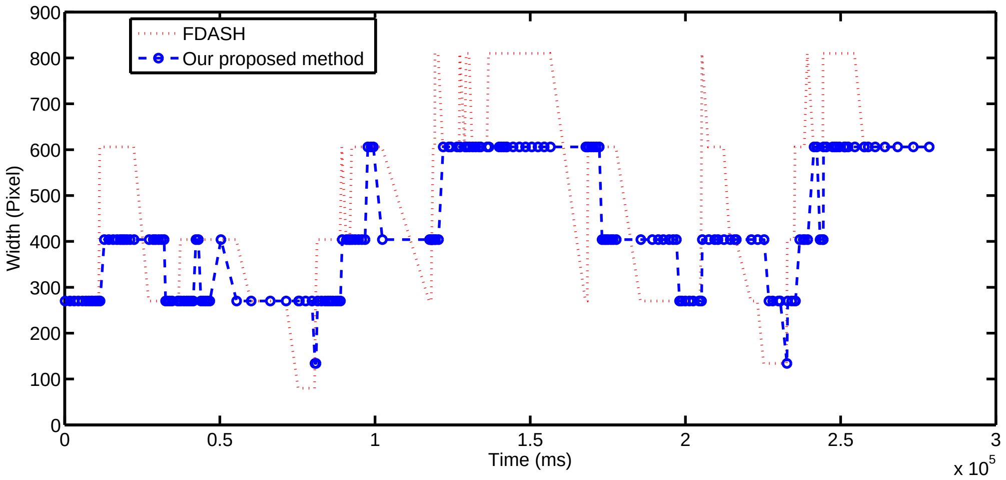

Figure 11. Video width variations.

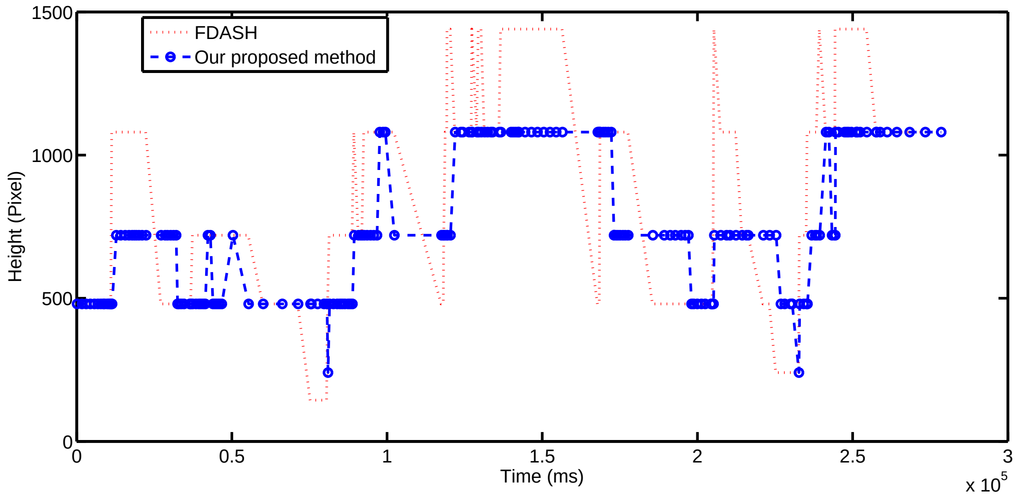

Figure 12. Video height variations.

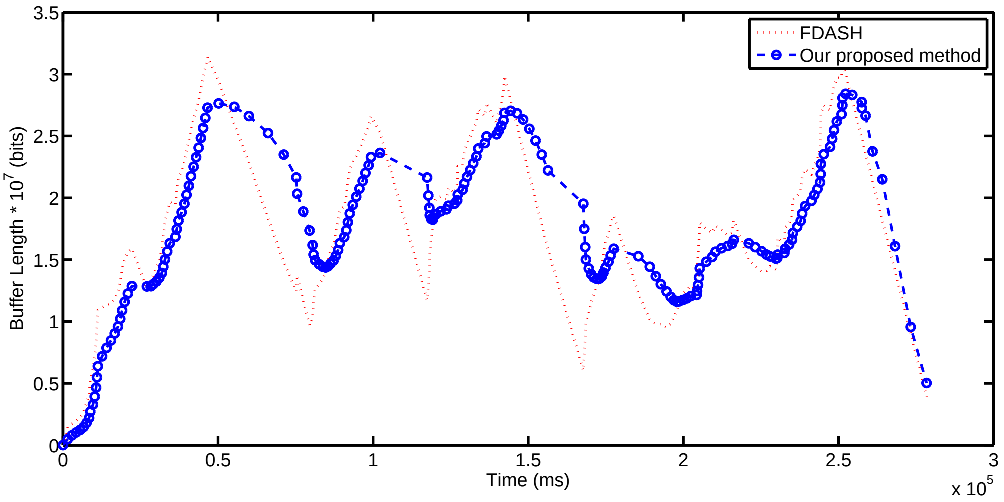

Figure 13. Comparison of buffer variation with another fuzzy-based method.

Finally, we used the above entropy Equations (9) and (10) to calculate the randomness of the sequence of the measurement values, and we got the results as shown in Table 2. Our proposed method (FMDASH) has a lower entropy rate at the buffer values and the fluctuation of segment response than the nonlinear method and FDASH. Apparently, our approach entropy rate on the left and right sides of Table 2 are different since we tested in two scenarios, which are real network tests to compare with nonlinear and NS3 test to compare with FDASH. Recall that lower rate of entropy indicates the stable of sequence. Thus, our method not only improves bandwidth utilization but also archives stable streaming than the nonlinear method.

Table 2. Entropy rate values.

Fluctuation Value	Method	Entropy Rate	Method	Entropy Rate
Buffer	FMDASH	7.5048	FMDASH	7.6221
	CrytallBall	7.5157	FDASH	7.6221
Segment	FMDASH	1.2965	FMDASH	1.6715
	CrytallBall	1.7769	FDASH	2.2047

The important factor in our approach is the look of network changes in the near past. The factor is the parameter k. If k is large, we refer to the long-term past. Therefore, the reference does not adapt to network fluctuation in the short term. In contrast, if k is small, we only rely on small network changes to infer the next state of the network. Therefore, the choice of the parameter k is an important determinant of the method effectiveness. More specifically, we tested the k parameter with different values in the two mentioned scenarios without using entropy rates that control the length of trace-back step. Regardless of entropy rate, we got different results with different input trace-back values in the experiment as shown in Figure 14. In addition, we achieved $k = 6$ in the experiment using an entropy rate without considering the choice of the initial trace-back step. Therefore, we concluded that the trace-back step equals $k = 6$ in our experimental scenarios and is the optimal value to reduce the frequent change of the network while enhancing the video quality compared to other values.

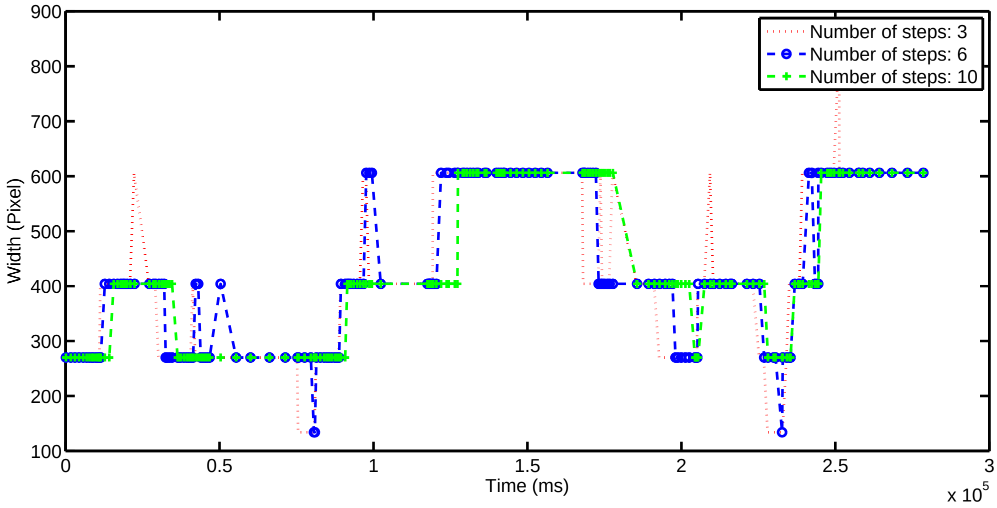

Figure 14. Simulation with different numbers of trace-back steps.

6. Conclusions

In this paper, we presented the fuzzy-based adaptive streaming method to reduce the mobile video quality fluctuations in DASH by minimizing the fluctuation entropy rate. We exploited two mathematics techniques, such as moving average and fuzzy logic. First, we found the average of previous fluctuation points by using the moving average calculation. Second, we trace-backed at certain steps of the last variation points to infer the degree of current network fluctuation. We then inferred the following representation request for a client. The experiment results showed that our proposal lowered the fluctuation of video quality, and improved bandwidth utilization by about 40% compared to existing methods.

The following points summarize our contributions in this research:

- We proposed an adaptive algorithm in DASH for mobile devices.
- Our algorithm reduced video quality fluctuations and improved network utilization compared to existing methods in DASH.

For future work, we intend to combine the proposed method with video compression techniques, such as H.264, to reduce the amount of transmitted and received bits between receivers and senders over the internet. We also plan to research the initial season of DASH cooperating with the proposed method. Furthermore, the question of choosing trace-back steps has not thoroughly been answered yet in this research. Therefore, we will put our effort towards employing mathematics optimization methods in order to find the optimal step for each network condition.

Acknowledgments: This research was supported by the IT R&D program of the MSIT (Ministry of Science and ICT), Korea/NIPA (National IT Industry Promotion Agency) 12221-14-1001, Next Generation Network Computing Platform Testbed, and was also supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2016-0-00314) supervised by the IITP (Institute for Information & communications Technology Promotion). Additionally, this research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST) (Grant No. NRF-2017R1D1A1B03034429). Furthermore, this work was supported by 'The Cross-Ministry Giga KOREA Project' grant from the Ministry of Science, ICT and Future Planning, Rep. of Korea(GK16P0100, Development of Tele Experience Service SW Platform based on Giga Media).

Author Contributions: Linh Van Ma and Hoyong Ryu conceived and designed the experiments; Linh Van Ma and Jinsul Kim performed the experiments; Jiseung Nam and Jaehyung Park analyzed the data; Hoyong Ryu contributed reagents/materials/analysis tools; Linh Van Ma and Jinsul Kim wrote the paper. All authors have read and approved the final manuscript.

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

References

1. Schulzrinne, H.; Rao, A.; Lanphier, R.; Westerlund, M.; Stiemmerling, M. Available online: https://tools.ietf.org/html/draft-ietf-mmusic-rfc2326bis-27 (accessed on 10 September 2011).
2. Garcia, M.N.; De Simone, F.; Tavakoli, S.; Staelens, N.; Egger, S.; Brunnström, K.; Raake, A. Quality of experience and HTTP adaptive streaming: A review of subjective studies. In Proceedings of the 2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX), Singapore, 18–20 September 2014; pp. 141–146.
3. Seufert, M.; Egger, S.; Slanina, M.; Zinner, T.; Hobfeld, T.; Tran-Gia, P. A survey on quality of experience of HTTP adaptive streaming. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 469–492.
4. Vergados, D.J.; Michalas, A.; Sgora, A.; Vergados, D.D. A fuzzy controller for rate adaptation in MPEG-DASH clients. In Proceedings of the 2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), Washington, DC, USA, 2–5 September 2014; pp. 2008–2012.
5. Kua, J.; Armitage, G.; Branch, P. A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming over HTTP. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1842–1866.
6. 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.
7. Jiang, J.; Sekar, V.; Zhang, H. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE. In Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, Nice, France, 10–13 December 2012.
8. Sobhani, A.; Yassine, A.; Shirmohammadi, S. A fuzzy-based rate adaptation controller for DASH. In Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, Portland, OR, USA, 18–20 March 2015.
9. Berbey Alvarez, A.; Merchan, F.; Calvo Poyo, F.J.; Caballero George, R.J. A Fuzzy Logic-Based Approach for Estimation of Dwelling Times of Panama Metro Stations. *Entropy* **2015**, *17*, 2688–2705.
10. Linh Van Ma, S.P.; Park, J.; Nam, J.; Jang, J.; Kim, J. A Fuzzy-based Method for Reducing Mobile Video-quality Fluctuation. *Int. J. Mob. Device Eng.* **2017**, *1*, 21–28.

11. Tilkov, S.; Vinoski, S. Node. js: Using JavaScript to build high-performance network programs. *IEEE Internet Comput.* **2010**, *14*, 80–83.
12. Begen, A.; Akgul, T.; Baugher, M. Watching video over the web: Part 1: Streaming protocols. *IEEE Internet Comput.* **2011**, *15*, 54–63.
13. Xiong, P.; Shen, J.; Wang, Q.; Jayasinghe, D.; Li, J.; Pu, C. NBS: A network-bandwidth-aware streaming version switcher for mobile streaming applications under fuzzy logic control. In Proceedings of the 2012 IEEE First International Conference on Mobile Services (MS), Honolulu, HI, USA, 24–29 June 2012; pp. 48–55.
14. Kuschnig, R.; Kofler, I.; Hellwagner, H. An evaluation of TCP-based rate-control algorithms for adaptive Internet streaming of H. 264/SVC. In Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems, Phoenix, AZ, USA, 22–23 February 2010.
15. Jammeh, E.A.; Fleury, M.; Wagner, C.; Hagras, H.; Ghanbari, M. Interval type-2 fuzzy logic congestion control for video streaming across IP networks. *IEEE Trans. Fuzzy Syst.* **2009**, *17*, 1123–1142.
16. Bouzakaria, N.; Concolato, C.; Le Feuvre, J. Overhead and performance of low latency live streaming using MPEG-DASH. In Proceedings of the 5th International Conference on Information, Intelligence, Systems and Applications (IISA), Chania, Greece, 7–9 July 2014; pp. 92–97.
17. Toni, L.; Aparicio-Pardo, R.; Simon, G.; Blanc, A.; Frossard, P. Optimal set of video representations in adaptive streaming. In Proceedings of the 5th ACM Multimedia Systems Conference, Singapore, 19 March 2014.
18. Mangla, T.; Theera-Ampornpunt, N.; Ammar, M.; Zegura, E.; Bagchi, S. Video through a crystal ball: Effect of bandwidth prediction quality on adaptive streaming in mobile environments. In Proceedings of the 8th International Workshop on Mobile Video, Klagenfurt, Austria, 10–13 May 2016.
19. Dubin, R.; Dvir, A.; Pele, O.; Hadar, O.; Katz, I.; Mashiach, O. Adaptation logic for HTTP dynamic adaptive streaming using geo-predictive crowdsourcing for mobile users. *Multimed. Syst.* **2016**, 1–13, doi:org/10.1007/s00530-016-0525-6.
20. Zhao, S.; Li, Z.; Medhi, D.; Lai, P.; Liu, S. Study of user QoE improvement for dynamic adaptive streaming over HTTP (MPEG-DASH). In Proceedings of the 2017 International Conference on Computing, Networking and Communications (ICNC), Santa Clara, CA, USA, 26–29 January 2017; pp. 566–570.
21. De Vleeschauwer, D.; Viswanathan, H.; Beck, A.; Benno, S.; Li, G.; Miller, R. Optimization of HTTP adaptive streaming over mobile cellular networks. In Proceedings of the 2013 Proceedings IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 898–997.
22. Han, B.; Wang, X.; Choi, N.; Kwon, T.; Choi, Y. AMVS-NDN: Adaptive mobile video streaming and sharing in wireless named data networking. In Proceedings of the 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Turin, Italy, 14–19 April 2013; pp. 375–380.
23. Wang, X.; Chen, M.; Kwon, T.T.; Yang, L.; Leung, V.C. AMES-cloud: A framework of adaptive mobile video streaming and efficient social video sharing in the clouds. *IEEE Trans. Multimed.* **2013**, *15*, 811–820.
24. Mok, R.K.; Li, W.; Chang, R.K. IRate: Initial video bitrate selection system for HTTP streaming. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 1914–1928.
25. Yan, Z.; Chen, C.W. RnB: Rate and brightness adaptation for rate-distortion-energy tradeoff in HTTP adaptive streaming over mobile devices. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 3–7 October 2016.
26. Zhou, C.; Lin, C.W.; Guo, Z. mDASH: A markov decision-based rate adaptation approach for dynamic HTTP streaming. *IEEE Trans. Multimed.* **2016**, *18*, 738–751.
27. Zhou, C.; Zhang, X.; Huo, L.; Guo, Z. A control-theoretic approach to rate adaptation for dynamic HTTP streaming. In Proceedings of the 2012 IEEE Visual Communications and Image Processing (VCIP), San Diego, CA, USA, 27–30 November 2012; pp. 1–6.
28. De Cicco, L.; Caldaralo, V.; Palmisano, V.; Mascolo, S. Elastic: A client-side controller for dynamic adaptive streaming over http (dash). In Proceedings of the 2013 20th International Packet Video Workshop (PV), San Jose, CA, USA, 12–13 December 2013; pp. 1–8.
29. Mok, R.K.; Luo, X.; Chan, E.W.; Chang, R.K. QDASH: A QoE-aware DASH system. In Proceedings of the 3rd Multimedia Systems Conference, Chapel Hill, NC, USA, 22–24 February 2012.
30. Vergados, D.J.; Michalas, A.; Sgora, A.; Vergados, D.D. A control-based algorithm for rate adaption in MPEG-DASH. In Proceedings of the 5th International Conference on Information, Intelligence, Systems and Applications (IISA), Chania, Greece, 7–9 July 2014; pp. 438–442.

31. Petkov, V.; Rajagopal, R.; Obraczka, K. Characterizing per-application network traffic using entropy. *ACM Trans. Model. Comput. Simul. (TOMACS)* **2013**, *23*, 14.
32. Namiki, S.; Kurosu, T.; Tanizawa, K.; Kurumida, J.; Hasama, T.; Ishikawa, H.; Nakatogawa, T.; Nakamura, M.; Oyamada, K. Ultrahigh-definition video transmission and extremely green optical networks for future. *IEEE J. Sel. Top. Quantum Electron.* **2011**, *17*, 446–457.
33. Sullivan, G.J.; Ohm, J.; Han, W.J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668.
34. Pugachev, V.S. *Probability Theory and Mathematical Statistics for Engineers*; Elsevier: Amsterdam, The Netherlands, 2014.
35. Lootsma, F.A. *Fuzzy Logic for Planning and Decision Making*; Springer: Berlin, Germany, 2013; Volume 8.
36. Gerla, G. *Fuzzy Logic: Mathematical Tools for Approximate Reasoning*; Springer: Berlin, Germany, 2013; Volume 11.
37. Gottwald, S. *Fuzzy Sets and Fuzzy Logic: The Foundations of Application—From a Mathematical Point of View*; Springer: Berlin, Germany, 2013.
38. Cintula, P.; Fermüller, C.; Noguera, C. *Handbook of Mathematical Fuzzy Logic*; College Publications: London, UK, 2015; Volume 3.
39. Yager, R.R.; Zadeh, L.A. *An Introduction to Fuzzy Logic Applications in Intelligent Systems*; Springer: Berlin, Germany, 2012; Volume 165.
40. Chliamovitch, G.; Dupuis, A.; Chopard, B. Maximum entropy rate reconstruction of Markov dynamics. *Entropy* **2015**, *17*, 3738–3751.
41. Mueller, C.; Lederer, S.; Poecher, J.; Timmerer, C. Demo paper: Libdash-an open source software library for the mpeg-dash standard. In Proceedings of the 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), San Jose, CA, USA, 15–19 July 2013; pp. 1–2.
42. FFmpeg. Available online: http://ffmpeg.org (accessed on 7 September 2017).
43. GPAC, Multimedia Open Source Project. Available online: https://gpac.wp.imt.fr/ (accessed on 7 September 2017).
44. Vergados, D.J.; Michalas, A.; Sgora, A.; Vergados, D.D.; Chatzimisios, P. FDASH: A Fuzzy-Based MPEG/DASH Adaptation Algorithm. *IEEE Syst. J.* **2016**, *10*, 859–868.