FDASH: A Fuzzy-Based MPEG/DASH Adaptation Algorithm

Dimitrios J. Vergados, *Member, IEEE*, Angelos Michalas, *Member, IEEE*, Aggeliki Sgora, Dimitrios D. Vergados, *Senior Member, IEEE*, and Periklis Chatzimisios, *Senior Member, IEEE*

***Abstract*—The evolution of broadband connection technologies, along with the explosive growth of mobile handheld devices, has made video content one of the major Internet traffic types. HTTP adaptive streaming (HAS) is one factor assisting the growing popularity of Internet video. One of its standards, the MPEG dynamic adaptive streaming over HTTP (DASH), allows users to access video streams of multiple resolutions available at a central repository. Various adaptation algorithms using the MPEG DASH protocol have been proposed, aiming to adapt the media bit rate delivered at the client to the current network conditions. In this paper, we present the FDASH rate adaptation scheme that employs fuzzy logic to control the buffering time and the video resolution delivered to the client in order to distribute video segments of the best quality, deliver undisrupted video playback, and avoid frequent changes of video resolution. The comparison of FDASH against other rate adaptation schemes shows that the former successfully provides the client with the optimal video rates, avoiding buffer underflows and unnecessary changes of video resolution in case of fluctuations of the available connection throughput.**

***Index Terms*—Adaptation algorithm, HTTP adaptive streaming (HAS), MPEG dynamic adaptive streaming over HTTP (MPEG DASH), transmission control protocol (TCP).**

I. Introduction

THE tremendous growth of wireless technologies, such as long-term evolution (LTE)/LTE Advanced [1] and Worldwide Interoperability for Microwave Access [2], the proliferation of mobile computing environments [3] (by utilizing mobile phones, smartphones, tablets, etc.), and cloud computing [4] gave the opportunity to the users to experience new multimedia services, such as video streaming, personal video broadcasting, and Internet Protocol television (IPTV) [5]–[7].

Video streaming already accounts for a large portion of all Internet traffic and is currently on a steady rise. This tendency has led to the development of a range of protocols, such as Silverlight smooth streaming, HTTP live streaming, and HTTP dynamic streaming, and standards, such as 3GPP-DASH and MPEG dynamic adaptive streaming over HTTP (MPEG DASH), that allow adaptive video streaming over HTTP.

Independently of the implementation, the basic concepts of HTTP adaptive streaming (HAS) are the following. The video content is encoded into temporal segments at different quality rates and stored on the HTTP server(s). The clients can download the segments through standard HTTP requests. In particular, the clients automatically choose the initial quality rate without requiring negotiation with the server and dynamically switch between different bit rate representations of the video content to achieve better quality.

Extensive research has been carried out concerning the optimization of the HAS delivery. At the server side, optimization is focused on the encoding schemes [8]–[10]. Other solutions focus on quality-of-experience (QoE) issues, dealing with QoE assessment, monitoring, and management for different network technologies and applications [11]–[13]. At the client side, optimization may be achieved by applying the appropriate adaptation algorithm.

In this paper, we focus on MPEG DASH (ISO/IEC 23009-1), a standard issued by MPEG in 2012 for HAS. MPEG DASH allows users to access video streams of multiple resolutions available at a central repository. However, the standard does not define how the user could adapt to time-varying bandwidth in order to achieve better quality.

In our previous work [14], [15], we proposed a novel MPEG DASH rate adaptation scheme, namely, FDASH, aiming to efficiently adjust the video rate delivered to each user in accordance with the current network conditions. This paper extends our previous work by comparing FDASH with other adaptation algorithms in the context of various network conditions with cross traffic and interference. Several modifications have been also introduced to our previous version of FDASH in order to improve its functionality. More specifically, a fast start playing phase is included so that the video starts playing at the client without having to wait until its buffer level reaches the target buffering time. Moreover, the configuration parameters of the fuzzy logic controller (FLC) have been adjusted after several tests to increase its performance.

This paper is structured as follows. In Section II, we present a review of the relevant literature. In Section III, we introduce our FLC, whereas in Section IV, the FDASH algorithm is presented. Section V provides simulation results indicating the efficiency of the proposed model. Finally, we conclude this paper and discuss ongoing and future research directions in Section VI.

Manuscript received October 30, 2014; revised March 12, 2015; accepted April 26, 2015. Date of publication December 29, 2015; date of current version May 30, 2016.

D. J. Vergados is with the Department of Computer Engineering and Informatics, University of Patras, 26504 Patras, Greece (e-mail: vergados@ceid.upatras.gr).

A. Michalas is with the Department of Informatics Engineering, Technological Educational Institute of Western Macedonia, 52100 Kastoria, Greece (e-mail: amichalas@kastoria.teiwm.gr).

A. Sgora and D. D. Vergados are with the Department of Informatics, University of Piraeus, 18534 Piraeus, Greece (e-mail: asgora@unipi.gr; vergados@unipi.gr).

P. Chatzimisios is with the Department of Informatics, Alexander Technological Educational Institute of Thessaloniki, 57400 Thessaloniki, Greece (e-mail: peris@it.teithe.gr).

Digital Object Identifier 10.1109/JSYST.2015.2478879

II. Related Work

Adaptive streaming is considered as a vital solution to support multimedia services over the Internet. Several adaptation schemes have been proposed to improve the video transmission and end users satisfaction.

The performance of rate adaptation streaming systems against nonadaptive ones is studied in [16]. The authors performed QoE subjective evaluation using as metrics the peak signal-to-noise ratio, the structural similarity, and the video quality metric (VQM). Simulation results showed that, in the case of video rate adaptation and particularly for the VQM rates, a considerable correlation of subjective results with subjective metrics obtained from 75 participants is achieved. Thus, it is validated that rate adaptation algorithms may improve system performance.

In [17], the authors considered the MPEG-2/MPEG-4 quantization scale factor in MPEG-2 transport stream to evaluate the video quality perceived by IPTV users. Network measurements are performed in terms of packet delay, loss, and jitter to evaluate the optimum quantization level according to network conditions. The work in [17] indicates that the scale factor evaluated at the end user can be dynamically adapted to reduce bandwidth consumption and improve the video quality.

In [18], the authors described an adaptation and scheduling algorithm for unicast and multicast video transmissions, using scalable video coding (SVC) over LTE networks. In the unicast scheme, the eNode uses channel quality indicator feedbacks received from user equipment to perform dynamic scheduling of time–frequency resources. Accordingly, cross-layer signaling between Medium Access Control and Real Time Protocol is performed to achieve video rate adaptation at the server. The multicast scheme uses the multicast-broadcast single-frequency channel of LTE for transmission. It performs adaptive modulation and coding and frequency scheduling to allocate multicast users to three channel quality regions based on users' signal-to-interference-plus-noise ratio. Simulation results showed that both schemes improve the video quality and optimize the utilization of bandwidth.

HTTP streaming is one more factor contributing to the growing popularity of Internet video. To this end, several research studies using the recently proposed DASH standard over HTTP have been carried out. These studies actually perform video rate adaptation at the client side in order to adapt the media bit rate based on the current network conditions and clients buffer occupancy.

The adaptation algorithm for adaptive streaming over HTTP (AAASH) presented in [19] adapts the video rate requested by the client using a complex adaptation mechanism with multiple conditions and configuration parameters. The algorithm selects the lowest representation for the first segment to be downloaded to reduce the time until the playback starts. Subsequently, it includes a fast start phase at the beginning of the playback to increase video quality and proceeds to the next phase in case one of the following conditions is met: 1) the highest video resolution has been reached; 2) the buffer level is not increased monotonically; 3) the bit rate of the segment resolution is close to the estimated throughput.

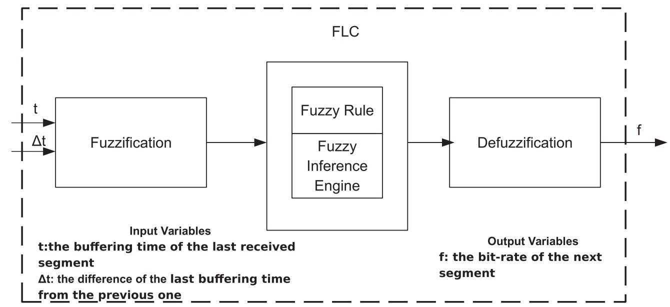


Fig. 1. Structure of the FLC.

In [20], a rate adaption algorithm is proposed using the DASH protocol for serial and parallel segment fetching time methods in content distribution networks (CDNs). The proposed scheme introduces a rate adaptation metric to detect network availability, which is defined as the ratio of the optimum over the measured segment fetch time. The optimum segment fetch time is evaluated by taking into account the media segment duration and the buffering time at the client. Accordingly, a stepwise switch-up strategy and a multistep switch-down strategy are adopted based on the suggested rate adaptation metric. An idling mechanism is also used to avoid buffer overflows, whereas priority segment fetch times are assigned to new clients to improve fairness.

The agile smooth video adaptation algorithm (SVAA) for DASH systems proposed in [21] estimates the video rate of the next downloadable video segment using the current playback buffering time of the client. The algorithm attempts to adapt the downloaded video rate to the available network throughput while it enforces a buffer cap to avoid buffer overflows. Moreover, it reduces the estimated video rate by a small rate margin to avoid unnecessary video rate adjustments.

The authors in [22] proposed a rate adaptation for adaptive HTTP streaming (RAAHS) algorithm, which evaluates the network bandwidth by comparing the segment fetch time and the playback time at the client. A stepwise switch-up method and an aggressive switch-down method are used to adjust download bit rate to the available throughput. Furthermore, an idle time is determined to control requests for video segments and confine the buffering time to a maximum limit.

The Adobe's Open Source Media Framework (OSMF) [23] for HTTP adaptive streaming estimates the available bandwidth by considering the download time of the last two video segments. The bit rate of the next video segment to be downloaded will be equal to the maximum among the applicable encoding rates for that estimated bandwidth.

III. The Fuzzy Logic Controller

Fuzzy Logic Controllers (FLCs) are one of the most important applications of fuzzy logic. An FLC consists of the following components, which are depicted in Fig. 1.

- Fuzzification: During this process, each element of input data is converted to degrees of membership by a lookup in one or several membership functions [24].
- Fuzzy rule (or knowledge) base: A fuzzy rule is a simple if-then rule with a condition and a conclusion. It

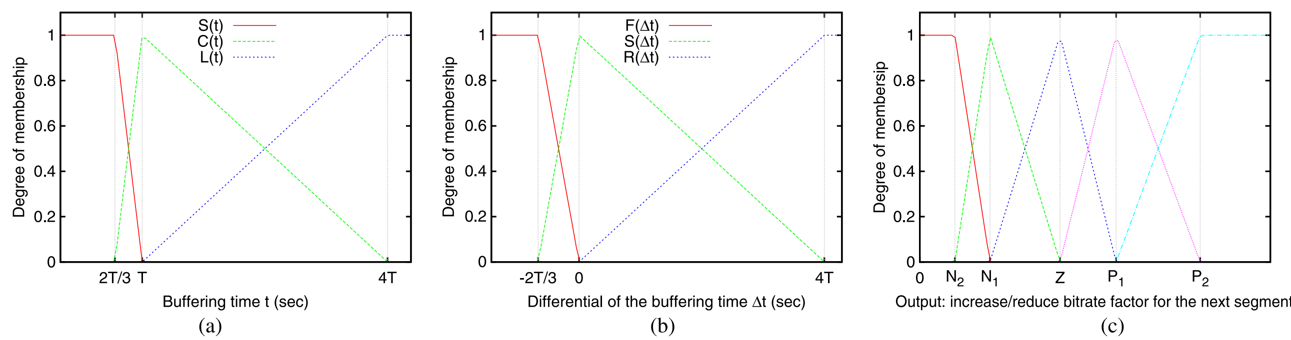


Fig. 2. Input and output membership functions.

associates the fuzzy output to the fuzzy inputs and is constructed to control the fuzzy output. In case that a rule has multiple parts, fuzzy operators may be used to combine more than one input. During this step, the rule matrix is also built to describe fuzzy sets and fuzzy operators in the form of conditional statements.

- Fuzzy inference engine: It performs the fuzzy inference process, by computing the activation degree and the output of each rule.
- Defuzzification: During this step process, the fuzzy set is transformed to a crisp set. The five defuzzification methods that can be used are the following: the centroid, the bisector, the middle of maximum, the largest of maximum, and the smallest of maximum [24].

In our controller, the input variables are the buffering time denoting the time t_i that the last received segment i waits at the client until it starts playing and the difference $\Delta t_i = t_i - t_{i-1}$ of the last buffering time from the previous one.

Three linguistic variables [short (S), close (C), and long (L)] are adopted for the buffering time in order to describe the distance of the current buffering time from a target buffering time T that FDASH has set in order to avoid buffer under runs and to retain the difference between the current and previous resolutions close to zero in order to reduce consecutive changes of video resolution subject to continuous variations of the network throughput. Moreover, since for the differential of the buffering time input Δt_i we need to describe the behavior of the rate between subsequent buffering times, the following linguistic variables are considered: falling (F), steady (S), and rising (R).

In a similar manner, the output of the FLC f represents an increase/decrease factor of the resolution of the next segment. Thus, the linguistic variables of the output are described as reduce (R), small reduce (SR), no change (NC), small increase (SI), and increase (I). Fig. 2 depicts the membership functions of the input and output variables.

The fuzzy if-then rules for our controller are the following:

Rule 1 (r1): if *(short)* and *(falling)* then R
Rule 2 (r2): if *(close)* and *(falling)* then SR
Rule 3 (r3): if *(long)* and *(falling)* then NC
Rule 4 (r4): if *(short)* and *(steady)* then SR
Rule 5 (r5): if *(close)* and *(steady)* then NC
Rule 6 (r6): if *(long)* and *(steady)* then SI
Rule 7 (r7): if *(short)* and *(rising)* then NC

TABLE I
SIMULATION PARAMETERS OF THE ALGORITHMS

Algorithm	Parameters	Value	Definition
FDASH	T	35 sec	Target buffering time
	d	60 sec	Time period estimating the connection throughput
	(N_2, N_1, Z, P_1, P_2)	(0.25, 0.5, 1, 1.5 ,2)	Factors of the output membership functions
AAASH	B_{min}	10 sec	Minimum buffer level
	B_{tar}	[20, 50] sec	Target buffer level interval
	D_β	1 sec	Discretization parameter for the buffer level
	δ_t	10 sec	Average throughput estimation period
	$(a1, .., a5)$	(0.75, 0.33, 0.5, 0.79, 0.9)	Safety margins
SFTM	bmt_{min}	20 sec	Idle period between consecutive requests
	$TBMT$	20 sec	Target buffered media time
	ρ	0.75	Remaining segment fetch time priority factor
SVAA	q_{ref}	20 sec	Target buffer size
	m	dynamic-m function	Responsiveness parameter
	q_{max}	20 sec	The buffer cap
	ρ_v	0, 05	Video rate reduction margin
	W	10 segments	Window of the last downloaded segments
RAAHS	γ_d	0.67	Switch down threshold
	t_{min}	9 sec	Minimum buffered media time

Rule 8 (r8): if *(close)* and *(rising)* then SI
Rule 9 (r9): if *(long)* and *(rising)* then I

The value of each rule is calculated as the minimum value among the two input functions that comprise it.

Finally, for the defuzzification step, the centroid method is used. Thus, the output f is calculated as

$$f = \frac{N_2 \times R + N_1 \times \text{SR} + Z \times \text{NC} + P_1 \times \text{SI} + P_2 \times I}{\text{SR} + R + \text{NC} + \text{SI} + I} \tag{1}$$

where

$$I = \sqrt{r_9^2} \tag{2}$$

$$\text{SI} = \sqrt{r_6^2 + r_8^2} \tag{3}$$

$$\text{NC} = \sqrt{r_3^2 + r_5^2 + r_7^2} \tag{4}$$

$$\text{SR} = \sqrt{r_2^2 + r_4^2} \tag{5}$$

$$R = \sqrt{r_1^2}. \tag{6}$$

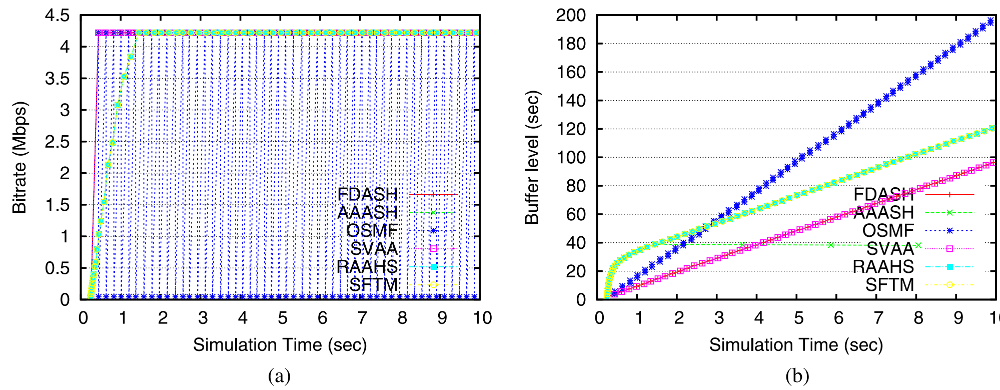


Fig. 3. One client, high-capacity links, with no background traffic. (a) Bit rate. (b) Buffer level.

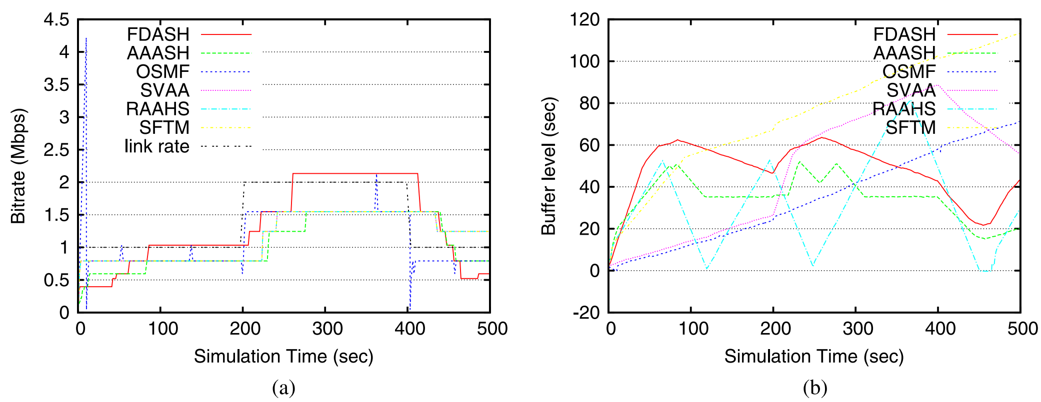


Fig. 4. One client, link speed changes, with no background traffic. (a) Bit rate. (b) Buffer level.

TABLE II
ONE CLIENT, LINK SPEED CHANGES, WITH NO BACKGROUND TRAFFIC

Algorithm	Interruptions	Avg. video rate (Kbps)	Resolution changes
FDASH	0	1206.350	15
AASH	0	967.282	16
OSMF	65	1153.380	21
SVAA	0	1107.930	5
RAAHS	286	735.868	21
SFTM	0	1062.790	16

IV. PROPOSED FDASH SCHEME

In our model, we assume that a video stream consisted of n segments of duration τ is available at the server. Each segment is encoded in multiple resolutions of quality. The throughput of the segment i is estimated at the client as

$$r_i = (b_i \times \tau)/(d_i - r_i) \tag{7}$$

where b_i, d_i, and r_i denote the bit rate of segment i, the time when the segment i has been started downloading, and the time the whole segment has been received at the client, respectively.

Each client requests the next video segment that its resolution should be determined. This is achieved through the FLC. As aforementioned, the output of the controller (f) is a factor that refers to the bit rate (and ultimately the resolution) of the next segment, i.e., b_{i+1}, in relation to the estimated channel throughput over the last period. In particular

$$b_{i+1} = f \times r_d \tag{8}$$

where r_d denotes the available connection throughput, which is estimated as the average segments throughput of the last k segments downloaded during a specified period of time d. Consequently, the available connection throughput is given by

$$r_d = 1/k \times \sum_{i=1}^{k} r_i. \tag{9}$$

It should be noted that b_{i+1} is quantized to the highest available resolution b_n that is lower than b_{i+1}.

Furthermore, in order to avoid unnecessary bit rate fluctuations, the following policies have been applied.

- If $b_n > b_i$ and by selecting the new bit rate b_{i+1}, the buffer level is estimated to be less than T for the next 60 s, then the bit rate remains unchanged.
- If $b_n < b_i$ but the old bit rate is estimated to produce a buffer level for the next 60 s that is larger than T, then the bit rate remains unchanged.
- In all other cases, the bit rate is set to b_n.

V. SIMULATION RESULTS

Our experiments were implemented by utilizing the simulation software ns-3 (http://www.nsnam.org). The efficiency of FDASH was evaluated against the well-known rate adaptation algorithms over HTTP, namely, AAASH [19], OSMF [23], SVAA [21], RAAHS [22], and segment fetching time method (SFTM) [20]. The simulation setup consisted of several nodes,

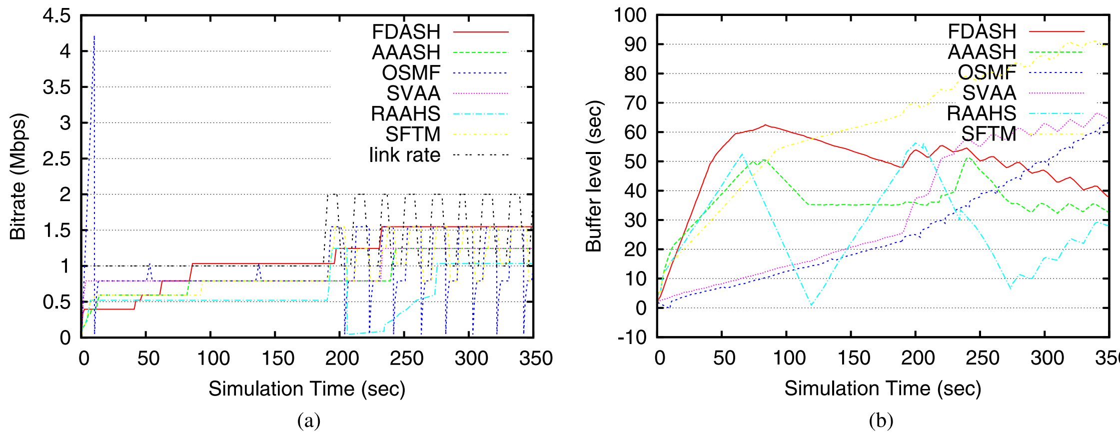


Fig. 5. One client, fast switching link speed, with no background traffic. (a) Bit rate. (b) Buffer level.

one acting as a DASH MPEG server, and the others acting as DASH MPEG clients. Each client node tries to download and play a video from the server using each one of the rate adaptation algorithms.

In particular, the simulations evaluated the performance of the rate adaptation algorithms over a link of variable throughput and in WiFi environments, with cross traffic and channel interference in terms of the following: 1) the attainable download bit rate; 2) the video resolution changes; 3) the playback buffering times; and 4) the buffer underflow incidents. The code of our experiments is available in https://github.com/djvergad/dash.

The parameters used in each of the adaptation algorithm, as well as their definitions and values used in the simulation experiments, are summarized in Table I.

The available video segments at the server have been produced using the trace files of several dash video streams given in http://www-itec.uni-klu.ac.at/ftp/data-sets/mmsys12/BigBuckBunny/. The duration of each video segment τ was equal to 2 s. The video segment resolutions used in the experiments were 45 000, 89 000, 131 000, 178 000, 221 000, 263 000, 334 000, 396 000, 522 000, 595 000, 791 000, 1 033 000, 1 245 000, 1 547 000, 2 134 000, 2 484 000, 3 079 000, 3 527 000, 3 840 000, and 4 220 000 b/s.

A. Single Client Over a High-Capacity Link With No Background Traffic

Fig. 3(a) depicts the download bit rate achieved by a single client over an unlimited bandwidth link, whereas Fig. 3(b) presents the clients' buffering time. It can be seen that both FDASH and SVAA adjust the bit rate in less than 0.3 s to the highest resolution, whereas AAASH achieves that after 1.4 s. In addition to this, OSMF performs oscillations between the highest and lowest available resolutions due to the fact that the algorithm performs rate adaptation by considering only the estimation of the available throughput parameter and disregards the available buffering time at the client. Furthermore, Fig. 3(b) shows that all algorithms increase the playback buffering time above the target limits after a short startup period of 2 s with no video stalls throughout the simulation.

TABLE III
ONE CLIENT, FAST SWITCHING LINK SPEED, NO BACKGROUND TRAFFIC

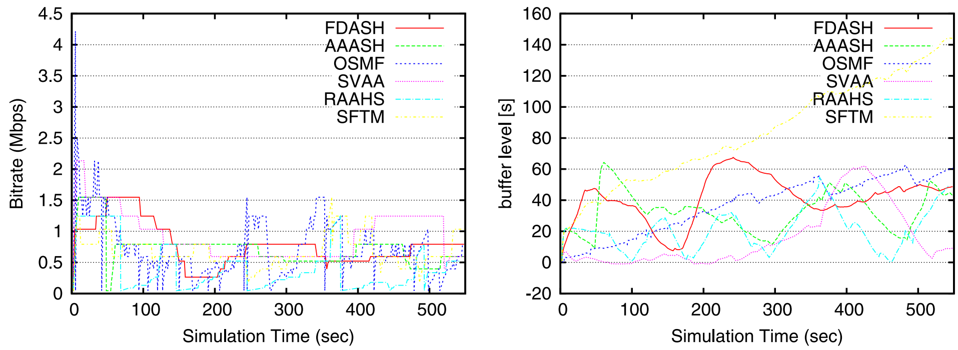
Algorithm	Interruptions	Avg. video rate (Kbps)	Resolution changes
FDASH	0	1069.990	10
AASH	0	907.584	12
OSMF	65	971.918	86
SVAA	0	981.259	3
RAAHS	0	787.654	25
SFTM	0	904.044	66

B. Single Client Over a Link With Constant Throughput Changes and No Background Traffic

Fig. 4 illustrates the performance of algorithms subject to constant differentiations of the line throughput. In particular, the line throughput changes from 1 Mb/s after the first 200 s to 2 Mb/s for 200 s and back to 1 Mb/s for the remaining 100 s. The number of video interruptions, the average download rate, and the number of resolution changes achieved per algorithm are outlined in Table II. As shown in Fig. 4(a), the download bit rate presented follows the link capacity changes within quite short intervals of time for all mechanisms. Furthermore, the video resolution provided by FDASH is higher than that provided by the other adaptation algorithms in most of the simulation time. In particular, FDASH achieves an average download bit rate of 1206.350 kb/s, whereas OSMF, achieving the next higher download bit rate, has an average of 1153.380 kb/s. However, OSMF performs oscillations among high and low video resolutions and exhibits 65 video stalls. At the same time, most mechanisms manage to keep the playback buffering time to acceptable levels, as illustrated Fig. 4(b), with an exception of RAAHS performing 286 video stalls during simulation. Moreover, the video resolution changes of DASH are more frequent than that of SVAA; nevertheless, the algorithm decides to change the video resolution in a stepwise fashion instead of moving abruptly from a low to a high video resolution.

C. Single Client Over a Link With Periodic Short-Term Throughput Changes and No Background Traffic

Fig. 5 presents the algorithms' behavior when the available throughput follows short-term periodic picks and falls lasting approximately 10 s each. Table III summarizes the number of



(a) (b)

Fig. 6. One client, WiFi link, with background TCP traffic. (a) Bit rate. (b) Buffer level.

TABLE IV
ONE CLIENT, WIFI LINK, WITH BACKGROUND TCP TRAFFIC

Algorithm	Interruptions	Avg. video rate (Kbps)	Resolution changes
FDASH	0	713.378	21
AASH	0	606.145	33
OSMF	0	704.688	173
SVAA	150	850.777	13
RAAHS	0	425.530	69
SFTM	0	677.657	60

video interruptions, the average download rate, and the number of resolution changes achieved per algorithm. We observe that FDASH, SVAA, AAASH, and RAAHS achieve a steady bit rate and avoid unnecessary adjustments of video resolution. On the contrary, OSMF and SFTM perform frequent resolution changes among high and low video bit rates; in addition, OSMF presents 65 video interruptions. Moreover, FDASH achieves the higher video representations, providing an average download bit rate of 1070 kb/s, whereas the SVAA algorithm, achieving the next higher download bit rate, has an average of 981 kb/s. In addition, as shown in Fig. 5(b), the rate adaptation algorithms keep the playback buffering time above the target time limits, apart from RAAHS, which occasionally fails to preserve its buffering time to acceptable levels.

D. Single Client per Method Over a WiFi Network With Background TCP Traffic

Fig. 6 illustrates the dynamics of the rate adaptation algorithms in the case of a WiFi environment with multiple flows and channel interference. The experiment included one client per algorithm and five background traffic Transmission Control Protocol (TCP) flows. From the obtained results, it can be observed that FDASH adjusts the download bit rate to the available throughput quite efficiently, trying to provide the best possible resolution to the client while at the same time avoiding buffer starvation. Moreover, as presented in Table IV, higher video representations are achieved by SVAA with an average download bit rate of 851 kb/s, although it performs 150 video stalls during the simulation. In contrast, FDASH achieves an average download bit rate of 713 kb/s, and its target buffering time is above the target limit in most of the simulation time.

E. Multiple Clients per Method Attempting to Access a DASH Server Over a 2-Mb/s Line With No Background Traffic

In Figs. 7–11, we evaluate the performance of FDASH against the other algorithms by utilizing two clients per method that compete for access to the server over a shared 2-Mb/s link. The evolution of the bit rate and the evolution of buffering time can be seen in each of the subfigures. FDASH compared with AAASH, OSMF, RAAHS, and SVAA algorithms results in a higher average download bit rate with no buffer underflows. In particular, FDASH clients show a marginal increase of the download bit rate to 2% when they compete for access to the server with AAAHS and SVAA clients. However, FDASH clients show a download bit rate increase of 22% and 30% when competing with OSMF and RAAHS clients, respectively. In addition, the average buffer underflow counts were 118 for OSMF, 4 for RAAHS, and 279 for SVAA clients; whereas in the case of FDASH clients, video interruptions do not occur. In addition, FDASH performs less video resolution changes compared with the other schemes. Particularly, FDASH has an average of 15 video resolution changes, whereas AAASH, OSMF, RAAHS, and SVAA exhibit 17, 165, 64, and 17 changes, respectively. The comparison of FDASH with SFTM, as illustrated in Fig. 10, shows that SFTM provides higher video resolutions to its clients. However, in the later case, FDASH clients achieve a steady bit rate with an average of 24 video resolution changes, whereas SFTM clients perform more than 104 changes. Moreover, video stalls do not exist for both schemes. Thus, the results confirm the effectiveness of FDASH concerning the quite stable video resolutions and the fairness provided to each client.

F. Multiple Clients per Method Attempting to Access a DASH Server Over a WiFi Network With Background Traffic

The previous scenario is replicated over a WiFi environment and by considering that the background traffic consisted of three TCP traffic flows. The obtained results presented in Figs. 12–16 show that FDASH clients keep the video resolution at a higher level most of the time and at the same time retain video resolution changes to a minimum possible level when competing for access with clients using the other adaptation algorithms. In particular, FDASH clients show a download bit rate increase

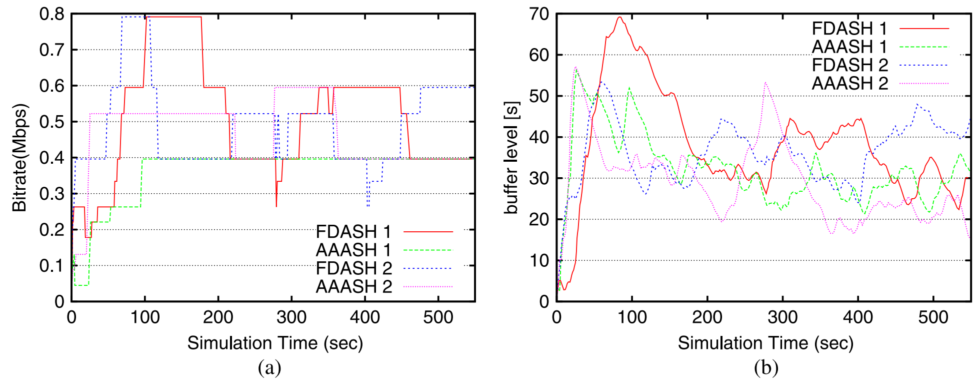


Fig. 7. Two AAASH and two FDASH clients, 2-Mb/s link, with no background traffic. (a) Bit rate. (b) Buffer level.

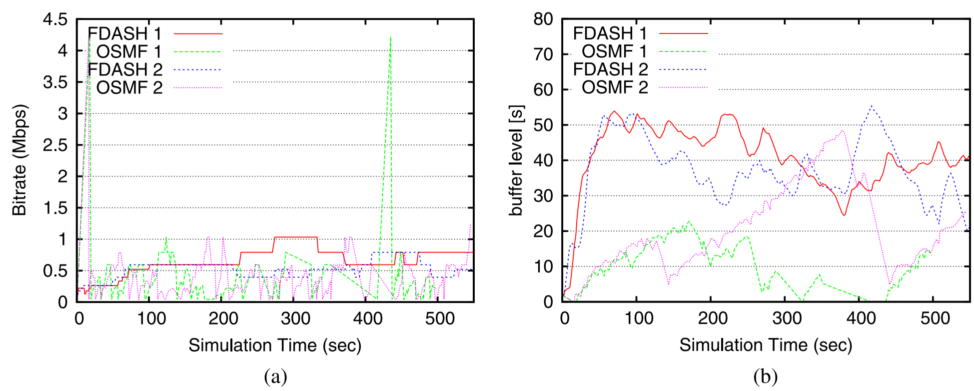


Fig. 8. Two FDASH and two OSMF clients, 2-Mb/s link, with no background traffic. (a) Bit rate. (b) Buffer level.

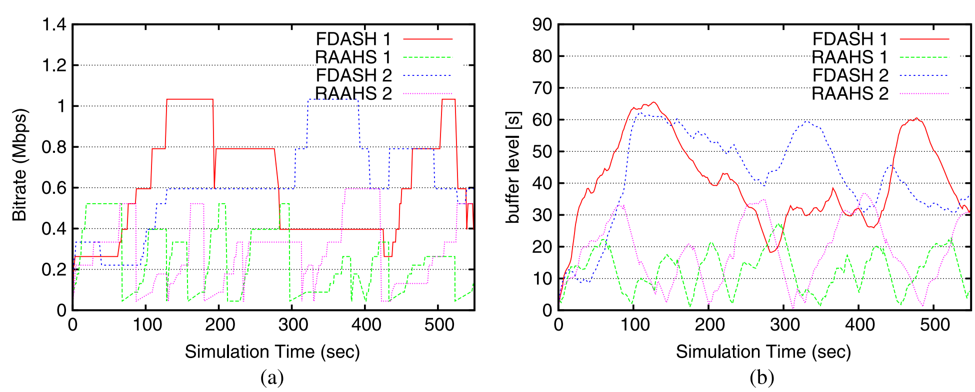


Fig. 9. Two FDASH and two RAAHS clients, 2-Mb/s link, with no background traffic. (a) Bit rate. (b) Buffer level.

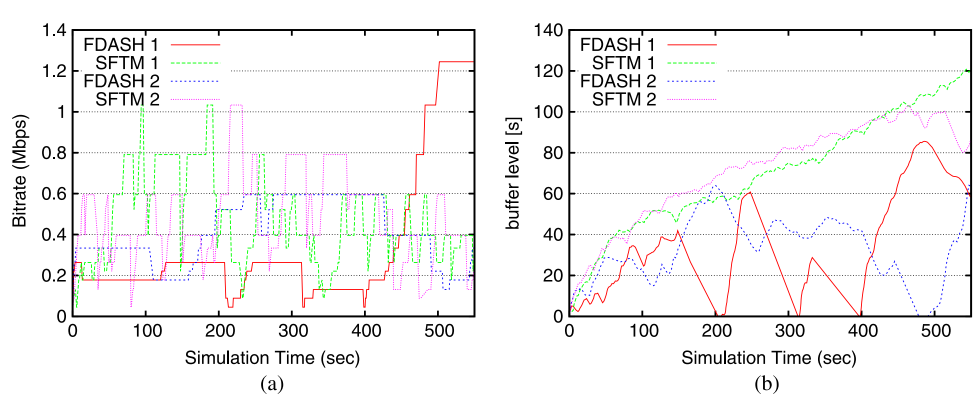


Fig. 10. Two FDASH and two SFTM clients, 2-Mb/s link, with no background traffic. (a) Bit rate. (b) Buffer level.

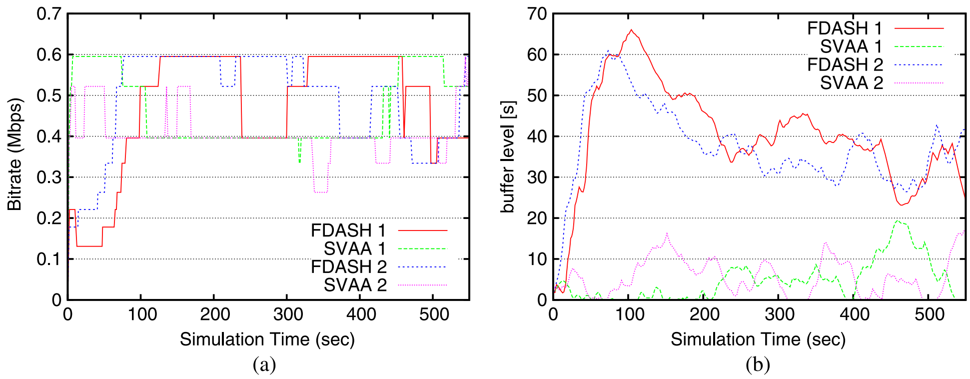


Fig. 11. Two FDASH and two SVAA clients, 2-Mb/s link, with no background traffic. (a) Bit rate. (b) Buffer level.

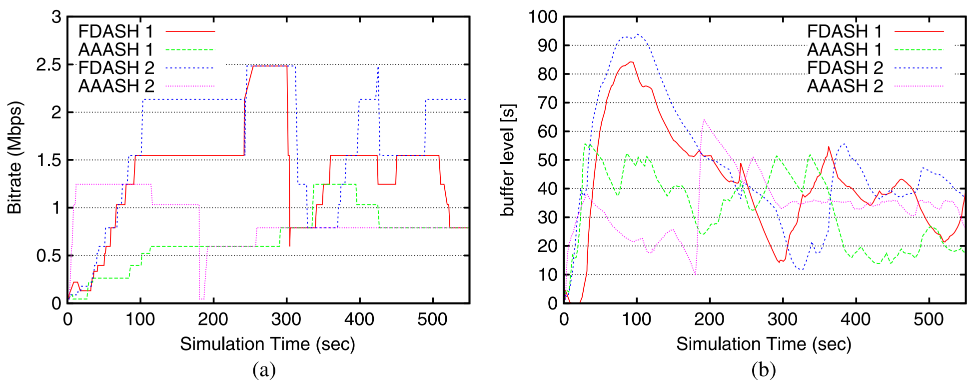


Fig. 12. Two AASH and two FDASH clients, WiFi, with no background traffic. (a) Bit rate, (b) Buffer level.

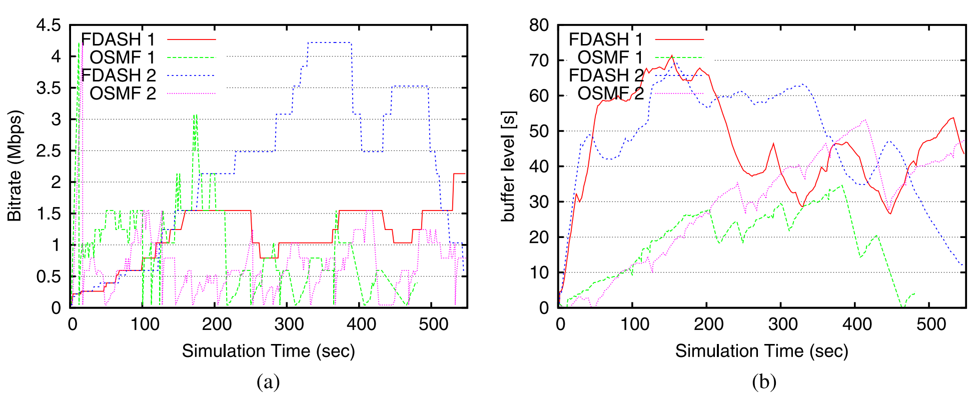


Fig. 13. Two FDASH and two OSMF clients, WiFi, with no background traffic. (a) Bit rate. (b) Buffer level.

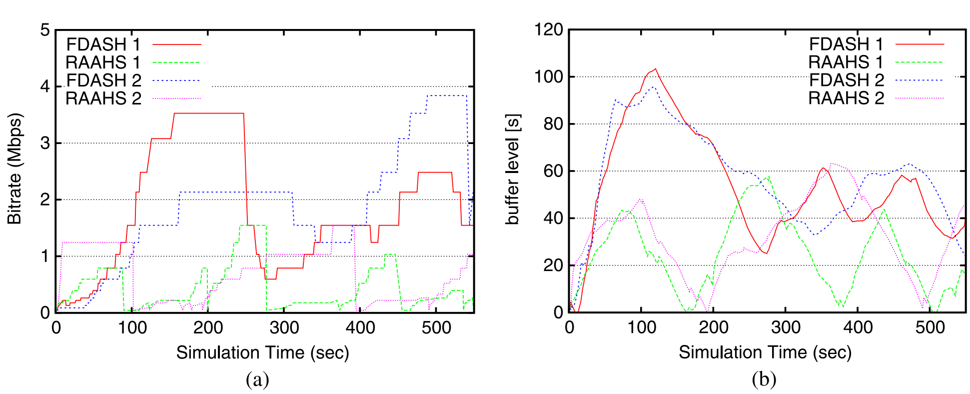


Fig. 14. Two FDASH and two RAAHS clients, WiFi, with no background traffic. (a) Bit rate. (b) Buffer level.

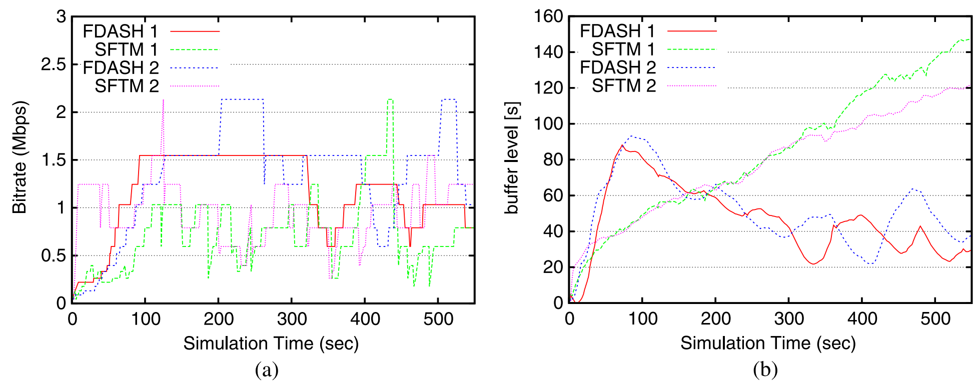


Fig. 15. Two FDASH and two SFTM clients, WiFi, with no background traffic. (a) Bit rate. (b) Buffer level.

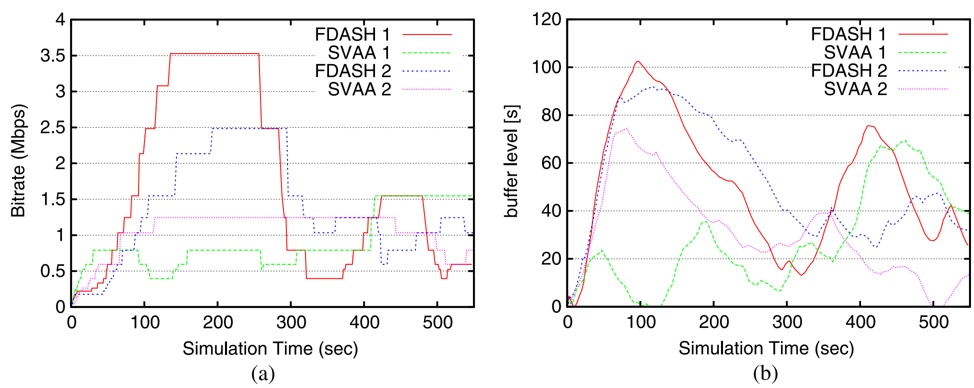


Fig. 16. Two FDASH and two SVAA clients, WiFi, with no background traffic. (a) Bit rate. (b) Buffer level.

of 27%, 31%, 35%, 13%, and 13% when they compete for access to the server with AAASH, OSMF, RAAHS, SVAA, and SFTM clients, respectively. Moreover, FDASH presents an average of 23 video resolution changes, whereas AAASH, OSMF, RAAHS, SVAA, and SFTM exhibit 23, 129, 65, 74, and 18 average changes. Finally, buffer depletes appear only to OSMF and SVAA clients, exhibiting more than 60 and 17 video interruptions per average, respectively.

VI. Conclusion and Future Work

In this paper, we have compared the performance of FDASH against other rate adaptation algorithms over a link of variable throughput and in WiFi environments, with cross traffic and channel interference in terms of the following: 1) the attainable download bit rate; 2) the video resolution changes; 3) the playback buffering times; and 4) the buffer underflow incidents. The obtained results showed that FDASH successfully provides the client with the optimal video rates, avoiding buffer underflows and unnecessary changes of video resolution in case of fluctuations of the available connection throughput.

Directions for future work include the realization of further wide-scale simulation trials in order to measure the benefits of the proposed scheme in terms of QoE. More specifically, we plan to integrate the pseudosubjective quality assessment [25] to ns-3 in order to evaluate the performance of our algorithm in terms of QoE. Furthermore, our future plans include the enhancement of FDASH algorithm with parallel fetching mechanisms, enabling the retrieval of segments concurrently from different servers, so that FDASH could operate more effectively and exploit the benefits of CDNs.

References

[1] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*. New York, NY, USA: Academic, 2013.
[2] J. B. Othman, L. Mokdad, and S. Ghazal, "Performance analysis of WiMAX networks AC," *Wireless Pers. Commun.*, vol. 74, no. 1, pp. 133–146, Jan. 2014.
[3] G. Mastorakis, E. Pallis, C. Mavromoustakis, and A. Bourdena, *Efficient Resource Management Utilizing Content-Aware Multipath Routing*, vol. 3, ser. Modeling and Optimization in Science and Technologies. New York, NY, USA: Springer-Verlag, 2014.
[4] C. Mavromoustakis, A. Bourdena, G. Mastorakis, E. Pallis, and G. Kormentzas, "An energy-aware scheme for efficient spectrum utilization in a 5G mobile cognitive radio network architecture," *Telecommun. Syst.*, vol. 59, no. 1, pp. 63–75, May 2015.
[5] A. Ksentini and T. Taleb, "QoE-oriented adaptive SVC decoding in DVB-T2," *IEEE Trans. Broadcast.*, vol. 59, no. 2, pp. 251–264, Jun. 2013.
[6] G. Baltoglou, E. Karapistoli, and P. Chatzimisios, "IPTV QoS and QoE measurements in wired and wireless networks," in *Proc. IEEE GLOBECOM*, Dec. 2012, pp. 1757–1762.
[7] C. Mavromoustakis *et al.*, "Dynamic cloud resource migration for efficient 3D video processing in mobile computing environments," in *Novel 3D Media Technologies*. New York, NY, USA: Springer-Verlag, 2015, pp. 119–134.
[8] N. Bouten *et al.*, "Deadline-based approach for improving delivery of SVC-based HTTP Adaptive Streaming content," in *Proc. IEEE NOMS*, 2014, pp. 1–7.
[9] A. J. Estepa, R. Estepa, J. Vozmediano, and P. Carrillo, "Dynamic VoIP codec selection on smartphones," *Netw. Protocols Algorithms*, vol. 6, no. 2, pp. 22–37, 2014.

[10] D. J. R. Jiménez, M. José, S. Sendra, and J. Lloret, "Choosing the best video compression codec depending on the recorded environment," in *Proc. IEEE GLOBECOM*, 2014, pp. 1364–1369.

[11] J. Lloret, A. Canovas, J. Tomas, and M. Atenas, "A network management algorithm and protocol for improving QoE in mobile IPTV," *Comput. Commun.*, vol. 35, no. 15, pp. 1855–1870, Sep. 2012.

[12] J. Lloret, M. Garcia, M. Atenas, and A. Canovas, "A QoE management system to improve the IPTV network," *Int. J. Commun. Syst.*, vol. 24, no. 1, pp. 118–138, Jan. 2011.

[13] J. Seppänen, M. Varela, and A. Sgora, "An autonomous QOE-driven network management framework," *J. Vis. Commun. Image Represent.*, vol. 25, no. 3, pp. 565–577, Apr. 2014.

[14] D. J. Vergados, A. Michalas, A. Sgora, and D. D. Vergados, "A control-based algorithm for rate adaption in MPEG-DASH," in *Proc. IEEE 5th Int. Conf. IISA*, 2014, pp. 438–442.

[15] D. J. Vergados, A. Michalas, A. Sgora, and D. D. Vergados, "A fuzzy controller for rate adaptation in MPEG-DASH clients," in *Proc. 25th IEEE Int. Symp. PIMRC*, 2014, pp. 2008–2012.

[16] A. Alvarez *et al.*, "Adaptive streaming: A subjective catalog to assess the performance of objective QoE metrics," *Netw. Protocols Algorithms*, vol. 6, no. 2, pp. 123–136, 2014.

[17] A. Canovas, M. Garcia, J. Lloret, M. Atenas, and R. Rizo, "Improving IPTV QoE taking the suitable MPEG-2/MPEG-4 quantizer based on jitter, delay and lost packets measurements," *Int. J. Adv. Telecommun.*, vol. 3, no. 3/4, pp. 16–26, 2011.

[18] R. Radhakrishnan, B. Tirouvengadam, and A. Nayak, "Cross layer design for efficient video streaming over LTE using scalable video coding," *Netw. Protocols Algorithms*, vol. 4, no. 4, pp. 101–125, 2012.

[19] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *Proc. 19th Int. IEEE PV Workshop*, 2012, pp. 173–178.

[20] C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj, "Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network," *Signal Process., Image Commun.*, vol. 27, no. 4, pp. 288–311, Apr. 2012.

[21] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proc. 8th Int. CoNEXT*, New York, NY, USA, 2012, pp. 109–120. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413190

[22] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. 2nd Annu. ACM Conf. MMSys*, New York, NY, USA, 2011, pp. 169–174. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943575

[23] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: A QoE-aware DASH system," in *Proc. 3rd MMSys Conf.*, New York, NY, USA, 2012, pp. 11–22.

[24] Z. Bingül and O. Karahan, "A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 1017–1031, 2011.

[25] G. Rubino, "Quantifying the quality of audio and video transmissions over the internet: The PSQA approach," in *Design and Operations of Communication Networks: A Review of Wired and Wireless Modeling and Management Challenges*. London, U.K.: Imperial College Press, 2005.

Dimitrios J. Vergados (M'04) received the Diploma degree in electrical and computer engineering and the Ph.D. degree in wireless networks from the National Technical University of Athens, School of Electrical and Computer Engineering.

Since then, he has worked as a Postdoctoral Researcher and a Visiting Lecturer in several universities in Greece and abroad. He is currently with the Department of Computer Engineering and Informatics, University of Patras, Patras, Greece. His research interests include wireless networks, simulation modeling, wireless sensor networks, wireless ad hoc networks, scheduling algorithms, and multihop networks. He has authored several publications in these areas.

Angelos Michalas (M'11) received the Ph.D. degree from the National Technical University of Athens, Zografou, Greece.

He is currently an Associate Professor and the Head of the Department of Informatics Engineering with the Technological Educational Institute of Western Macedonia, Kastoria, Greece. His research interests focus on the design and performance evaluation of wireless and wired broadband communication networks, network management and quality-of-service/quality-of-experience provisioning, distributed systems and algorithms, and cloud computing. He has authored several peer-reviewed publications in these areas.

Dr. Michalas is a member of the Technical Chamber of Greece. He serves as a reviewer of international journals and conferences.

Aggeliki Sgora received the B.Sc. degree in mathematics, the M.Sc. degree in technologies and management of information and communication systems, and the Ph.D. degree in computer networks from the University of the Aegean, Athens, Greece.

She has been a Postdoctoral Researcher and an Adjunct Lecturer in several universities in Greece and abroad and has been actively involved in a number of European and national research projects. She is currently a Postdoctoral Researcher with the University of Piraeus, Piraeus, Greece. She has several publications in journals, books, and conference proceedings. Her research interests include communication networks, quality of service/quality of experience, resource management, vehicular communications, cloud computing, and media streaming.

Dr. Sgora is a reviewer in several journals.

Dimitrios D. Vergados (SM'11) received the Ph.D. degree from the National Technical University of Athens, Zografou, Greece.

He is currently an Assistant Professor with the Department of Informatics, University of Piraeus, Piraeus, Greece. He has several publications in journals, books, and conference proceedings. His research interests are in the area of computer networks and telecommunication systems, cloud computing and green technologies, and computer vision.

Dr. Vergados has served as a committee member and evaluator in national and international organizations and agencies and as a chair and a technical program committee member in several international conferences. He is a member of the editorial board in international journals and a reviewer in several journals and conferences.

Periklis Chatzimisios (SM'12) received the B.Sc. degree from Alexander Technological Educational Institute (TEI) of Thessaloniki, Thessaloniki, Greece, in 2000 and the Ph.D. degree from Bournemouth University, Dorset, U.K., in 2005.

He is currently an Associate Professor with Alexander TEI of Thessaloniki. Recently, he has been a Visiting Academic/Researcher with the University of Toronto, Toronto, Canada, and the Massachusetts Institute of Technology, Cambridge, MA, USA. He is the author/editor of eight books and over 85 peer-reviewed papers on the topics of performance evaluation and standardization activities of mobile/wireless communications with over 1300 citations received by other researchers.

Dr. Chatzimisios is involved in several standardization activities serving as a Member of the Standards Development Board of the IEEE Communication Society (2010 to present) and, lately, as an active member of the IEEE Research Groups on IoT Communications and Networking Infrastructures and on Software Defined and Virtualized Wireless Access. He holds editorial board positions for several IEEE/non-IEEE journals.