

RESEARCH

Open Access



# ANFIS-based rate adaptation scheme for adaptive streaming over HTTP

Ye Seul Son , Hyun Jun Kim and Joon Tae Kim\*

## Abstract

Recently, HTTP-based video streaming traffic has continued to increase. Therefore, video service providers have been using HTTP-based adaptive streaming (HAS) technology to reduce the traffic load of the HTTP server. Accordingly, many adaptive bit rate (ABR) schemes have been proposed to provide a high quality of experience (QoE) to video service clients. In this paper, we propose a new ABR scheme using an adaptive network-based fuzzy inference system (ANFIS), which is one of the neuro-fuzzy structures. The proposed scheme learns optimal fuzzy parameters by using (1) the learning ability of ANFIS and (2) the video streaming data providing high QoE to clients. Then, the bit rate of the next segment is determined according to these trained parameters. In the simulation using NS-3, we show that the proposed scheme selects the appropriate bit rate under various wireless network conditions and provides better QoE to clients than the existing schemes.

**Keywords:** HAS technology, MPEG-DASH, ABR algorithm, Adaptive streaming, ANFIS, Neuro-fuzzy system

## 1 Introduction

In recent years, HTTP-based video streaming traffic has consistently increased. It is expected to increase more than ever due to the expansion of streaming audio and video adoption and the emergence of state-of-the-art video technologies such as 4K, high-dynamic-range video, and virtual reality [1]. This traffic growth has led to the adoption of HTTP-based adaptive streaming (HAS) technology by video service providers to reduce the traffic load of the HTTP server. HAS technology encodes video at multiple bit rates and divides the video at each bit rate into multiple segments with a fixed playback duration. Further, data such as media representation, segment duration, and segment URLs are stored as a file in the HTTP server. The clients download the file before the video starts and sequentially request a segment at the appropriate bit rate by using the file to download and play the segment. Thus, HAS technology reduces the traffic load on the server and saves bandwidth resources by sending only a single segment for each client request rather than the entire video [2].

The dynamic streaming over HTTP (DASH) standard [3, 4] developed by MPEG is one of the most popular

HAS technologies. DASH has been adopted by major video service providers such as YouTube and Netflix. As DASH is widely used, various methods have been proposed to optimize its service. In [5], a quality of experience (QoE)-aware wireless resource allocation method considering DASH clients has been proposed. The proposed method reduces rebuffering time by considering the buffer level of DASH clients. It, however, does not consider non-DASH clients. Therefore, a new method considering both DASH clients and non-DASH clients has been proposed in [6]. This method reduces the rebuffering time and guarantees fairness among non-DASH clients. The authors in [7] introduce a new transmission architecture for DASH video streaming over the LTE system. They calculate the importance of each packet and propose a packet scheduling strategy and resource allocation method based on the calculated importance of each packet. In addition, they also proposed a rate adaption scheme based on the MAC queue, the client buffer, and the estimated transmission rate.

Because DASH does not define how the client selects the bit rate of the next segment, many adaptive bit rate (ABR) algorithms have been proposed to select the next bit rate [8–12]. These algorithms determine the next bit rate based on parameters such as the TCP throughput estimated by the application layer [8], current

\* Correspondence: [jtkim@konkuk.ac.kr](mailto:jtkim@konkuk.ac.kr)

Department of Electronic Engineering, Konkuk University, Seoul, Republic of Korea

playback buffer level [9, 10], or both [11, 12]. The main goal of these ABR algorithms is to provide the clients with high QoE. In order to provide high QoE, ABR algorithms must meet various conditions such as prevention of rebuffering events, minimizing video start delays, maximizing video quality, minimizing the number of video quality switching events, and smoothly changing the video quality [13–16]. However, all of these requirements cannot be met because they have trade-offs with each other [17]. Therefore, the goal of ABR algorithms is to provide clients with high QoE by appropriately meeting each condition to the extent possible.

One of the recently proposed ABR schemes, FDASH [11] is a fuzzy logic-based bit rate determination technique. The fuzzy logic controller (FLC) of FDASH receives the current playback buffer level and the differential of the client buffer level. Then, the FLC performs fuzzification, fuzzy rule-based inference, and de-fuzzification to create output  $f$ , which represents the increase/decrease factor of the segment bit rate. FDASH is better in terms of average video quality and the number of video quality switching compared to other existing ABR schemes. It, however, is not explained why the fuzzy parameters used for fuzzification and de-fuzzification in the FLC are set. It is therefore not clear whether these set fuzzy parameters are optimal.

In this paper, we propose a new ABR scheme using adaptive network-based fuzzy inference system (ANFIS) [18], one of the neuro-fuzzy system structures. The neuro-fuzzy system is a hybrid intelligence system that combines a fuzzy system and an artificial intelligence network. It can find optimal fuzzy membership functions via neural network learning using a training data set [19]. The proposed scheme uses the video streaming data (training data) to provide high QoE to DASH clients. The training data are extracted from streaming cases that provide high QoE to DASH clients and provides a basis for setting up the fuzzy parameters of

ANFIS. With the trained fuzzy parameters, the proposed scheme can improve the performance of the video streaming service by selecting the appropriate bit rate of the next segment.

Finally, we show that the proposed scheme provides higher QoE performance compared to the existing schemes in the time-varying wireless network environment using NS-3 simulation. In this environment, the proposed scheme shows that the number of video quality switching is similar to FDASH, that the average segment bit rate is about 25% higher than that in FDASH, and that the QoE is increased by nearly 24% compared to FDASH.

The rest of the paper is organized as follows: Section 2 describes ANFIS structure, and Section 3 presents a new ABR scheme using ANFIS. In Section 4, we verify the performance of the proposed method through simulation, and we finally conclude the paper in Section 5.

### 2 ANFIS structure

The neuro-fuzzy system is a combination of a fuzzy system and a neural network. The nodes in the neuro-fuzzy system operate as FLC. It also automatically learns the appropriate fuzzy parameters according to the training data provided [19].

ANFIS [18] is one of the structures of this neuro-fuzzy system, which was designed to functionally resemble the Sugeno-type fuzzy model [20]. As shown in Fig. 1, ANFIS consists of five layers. The ANFIS shown in Fig. 1 contains the single output  $z$  with two input values  $x, y$ , and the operation is as follows. First, for the input values  $x$  and  $y$ , the first layer performs fuzzification via a fuzzy membership function. In fuzzification, the output expression varies depending on the fuzzy membership function selected. For example, the output of the first layer for the  $i$ th node is expressed by Eq. (1) if the fuzzy membership function is selected by the Gaussian membership function and

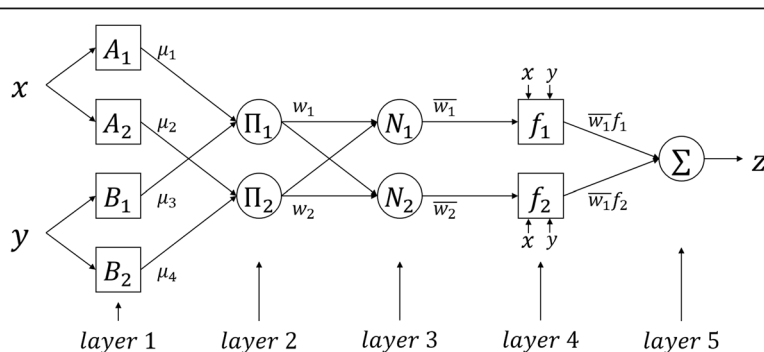


Fig. 1 The ANFIS architecture

**Table 1** Hybrid learning process

	Forward path	Backward path
Premise parameters	Fixed	Gradient descent
Consequent parameters	Least square method	Fixed
Reference signals	Node outputs	Error rates

by Eq. (2) if the fuzzy membership function is selected by the bell membership function.

$$\mu_i(x) = \exp\left(-\frac{(x-c_i)^2}{2\sigma_i^2}\right), i = 1, 2, 3, 4 \tag{1}$$

$$\mu_i(x) = \frac{1}{1 + \left|\frac{x-c_i}{a_i}\right|^{2b_i}}, i = 1, 2, 3, 4 \tag{2}$$

where  $\{c_i, \sigma_i\}$  in Eq. (1) and  $\{a_i, b_i, c_i\}$  in Eq. (2) are referred to as premise parameters and are learned using training data. As the value of these parameters changes, the outputs of the fuzzy membership functions vary accordingly.

The nodes in the second layer correspond to fuzzy rules. The output of the second layer is the product of all the input signals connected to each node. For example, in Fig. 1, the output of the first node is expressed by Eq. (3):

$$w_1 = \mu_1 \times \mu_3 \tag{3}$$

The third layer performs normalization of the output of the second layer. The  $i$ th node of the third layer calculates the ratio of the  $i$ th output value of the second layer to the sum of all output values of the second layer. It is expressed by Eq. (4):

$$\bar{w}_i = \frac{w_i}{\sum_{j=1}^n w_j}, i = 1, 2 \tag{4}$$

where  $n$  is the number of nodes in the second layer.

The fourth layer performs de-fuzzification via an output fuzzy membership function. The output of the fourth layer for the  $i$ th node is determined as

$$\bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i), i = 1, 2 \tag{5}$$

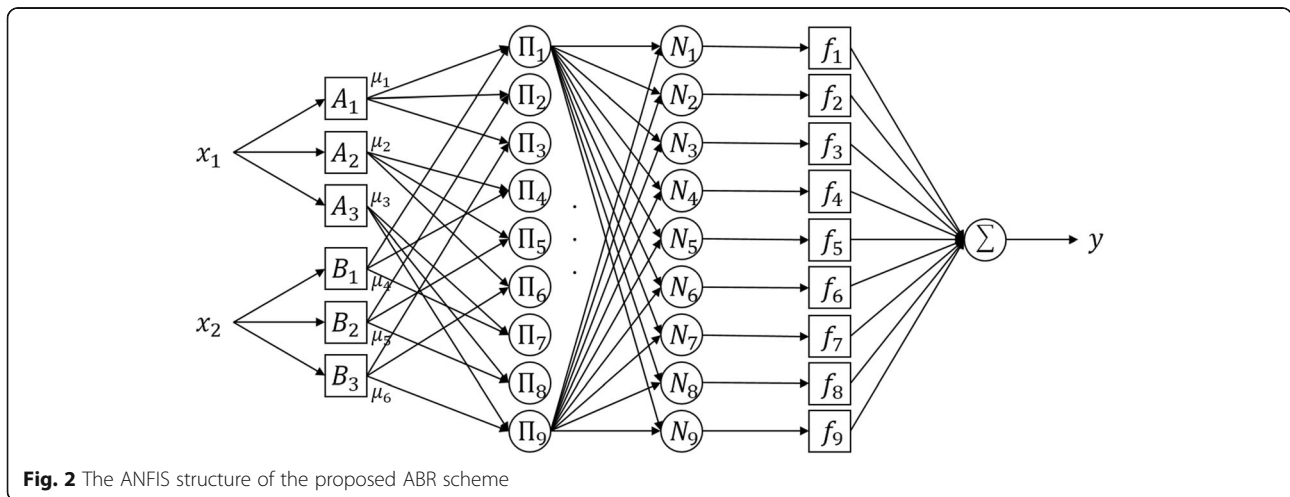
where  $\{p_i, q_i, r_i\}$  are referred to as consequent parameters and are learned using training data. The Sugeno-type fuzzy model uses a linear function or a constant function as an output fuzzy membership function. The output membership function of Eq. (5) is a linear function, but when  $p_i$  and  $q_i$  are zero, it becomes a constant function.

Finally, the node in the fifth layer adds all the output values of the fourth layer to generate the final output  $z$ , as shown in Eq. (6):

$$z = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{6}$$

The hybrid learning process [18], which combines the least square method and the gradient descent method, is used to train the premise parameters and the consequent parameters. The learning process of the hybrid learning process is summarized in Table 1.

In the forward path, the hybrid learning process fixes the premise parameters and trains the consequent parameters by using the least square method. Conversely, in the backward path, the process fixes the consequent parameters and trains the premise parameters by using the gradient descent method. The process iterates the least square and gradient descent methods and is terminated when the error between the output value of the training data and the output



**Fig. 2** The ANFIS structure of the proposed ABR scheme

**Table 2** The ANFIS properties of the proposed scheme

Number of input values	2
Number of output values	1
Number of membership functions per input	3
Number of fuzzy rules	9
Type of input membership function	Gaussian membership function
Type of output membership function	Constant function

value of the ANFIS meets the specified value or when the process reaches the specified iteration number.

### 3 Methods

In this section, we propose a new ABR scheme that provides high QoE to clients in DASH environment using the ANFIS structure described in Section 2. In the FDASH scheme mentioned in Section 1, the current playback buffer level and the differential of the client buffer level are used as the inputs of the FLC. However, the proposed method determines the next bit rate based on the current playback buffer level and the estimated download time, which is advantageous because the input values of the FLC of FDASH do not consider network throughput. A detailed explanation is as follows.

#### 3.1 Proposed ABR scheme

The ANFIS structure of the proposed ABR scheme is shown in Fig. 2, and its corresponding properties are summarized in Table 2. The proposed structure contains two inputs,  $x_1$  and  $x_2$ , and output  $y$ . The first input parameter  $x_1$  is the current playback buffer level, and the

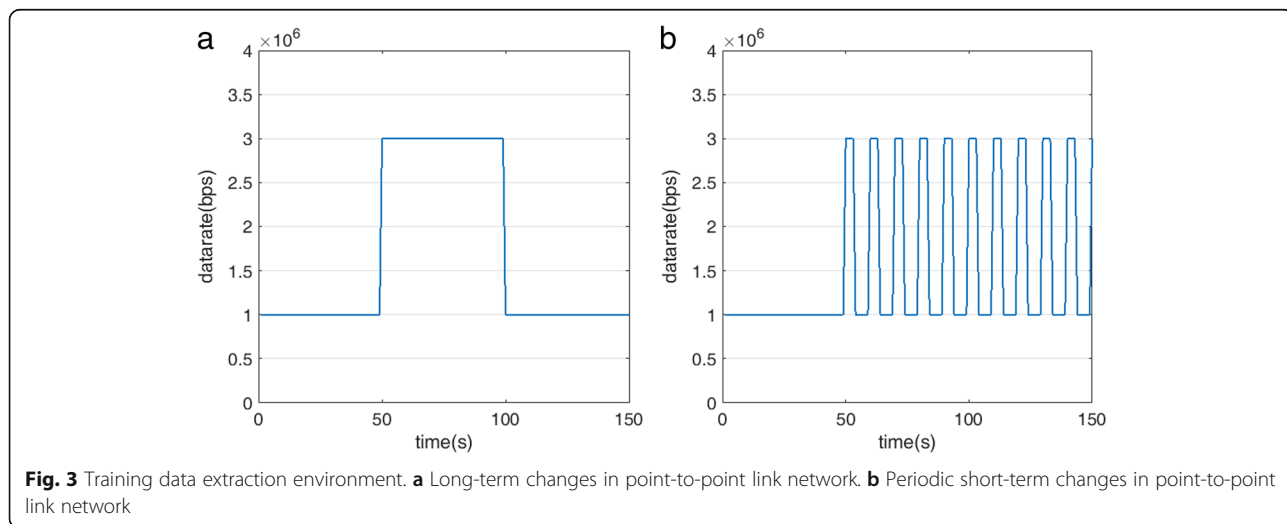
second input parameter  $x_2$  is the estimated download time. The estimated download time  $T_{next}$  is calculated using the size of the next segment  $S_{next}$  and the network throughput  $R$  as expressed in Eq. (7):

$$T_{next} = \frac{S_{next}}{R} \tag{7}$$

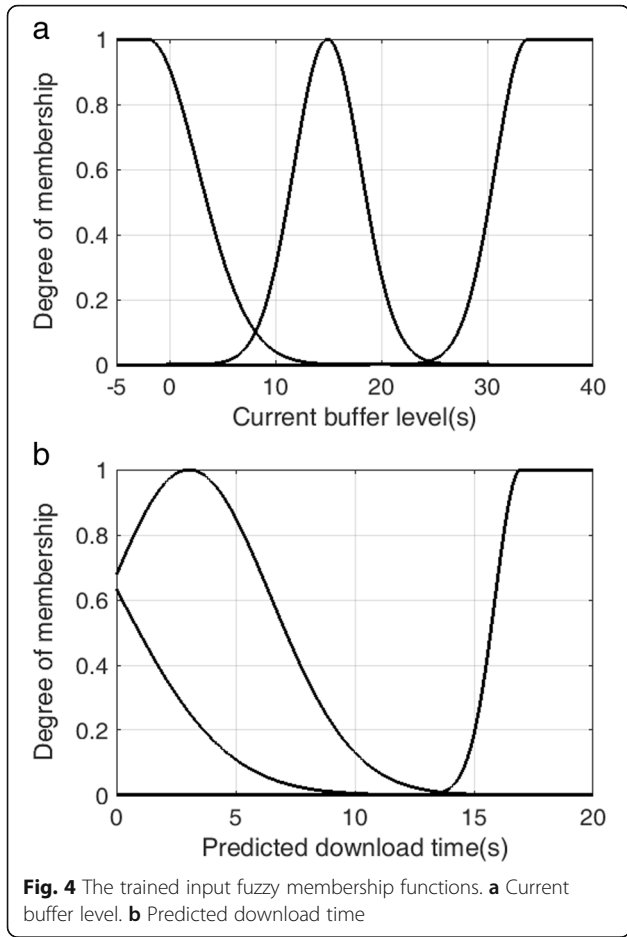
$$R = \frac{S_{last}}{T_{last}} \tag{8}$$

where  $S_{last}$  and  $T_{last}$  represent the size of the last segment and the actual download time for the last segment, respectively.

Because the network throughput fluctuates significantly over time, calculating the network throughput with only the size of the last segment and the download time can lead to substantial errors. Therefore, ABR schemes using the network throughput usually use average network throughput for a period [8]. However, this causes a major error in the calculated throughput when non-stationary throughput changes (e.g., level shift) occur. Therefore, the previous throughput is ignored in this case [21]. Similarly, the



**Fig. 3** Training data extraction environment. **a** Long-term changes in point-to-point link network. **b** Periodic short-term changes in point-to-point link network



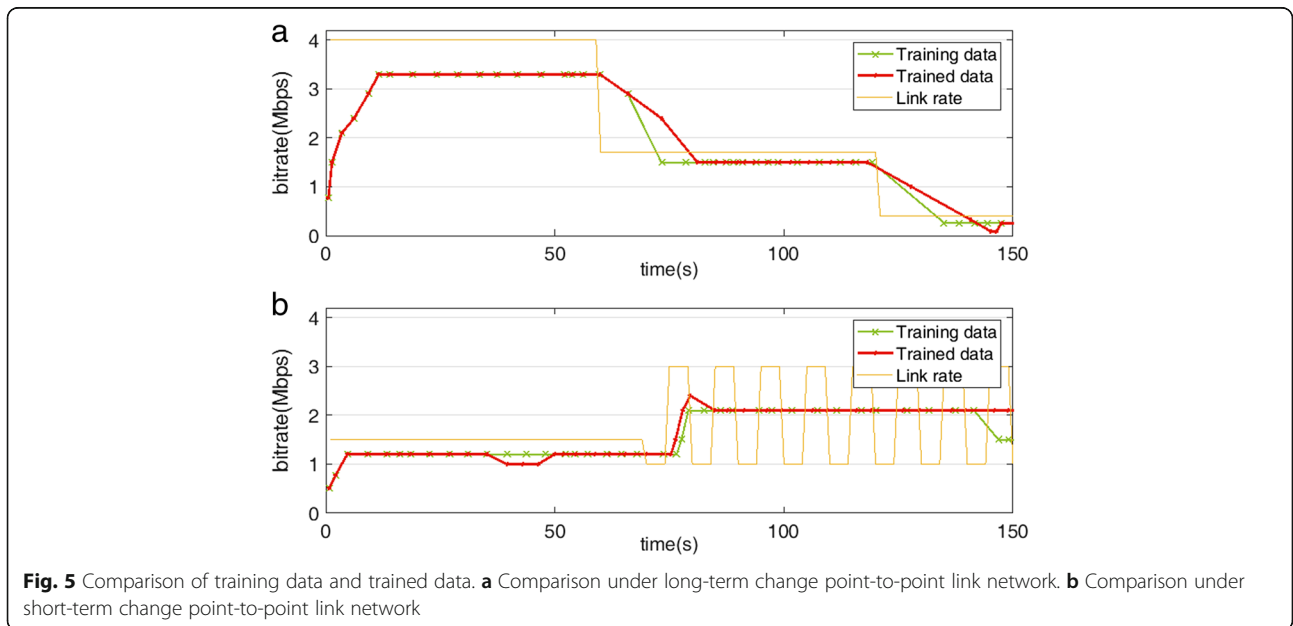
proposed scheme uses average download time as the second input parameter as expressed in Eq. (9):

$$x_2 : T_i = \begin{cases} T_{next} & |T_{next} - T_{i-1}| > D \\ \frac{T_{i-1} + T_{next}}{2} & otherwise \end{cases} \quad (9)$$

$T_i$  is the predicted download time for the  $i$ th segment and is calculated as the average of the last predicted download time  $T_{i-1}$  and  $T_{next}$ . However, if the difference between  $T_{i-1}$  and  $T_{next}$  is greater than a segment playback duration  $D$ , it indicates a non-stationary throughput change, meaning the calculation with  $T_{i-1}$  can cause a significant error. Therefore, in this situation,  $T_{i-1}$  is ignored and only  $T_{next}$  is used in the calculation.

The two inputs are passed to the first layer and are computed using a Gaussian membership function as shown in Table 2 and Fig. 2. In this regard, since the range of the training data may be smaller than the range of the actual input values, the trained membership function may not provide the correct value for all the input values in the form of a normal Gaussian membership function. Therefore, the Gaussian membership function corresponding to  $A_1$ ,  $B_1$ ,  $A_3$ , and  $B_3$  is transformed as follows:

$$\mu_i(x) = \begin{cases} \exp\left(-\frac{(x-c_i)^2}{2\sigma_i^2}\right) & x \geq c \\ 1 & x < c, i = 1, \end{cases} \quad (10)$$





$$\mu_i(x) = \begin{cases} \exp\left(-\frac{(x-c_i)^2}{2\sigma_i^2}\right), & x \leq c \\ 1, & x > c, i = 3, \end{cases} \quad (11)$$

The final output  $y$  is obtained after the calculation in each layer of the proposed structure. The output  $y$  represents the increase/decrease factor of the segment bit rate. Therefore, the next bit rate is the bit rate closest to  $Q_{last} + y$ , where  $Q_{last}$  represents the bit rate of last segment.

When the client requests the first segment, the client cannot calculate  $R$ , because it does not know  $S_{last}$  and  $T_{last}$ . Moreover, a prolonged download time of the first segment leads to a longer start time for the client watching the video, which has a negative effect on the client's QoE [15]. Therefore, the proposed scheme requests the lowest bit rate segment as the first segment, allowing the client to begin watching the video as quickly as possible.

### 3.2 Training data

The proposed scheme requires training data for training. The data that provide high satisfaction to clients in actual streaming services can be used as training data. However, if this data cannot be obtained, arbitrary training data should be created. To create the training data, we have set arbitrary situations using NS-3. First, we have set up two types of point-to-point link network environments, as shown in Fig. 3. Then, a client that knows the actual link capacity of the network requests a segment of the bit rate suitable for the actual link capacity. In this process, the data corresponding to  $x_1$ ,  $x_2$ , and  $y$  of the proposed structure are obtained.

In the simulation, the data are collected from a variety of link networks with different link capacity variations, different change times of the link capacity, and different initial link capacities in order to obtain different types of data corresponding to all the fuzzy rules of the proposed structure. After collecting the training data, the training





process is iterated until the error between the output value of the training data and the output value of the proposed structure converges to the specified value.

#### 4 Results and discussion

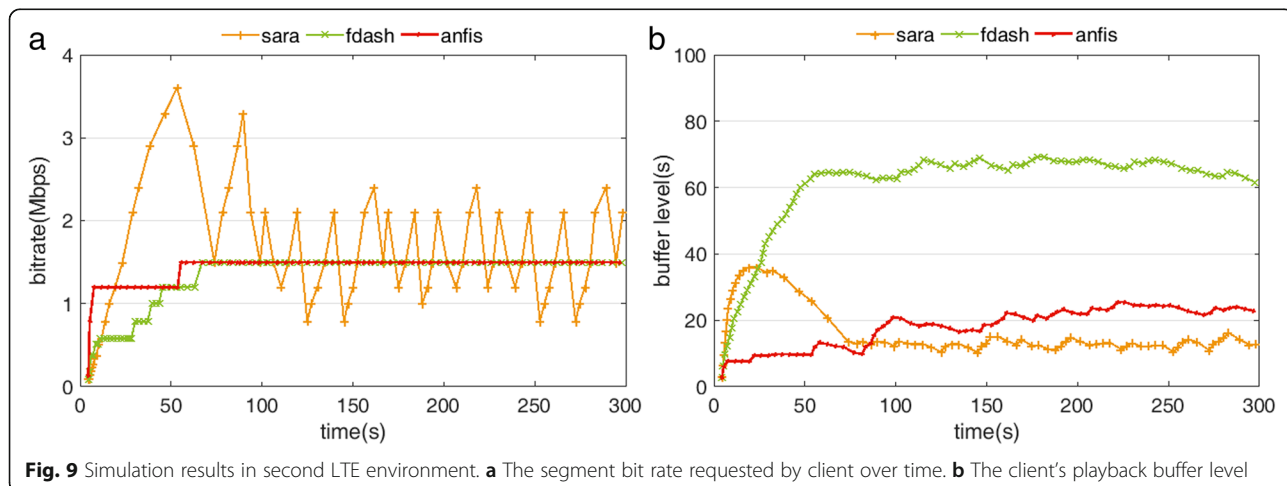
In this section, we evaluate the proposed scheme by analyzing the NS-3 simulation results compared to those of FDASH [11] and segment aware rate adaptation algorithm (SARA) [10]. In the simulations, we used the Big Buck Bunny [22] with a segment playback duration  $D$  of 4 s and available bit rates of 45, 89, 129, 177, 218, 256, 323, 378, 509, 578, 783, 1000, 1200, 1500, 2100, 2400, 2900, 3300, 3600, and 3900 kbps. In addition, the client's maximum buffer size was 100 s. The target buffer level of FDASH was set to 35 s, and in the SARA,  $I$  was set to 8 s,  $B_\alpha$  was set to 30 s, and  $B_\beta$  was set to 30 s.

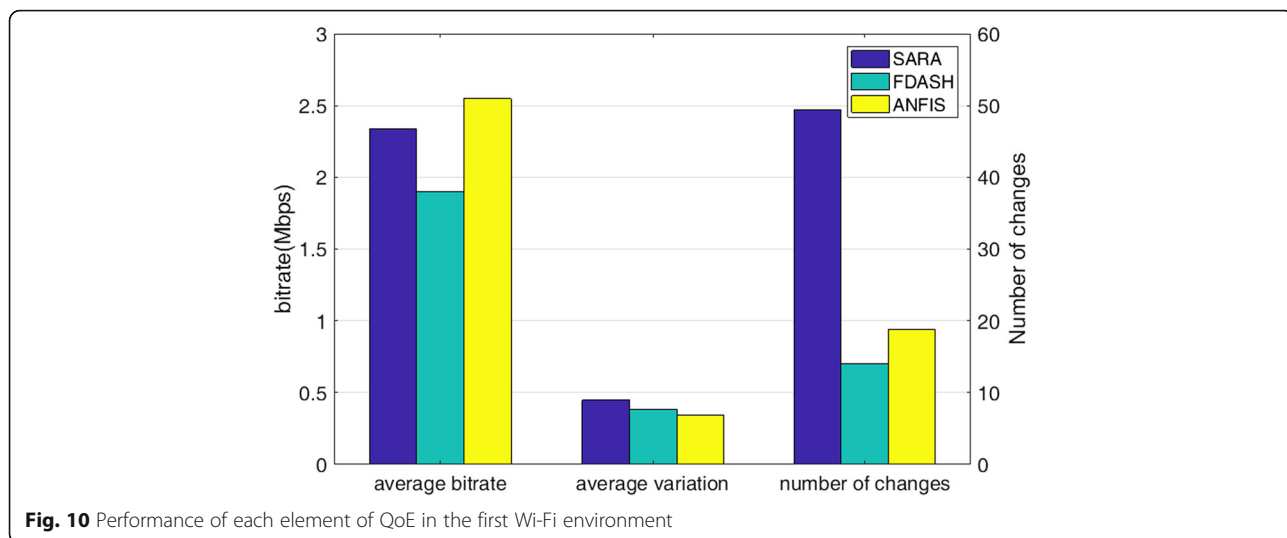
To evaluate the performance of each scheme, the following QoE matrix [12], which is a weighted combination

of video quality, quality variations, rebuffering time, and startup delay, was used:

$$QoE = \sum_{k=1}^K q(R_k) - \lambda \sum_{k=1}^{K-1} |q(R_{k+1}) - q(R_k)| - \mu \sum_{k=1}^K T_{reb} - \mu_s T_s \tag{12}$$

The first term in Eq. (12) represents the total video quality.  $K$  is the number of downloaded segments, and  $R_k$  is the bit rate of the  $k$ th downloaded segment.  $q(\cdot)$  is a function which maps  $R_k$  to the video bit rate perceived by the client. The second term represents the total magnitude of the changes in video quality.  $T_{reb}$  in the third term is the rebuffering time.  $T_s$  in the final term is the startup delay.  $\lambda$ ,  $\mu$ , and  $\mu_s$  are non-negative weighting parameters corresponding to video quality variations, rebuffering time, and startup delay, respectively. The value of each parameter determines how much each term affects QoE.





In this simulation, we assume  $q(R_k) = R_k$ , and use the weights  $\lambda = 1, \mu = \mu_s = 3000$  [12]. The simulation was performed in two Wi-Fi environments and two LTE environments. In each environment, all DASH clients started watching videos using FDASH, SARA, or the proposed scheme as ABR scheme at the same time. Each simulation was performed 10 times for 300 s. Since the number of segments downloaded by each scheme over the period of 300 s was different, we set  $K$  as 75, which was the result of 300 s divided by  $D$ .

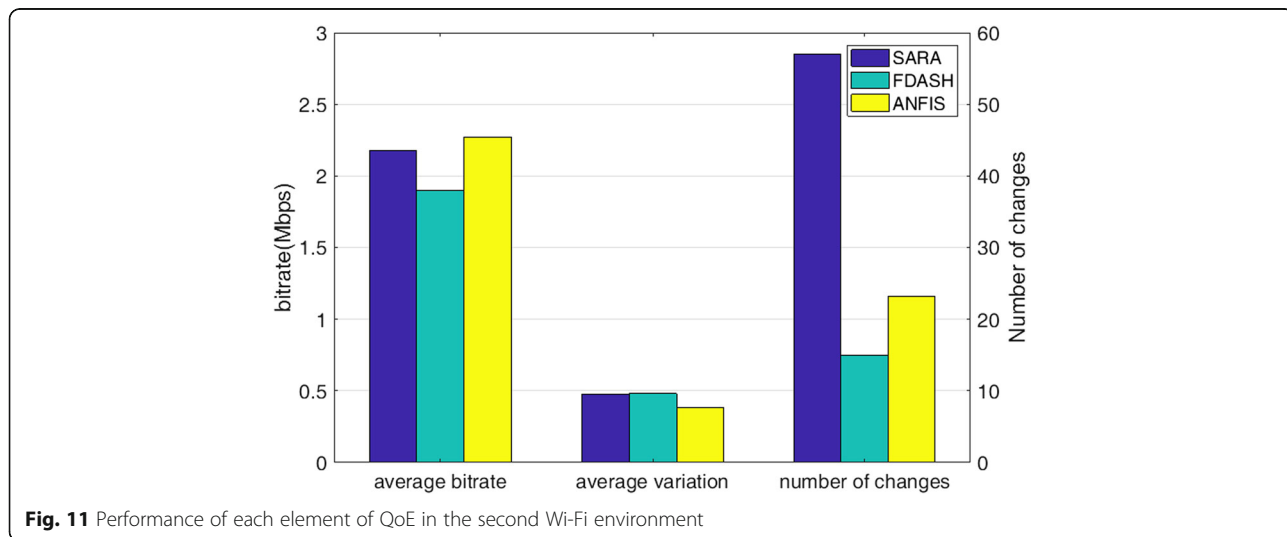
**4.1 Training results**

The training results of the proposed scheme are shown prior to the simulation results, as illustrated in Figs. 4 and 5. To fit the QoE matrix, we created the training data under the following two conditions: (1) selection of the highest bit rate without any expected rebuffering event and (2) reducing the amount of quality variations.

Figure 4 represents the fuzzy membership function based on the trained premise parameters. Figure 5 represents a comparison of the streaming results applying trained ANFIS to a few parts of the training data. In Fig. 5, the orange line indicates the link capacity set in the point-to-point link network, the green line indicates the bit rate training data, and the red line indicates the bit rate determined using the trained ANFIS. Although the red and green lines are not perfectly matched, they exhibit similar shapes and request similar bit rate values.

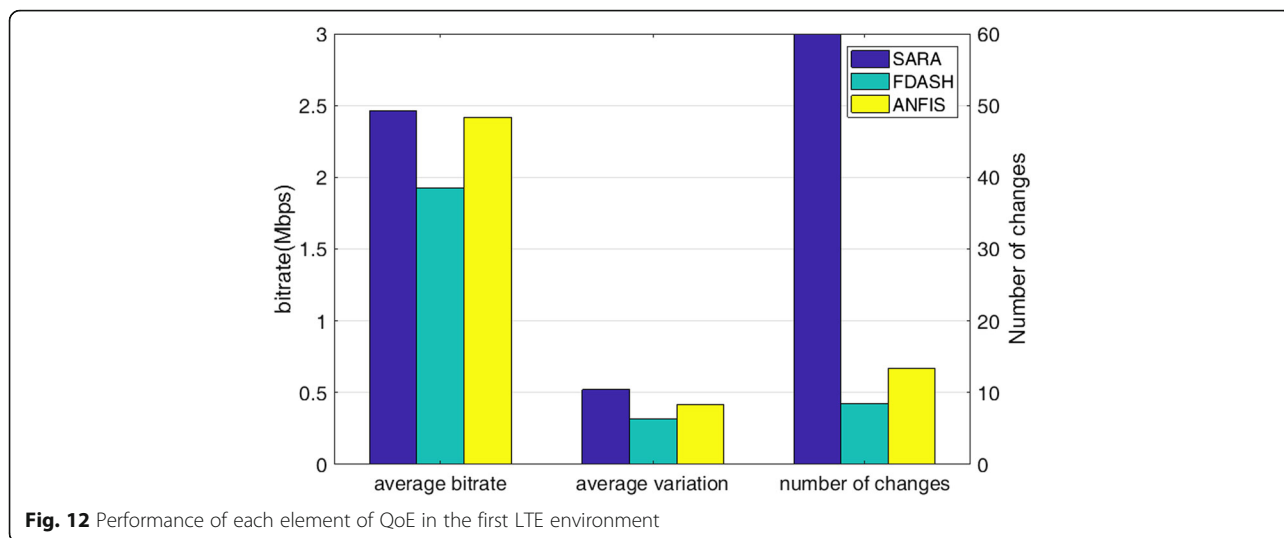
**4.2 Simulation results**

The simulation results in the two Wi-Fi environments are shown in Figs. 6 and 7 in order. The first Wi-Fi environment contains a DASH server, a single DASH client, and five background TCP traffic. The second Wi-Fi environment contains a DASH server and six DASH clients, two for each ABR scheme. In the figures, (a) shows



**Fig. 11** Performance of each element of QoE in the second Wi-Fi environment





the segment bit rate requested by each algorithm over time and (b) shows the client’s playback buffer level.

In those simulation results, FDASH requests many segments at a low bit rate until the target buffer level is reached. After reaching the target buffer level, the bit rate of the next segment gradually increases. In Fig. 6, the FDASH client requests the segment of 1500 kbps at 73 s for the first time, which is the 34th request, while the proposed scheme requests a segment of 1500 kbps at 18 s for the first time, which is the 7th request. In other words, the FDASH client has to wait 136 s ( $34 \times D$ ) to view the video at 1500 kbps. The SARA increases the client buffer level quickly by requesting many low bit rate segments and requests a high bit rate segment when there is a sufficient buffer level. However, because it requests an excessively high bit rate segment, the

download time for the segment is too long and the client buffer is quickly consumed. This causes an abrupt decrease in the segment bit rate. Moreover, since the SARA method increases the bit rate step by step from the low bit rate following an abrupt reduction in the bit rate, the bit rate changes frequently. In addition, although FDASH and SARA show small variations when the bit rate increases, they generally show large variations when the bit rate decreases. On the other hand, the proposed scheme shows a small variation in both the increases and decreases in bit rate, according to the characteristics of the training data.

LTE environments contain a single base station and 50 physical resource blocks (PRBs) (10 MHz bandwidth) used as radio resources. The base station uses a proportional fair resource allocation scheme as the resource

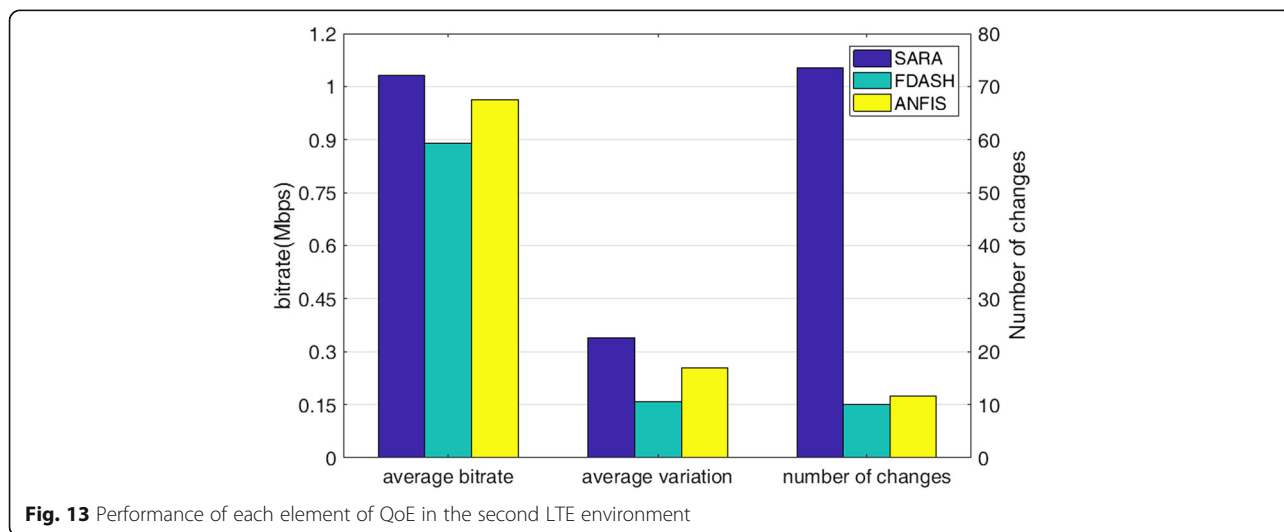


Fig. 13 Performance of each element of QoE in the second LTE environment

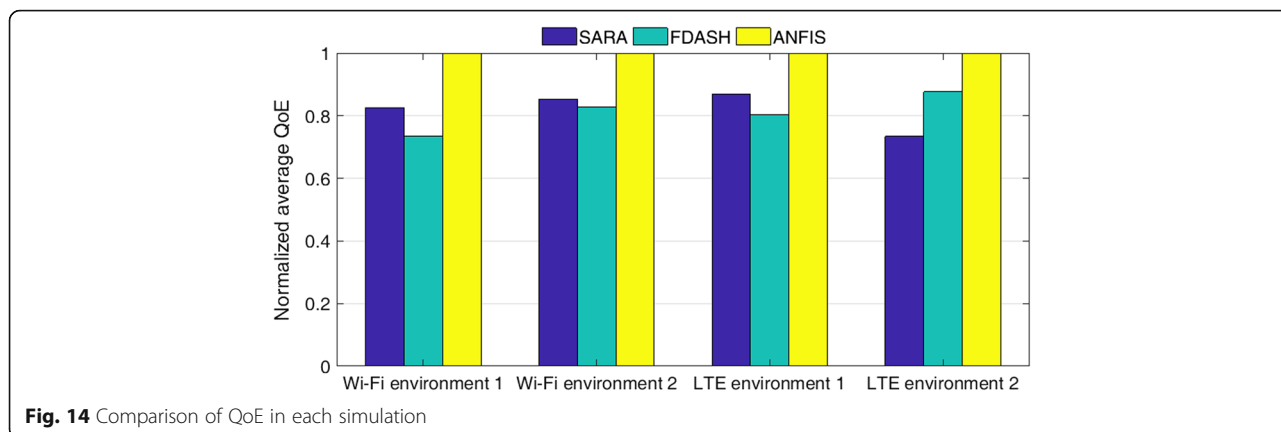


Fig. 14 Comparison of QoE in each simulation

allocation scheme. In addition, the extended typical urban (ETU) fading model and log-distance propagation loss model are used.

The first LTE environment contains ten background TCP traffic and three DASH clients, one for each ABR scheme. All DASH clients are fixed at the same location and all TCP traffic move randomly at a speed of 3 m/s. The second LTE environment contains a single DASH client and 20 background TCP traffic. Both a DASH client and all TCP traffic randomly move at a speed of 3 m/s. The simulation results under each LTE environment are shown in Figs. 8 and 9 in order.

In the figures, FDASH shows a slow increase in bit rate and stably maintains the appropriate bit rate after reaching it. On the other hand, SARA does not maintain an appropriate bit rate even though it reaches a high bit rate quickly. The proposed scheme quickly reaches a high bit rate and maintains this bit rate.

To sum up the results, FDASH maintains the appropriate bit rate and shows a low number of bit rate changes. However, it requests a low bit rate segment until the client buffer reaches the target buffer level, so clients have to wait a long time to watch high-quality video. SARA reaches a high bit rate faster than other methods. However, it consumes client buffers quickly by requiring segments with an excessively high bit rate. The proposed scheme reaches a high bit rate faster than FDASH, and it maintains the bit rate longer than SARA after reaching the bit rate. In addition, it usually shows a small quality variation.

### 4.3 QoE evaluation

We compared the performances of each element in Eq. (12). As all the schemes in our simulation request the

lowest bit rate segment as the first segment, we did not compare the startup delays. The average results for the elements under each environment are shown in Figs. 10, 11, 12, and 13, in order.

In Figs. 10 and 11, which are the results in the two Wi-Fi environments, FDASH shows the lowest average bit rate, although its number of bit rate changes is smallest. SARA shows a higher average bit rate than that of FDASH. However, it shows the largest number of bit rate changes, and its variation in bit rate is also the largest. The proposed scheme shows a similar number of changes as FDASH. In addition, the proposed scheme shows the smallest variation and the highest average bit rate of all the schemes evaluated. Thus, it is clearly shown that the proposed method appropriately meets each element of a given QoE matrix.

Figures 12 and 13 represent the results in LTE environments. In these figures, SARA shows the highest average bit rate. However, it also shows the highest number of bit rate changes. FDASH shows the lowest average bit rate, although it shows the lowest number of changes and the lowest average variation because of the reduced occurrence of bit rate reduction. In other words, FDASH and SARA show some advantages, but they also show clear drawbacks. In contrast to those two schemes, the proposed method shows an average bit rate similar to SARA and a number of changes similar to FDASH. Therefore, in LTE environments, the proposed method appropriately meets each element, as in the simulation of the Wi-Fi environment.

Finally, we calculated the QoE for all the simulations and normalized the QoE for the proposed scheme as shown in Fig. 14. Compared with SARA and FDASH in each environment, the proposed scheme provides approximately 22% and 24% better QoE to clients, respectively.

Our simulation results show that the proposed scheme leads to better performance than the existing ABR scheme by learning through the training data appropriate to the

Table 3 Operation time of ABR schemes

ABR scheme	SARA	FDASH	Proposed scheme
Operation time	92.1420 $\mu$ s	95.3444 $\mu$ s	110.3276 $\mu$ s

QoE matrix. In addition, the application of the proposed scheme to actual service is expected to improve the performance of the scheme by directly collecting and using the real video playback data that the client has satisfactorily watched.

#### 4.4 Computational complexity

In order to compare the computational complexity of each ABR scheme, we measured the operating time of each ABR scheme. The operating time is the time to calculate the next bit rate to request a segment. Table 3 represents measured operating time. In Table 3, SARA takes the least amount of time and FDASH takes time similar to SARA. The proposed scheme operates similarly with FDASH, but it takes more time than FDASH since the input membership functions of the proposed scheme perform more complex calculations than the functions of FDASH. However, the proposed scheme can provide better QoE to DASH clients by using the more complex membership functions.

#### 5 Conclusion

In this paper, we proposed a new ABR scheme using ANFIS, which is one of the neuro-fuzzy system structures. While the fuzzy membership functions of the FDASH scheme lack the basis for the function setting, the proposed scheme learns the fuzzy membership functions via the training data and provides the basis. In our simulation, we trained the fuzzy membership functions of the proposed scheme with training data optimized for the given QoE matrix and demonstrated that the proposed scheme using the trained fuzzy membership functions provided better QoE than the existing schemes. In addition, it is expected that the proposed scheme will perform better by using the data that has already satisfied the client when implemented in an actual video streaming service.

#### Abbreviations

ABR: Adaptive bit rate; ANFIS: Adaptive network-based fuzzy inference system; DASH: Dynamic adaptive streaming over HTTP; ETU: Extended typical urban; FDASH: Fuzzy-based MPEG/DASH adaptation algorithm; FLC: Fuzzy logic controller; HAS: HTTP-based adaptive streaming; HTTP: Hypertext transfer protocol; LTE: Long-term evolution; PRB: Physical resource block; QoE: Quality of experience; SARA: Segment aware rate adaptation algorithm

#### Acknowledgements

This research was supported by the KIAT (Korea Institute for Advancement of Technology) grant funded by the Korea Government (MOTIE: Ministry of Trade Industry and Energy). (No. N0001884, HRD program for Embedded Software).

#### Funding

Not applicable.

#### Availability of data and materials

Please contact the corresponding author at kys221@konkuk.ac.kr.

#### Authors' contributions

YSS is responsible for writing the paper, conducting the simulations, and inventing the proposed scheme. HJK is responsible for writing the paper and conducting the simulations. JTK is responsible for inventing the proposed scheme. All authors read and approved the final manuscript.

#### Competing interests

The authors declare that they have no competing interests.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 25 December 2017 Accepted: 23 October 2018

Published online: 14 November 2018

#### References

1. SANDVINE. 2016 Global Internet Phenomena Report. North America and Latin America. 2016
2. D.K. Krishnappa, D. Bhat, M. Zink, in *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*. DASHing YouTube: an analysis of using DASH in YouTube video service (IEEE, 2013), pp. 407–415
3. I Mpeg. Information Technology-Dynamic Adaptive Streaming over Http (Dash)-Part 1: Media Presentation Description and Segment Formats. ISO/IEC MPEG, Tech. Rep (2012)
4. T. Stockhammer, in *Proceedings of The Second Annual ACM Conference on Multimedia Systems*. Dynamic Adaptive Streaming over HTTP--: Standards and Design Principles (ACM, 2011), pp. 133–144
5. K. Xiao, S. Mao, J.K. Tugnait, in *Global Communications Conference (GLOBECOM), 2016 IEEE*. QoE-driven resource allocation for DASH over OFDMA networks (IEEE, 2016), pp. 1–6
6. H.J. Kim, Y.S. Son, J.T. Kim, in *Consumer Electronics (ICCE), 2018 IEEE International Conference on*. KKT-conditions based resource allocation algorithm for DASH streaming service over LTE (IEEE, 2018), pp. 1–3
7. L. He, F. Li, Content and buffer status aware packet scheduling and resource management framework for video streaming over LTE system. *EURASIP J. Image Video Process.* 1(2017), 73 (2017)
8. J. Jiang, V. Sekar, H. Zhang, in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive (ACM, 2012), pp. 97–108
9. T.-Y. Huang et al., A buffer-based approach to rate adaptation: evidence from a large video streaming service. *ACM SIGCOMM Comput. Commun. Rev.* 44(4), 187–198 (2015)
10. P. Juluri, V. Tamarapalli, D. Medhi, in *Communication Workshop (ICCW), 2015 IEEE International Conference on*. SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP (IEEE, 2015), pp. 1765–1770
11. D.J. Vergados et al., FDASH: A fuzzy-based MPEG/DASH adaptation algorithm. *IEEE Syst. J.* 10(2), 859–868 (2016)
12. X. Yin et al., A control-theoretic approach for dynamic adaptive video streaming over http. *ACM SIGCOMM Comput. Commun. Rev.* 45(4), 325–338 (2015)
13. M. Seufert et al., A survey on quality of experience of HTTP adaptive streaming. *IEEE Commun. Surv. Tutorials* 17(1), 469–492 (2015)
14. F. Dobrian et al., Understanding the impact of video quality on user engagement. *ACM SIGCOMM Comput. Commun. Rev.* 41(4), 362–373 (2011) ACM
15. L. Yitong et al., in *Communications Workshops (ICC), 2013 IEEE International Conference on*. A study on quality of experience for adaptive streaming service (IEEE, 2013), pp. 682–686
16. D.o.Z. Rodriguez et al., The impact of video-quality-level switching on user quality of experience in dynamic adaptive streaming over HTTP. *EURASIP J. Wirel. Commun. Netw.* 2014(1), 216 (2014)
17. J. Kua, G. Armitage, P. Branch, A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP. *IEEE Commun. Surv. Tutorials* 19(3), 1842–1866 (2017)
18. J.-S.R. Jang, ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man. Cybern.* 23(3), 665–685 (1993)

19. C.-T. Lin, C.S. George Lee, Neural-network-based fuzzy logic control and decision system. *IEEE Trans. Comput.* **40**(12), 1320–1336 (1991)
20. T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man. Cybern.* **1**, 116–132 (1985)
21. Q. He, C. Dovrolis, M. Ammar, On the predictability of large transfer TCP throughput. *ACM SIGCOMM Comput. Commun. Rev.* **35**(4), 145–156 (2005) ACM
22. The Big Buck Bunny Movie. <http://www.bigbuckbunny.org>. Accessed 2 Aug 2017.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---