A NEW QUALITY OPTIMIZATION FRAMEWORK FOR DASH STREAMING OVER WIRELESS CHANNELS

Leonardo Favario and Enrico Masala

Control and Computer Engineering Department, Politecnico di Torino, Torino, Italy
leonardo.favario@polito.it, masala@polito.it

ABSTRACT

Mobile devices are increasingly used as terminals for playback of multimedia content. However, maximizing the user's quality of experience is challenging due to the highly variable conditions of the wireless channels. A possibility to cope with such a variability is to dynamically adapt the source coding rate during the transmission, which is the underlying idea of the DASH standard. This work proposes a new framework to improve the quality of the DASH-based streaming experience by allowing to adjust the tradeoff between the quality of received content and the risk of playback freeze due to an empty buffer, which is a strong quality-disruptive event. The problem is analytically formulated and an efficient method to compute the playback freeze probability as a function of the representation choices over time is presented. Numerous simulation results using real download rate traces of 3G channels show the performance improvement compared to other bandwidth-adaptive algorithms as well as the robustness of the framework to variations of its most important parameters.

Index Terms— DASH, mobile video streaming, video quality, bandwidth adaptation, freeze probability

1. INTRODUCTION

Multimedia communications are currently a large share of the traffic on packet networks, both wired and wireless. In recent years, multimedia communication and streaming in particular shifted from using streaming servers based on the UDP/RTP protocol [1], to the so called HTTP streaming, where transmission uses the reliable TCP protocol which supports HTTP communications. In this scenario the client control the communication by means of its requests.

Standards such as the MPEG Dynamic Adaptive Streaming over HTTP (DASH) [2] follows this paradigm, standardizing how to describe multimedia resources available on a server and how to segment them to allow easier adaptation of the communication to the channel conditions. The client is, in fact, free to choose any representation of the resource which deems appropriate for the current channel conditions, requesting segments of such a representation and potentially changing the representation each time a new segment has to be requested.

The standard does not specify any client adaptation logic, since this is both not required for interoperability and it would be counterproductive due to the several different scenarios in which clients can operate. However, an adaptation logic is needed for a client to take the maximum advantage of all the representations available on the server, that is, to optimize the quality of the communication subject to the available resources.

In order to devise effective adaptation strategies, different aspects should be taken into account. First, estimating the available channel bandwidth is important since wrong estimation may cause playback freezes due to the impossibility to receive the multimedia data on time for playback. A typical case could be requesting representations whose bandwidth is excessive compared to the available channel bandwidth. Second, several decisions must be taken during the communication. At the end of the download of each DASH segment, the next one can be requested. At this time it is possible to change the representation, if needed, in order to increase the quality or, conversely, to decrease it to reduce the bandwidth requirement. How to optimize such decisions is currently the target of several research efforts.

Multimedia communications based on the UDP/RTP protocol have received a lot of attention and even complex rate-distortion optimized frameworks are available for such systems [3]. For the case of HTTP-based multimedia communication systems similar approaches can be used to formulate the problem, but one of the major difficulties is that DASH does not provide, at least in a standardized format in the Media Presentation Description (MPD), detailed quality parameters that can be used to perform such a rate-distortion optimization.

Therefore, several heuristic algorithms have been proposed to dynamically decide which representation should be requested each time is it possible to download a new segment. The main parameters considered by such systems are the current buffer level, i.e., how far is the playback system from a freeze event in case the channel degrades significantly, and an estimate of the current channel conditions. Both values are easily available at the client. An example of such an approach

The final publication is available at IEEE via: http://dx.doi.org/10.1109/ICME.2015.7177479

is proposed in [4], which considers both a threshold for the buffer level and a conservative approach in the representation selection, i.e., a slow rate increase if the channel conditions are significantly good and a rapid decrease in the opposite case.

Other researchers proposed methods to adapt the classical rate-distortion optimization frameworks also to the case of HTTP streaming. For instance, the work in [5] tailors a classical rate-distortion optimization approach to the case of HTTP streaming using segmented content. However, the proposed system relies on the availability of information about the distortion values corresponding to each segment which, in a DASH system, should be conveyed outside the MPD. Others suggest to use a Markov decision process to decide, each time a new segment can be requested, which is the best decision to take, as in [6]. However, the system relies on the definition of a series of rewards associated with the states of the process whose tuning is critical to achieve the desired performance.

In this work we propose a new framework that optimizes the video quality of the communication by maximizing a quality measure based on the chosen representation and its variation over time, while at the same time it keeps the the probability of having freezes, a strong quality-disrupting event, under a given threshold. Since in a real bandwidth-limited scenario maximizing the quality typically implies taking advantage, to the maximum extent, of the available bandwidth to request the best representations, the buffer level will always be critical for good performance and in particular to avoid freezes. We argue that being able to explicitly track the probability of such events allows to better tune the tradeoff between quality and uninterrupted playback.

The paper is organized as follows. Section 2 provides background information about the adaptive HTTP streaming and the DASH standard. The proposed framework is detailed in Section 3, followed by Section 4 that presents the simulation setup. Results are presented and discussed in Section 5. Finally, conclusions are drawn in Section 6.

2. ADAPTIVE HTTP STREAMING WITH DASH

The HTTP protocol has been deemed for long time unsuitable for multimedia communications. However, in recent years the popularity of approaches based on HTTP for streaming applications has radically changed the perception.

In fact, the MPEG DASH [2] standardizes such an approach, aiming at delivering multimedia content over Internet using HTTP-based streaming. Following the typical MPEG philosophy, the specification covers all the aspects necessary for interoperability, in particular the Media Presentation Description (MPD) and the format of the segments in which the content is subdivided. All other aspects are left outsize the scope of the standard itself.

DASH naturally provides adaptation, since a key DASH feature is the possibility to make multiple representations, i.e., quality levels, available to the client. Such levels may be provided by several different encodings of the same content at different bitrates, or using other approaches such as scalable coding systems.

All resources represented by means of the DASH standards are to be split into segments that can be individually addressed and requested by the clients. Segments of different representations are aligned so that is it possible to switch from one representation to another at their boundaries without incurring in content decoding disruptions. However, a variation in the requested representation clearly affects the user's perceived quality.

While this flexibility is one of the key strengths of DASH, implementing a good adaptation algorithm that maximizes the quality of the communication is not a trivial task especially when there are many representations to choose from. The DASH standard does not provide algorithms or strategies since this is not needed for interoperability, as well as they are highly dependent on the scenario.

However, some information about the representations in order to support adaptation is available through the MPD description. In particular, the size of each segment is typically available either in the form of the number of bytes (by means of byte ranges information) or the average rate of the representation itself. Such information is important for the optimization algorithm. Notably, for the current version of the standard no consensus has been reached in order to insert quality information attached to each representation, partly due to the different ways in which quality might be defined. Therefore, this is a limitation that has to be taken into account by a generic optimization algorithm.

However, the use of a reliable transmission protocol such as HTTP over TCP partly mitigates this issue with respect to the case of UDP/RTP streaming since it assures that the data is either available in an uncorrupted form or it is not available at all. Therefore, most of the DASH-based systems focus on avoiding interruptions (freezes) in the playback process and on minimizing the number of switches between different representations, since this might be annoying for the user [7].

3. OPTIMIZATION FRAMEWORK

Consider a multimedia content, such as video, that is divided into segments, each one of them representing a certain time interval of the media content, e.g., 2 seconds. Each segment is encoded into several representations, with different quality and consequently average bitrate. The bitrates are known to the server and they are communicated to the client by means of the MPD.

The client performs HTTP download requests to the server in a sequential manner, requesting segments that represents temporally consecutive parts of the original content. After each request, the client waits for the download to be completed, makes the segment available for playback and de-

The final publication is available at IEEE via: http://dx.doi.org/10.1109/ICME.2015.7177479

cides which representation has to be requested for the next segment. At the same time, the client also performs an estimate of the available download bandwidth on the basis of the download time of the latest segment.

More formally, let N_c be the number of frames in each segment, f be the frame rate of the video content, S be the total number of segments of the video sequence, b the current number of frames in the client buffer, and $\{r_0, ..., r_{R-1}\}$ the set of the average bitrates of the R available representations. We also define a set $\{q_0, ..., q_{R-1}\}$ which relates each representation to a certain quality value. The global quality Q of a video playback experience depends on the set of representations $\{\pi_0, ..., \pi_{S-1}\}$ chosen for each segment i, where $\pi_i \in \{0, ..., R-1\}$ corresponds to the index of the representation in the set of average bitrates or quality.

Each time the client has the possibility to begin downloading a new segment s the following problem must be solved:

$$\begin{array}{rl} \max_{\{\Pi\}} & Q = \frac{1}{S}\{q_{\pi_0} + \sum_{j=1}^{S-1}(q_{\pi_j} - \beta|\Delta q_{\pi_j,\pi_{j-1}}|)\} \\ \texttt{s.t.} & P(b_k < 0) < P_T, k \in \{s, ..., S-1\} \end{array} \quad (1)$$

where $\Delta q_{\pi_j,\pi_{j-1}} = q_{\pi_j} - q_{\pi_{j-1}}$ is the quality variation from the $(j-1)$-th segment to the j-th segment, β is a penalty factor for the change, $P(b_k < 0)$ is the probability that the client buffer is empty at any future downloading decision for segment k, P_T is a probability threshold and Π is the set of all possible ordered combinations of $(\pi_i, ..., \pi_{S-1})$ values which have not already been decided. P_T expresses the willingness of the client to risk that variations in the available bandwidth make the playback freeze. Taking a higher risk means that it is typically possible to achieve a better quality Q.

When a new downloading decision can be taken, a number $s-1$ of $\pi_j, j \in \{0, ..., s-1\}$ have already been decided since the corresponding HTTP request have already been sent. The future $\pi_j, j \in \{s, ..., S\}$, instead, need to be determined in order to solve the optimization problem.

One of the key difficulties lies in computing the term $P(b_k < 0)$ which is, of course, function of $\pi_j, j \in \{s, ..., S\}$, as well as of the future available channel download bandwidth. When the download decision for segment s can be taken, the number of frames in the buffer b_s is known. Let Δb_s the variation of b_s just after the next segment has been completely downloaded:

$$\Delta b_{s+1} = N_c - N_c \frac{r_{\pi_s}}{d_s} \quad (2)$$

where d_s is the effective download rate of the next segment s, which is not known when the decision is taken. Therefore, Δb_s has to be considered a random variable. Eq.(2) can be solved for d_s:

$$d_s = \frac{r_{\pi_s} \cdot N_c}{N_c - \Delta b_{s+1}} \quad (3)$$

Assuming that the download rate d_s remains constant during the segment download it is possible to compute the probability that a given range of Δb_{s+1} values happen by integrating the pdf of d_s between the extremes that yield the desired range. Therefore, considering the case $\Delta b_{s+1} \in (-\infty, 0)$ it is possible to compute the $P(b_k < 0)$ value.

3.1. Bandwidth Estimation

We propose to compute an estimate $\hat{d}_s$ of the future average download bandwidth d_s for segment s by considering it as a normally distributed random variable with mean and variance as described in the following. The mean is computed on the basis of the previously observed download bandwidth values according to an exponential average with parameter α:

$$\hat{d}_s = \alpha d_{s-1} + (1-\alpha)\hat{d}_{s-1}. \quad (4)$$

Each time a new segment is completely downloaded this estimation is updated. The variance σ^2 of the future average download bandwidth for the next segment is instead assumed to be constant. Although this might seem a quite rough estimation of the channel behavior, the results section will show that the performance is sufficient to perform an effective optimization.

3.2. Freeze Probability Estimation

Given the current number of frames in the buffer b_j (i.e., the buffer level) and the pdf of the download bandwidth it is possible to compute the pdf of the buffer level b_{j+1} at the end of the download of the segment j, from which the estimation of $P(b_{j+1} < 0)$ is straightforward. The process can be repeated for subsequent downloads, considering the probability of each possible value of the buffer level b_{j+1} and reapplying the same procedure.

The situation is depicted in Fig. 1, where the procedure is represented by means of a trellis, in which at each stage a new segment is downloaded, and each node at the same stage represents a different amount of frames in the buffer.

Given a certain sequence of representation requests for future segments $\pi_j, j \in s, ..., S$, it is possible to determine the probability of being in any node of the trellis by multiplying the probability of the previous node with the probability of the transition to the current node which is equal to the one of the Δb_j values that lead to that node. The freeze probability, i.e., $P(b_k < 0), k \in \{s, ..., S\}$, is simply the probability of the corresponding black node in the trellis.

Note that from a black node there is only one edge to the next black node. This is because we are interested in estimating the probability that, at any new segment request in the future, there has been a freeze event.

As we move on in the trellis towards the right (i.e., towards future segment requests) the number of nodes increases. Although some probabilities will become negligible and can be neglected for the purpose of computing $P(b_k < 0)$, the complexity of the computation significantly increases.

The final publication is available at IEEE via: http://dx.doi.org/10.1109/ICME.2015.7177479

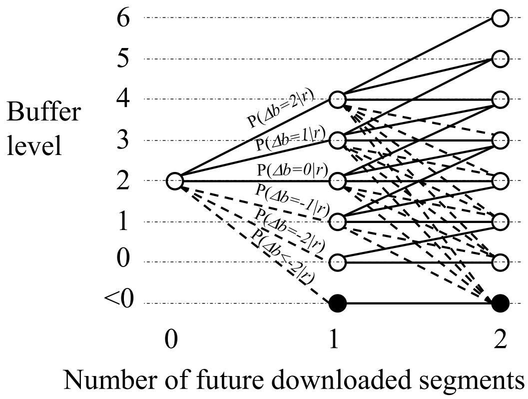

Fig. 1. The trellis representing the possible situations of buffer level once a given number of segments have been downloaded (2 in the figure). The size of the segment is assumed to be 2 frames. For all dashed edges the download rate is lower than the rate of the requested representation r. The objective is to compute the probability of the black-filled nodes representing a freeze event.

For this reason, as well as because the estimation of the download rate of the channel in the future is more and more uncertain as the distance from a known value increases, we limit the number of future segments considered, thus limiting the complexity of the estimation. The results section will show that considering just two or three future segments yields a satisfactory performance.

Therefore, if N future segments are considered, the problem stated in Eq. (1) reduces to:

$$\begin{array}{ll} \max_{\{\Pi'\}} & Q = \frac{1}{S}\{q_{\pi_0} + \sum_{j=1}^{s-1}(q_{\pi_j} - \beta|\Delta q_{\pi_j,\pi_{j-1}}|) + \\ & \sum_{j=s}^{s+N-1}(q_{\pi_j} - \beta|\Delta q_{\pi_j,\pi_{j-1}}|)\} \\ \texttt{s.t.} & P(b_k < 0) < P_T, k \in \{s, ..., s+N-1\} \end{array} \quad (5)$$

where part of the expression of Q (the first row) is constant and Π' represents the set of all possible ordered combinations of the next $(\pi_s, ..., \pi_{s+N-1})$ values which have not already been decided.

3.3. Solving the Optimization Problem

All elements are now available to model the system as a Markov decision process. At any time a request decision can be made there is a limited set of options (i.e., the index of the requested representation for the next segment π_s) that lead to different states, characterized by a buffer level and a video quality, with certain probabilities. Note that the buffer level and video quality characterizing each state can be computed by using only those of the states in the previous stage. Hence, the Markov property is satisfied.

Therefore, given a certain threshold for the freeze probability, i.e., the P_T value, the maximization problem in Eq. (5) can be solved by exploring the various states of each trellis stemming from each combination of $(\pi_s, ..., \pi_{s+N-1})$. Once the $P(b_k < 0)$ values and corresponding quality values are known, it is trivial to find the state that maximizes the quality subject to the P_T constraint.

Note that the computational burden of the operation can be reduced by means of precomputing the probabilities of the edges in the trellis, which can be reused for all the trellises that correspond to the explored combinations of $(\pi_s, ..., \pi_{s+N-1})$ values. In particular, the quantities $P(\Delta b|r)$ can be precomputed for all possible rates r of the representations and for all the Δb values needed.

Finally, note that in order to reduce the complexity, each node could be used to represent a range of frames B_T in the buffer and not a single integer value. This may greatly reduce the number of operations both in terms of both the Δb values needed and the combinations of probabilities to get the value for the nodes in the trellises corresponding to $P(b_k < 0)$. Though such an approach may reduce the precision of the computation of $P(b_k < 0)$, however, the results section will show that an acceptable performance can still be achieved even with such approximation.

4. SIMULATION SETUP

To assess the performance of the proposed algorithm several simulations have been run. We considered DASH streaming sessions of a 40-second long video clip. Each segment is 2 s, so the video clip is composed of 20 segments. The used representations are shown in Table 1 in terms of bitrate. Moreover, we associated a quality value to each representation. With reasonably spaced rate intervals between the representations, we assume that adjacent representations provide equal quality increments from the point of view of the final user. However, any reasonable value is suitable for the proposed framework. Though it would also be possible to compute the quality of each segment using objective quality metrics such as PSNR, we preferred to use such quality indexes in order to better parametrize the annoyance caused by the switch from one representation to another one as expressed by the β parameter in the Q term of Eq. (1). However, the results section will investigate the influence of such a parameter on the performance.

We also compared the performance of the technique proposed in our framework with the one provided by the technique described in [4]. This uses, as main input parameters, the current buffer level, compared with a predefined threshold to determine if the buffer is too empty, and an adjustable parameter γ to decide when moving towards a lower bitrate representation (switch down). If the ratio between the bitrate of the downloaded segment and its actual fetch time is lower than γ, a switch down is performed.

We simulate the transmission by relying of actual rate traces captured from 3G HTTP download sessions using a mobile device. Figure 2 shows the bitrate of the eight traces

The final publication is available at IEEE via: http://dx.doi.org/10.1109/ICME.2015.7177479

repr ID	rate	quality
#0	200	1.0
#1	300	1.5
#2	400	2.0
#3	550	2.5
#4	700	3.0
#5	850	3.5
#6	1000	4.0

Table 1. Average bitrate and assumed quality for each of the representations used in the experiments.

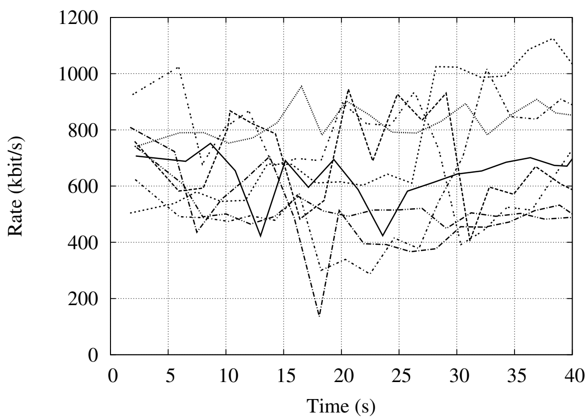

Fig. 2. The rate of the eight HTTP download speed traces used in the experiments as a function of time.

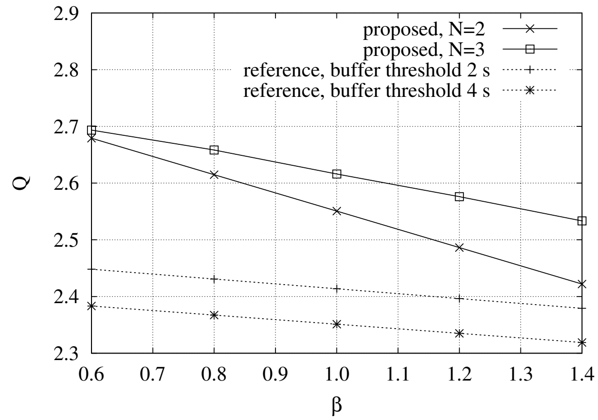

Fig. 3. Comparison of the proposed technique and the one in [4] in terms of video quality as a function of the β parameter. Average over all the traces.

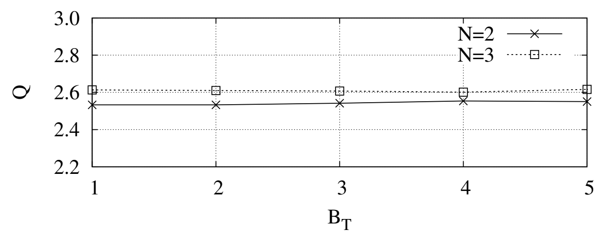

Fig. 4. Average quality performance of the proposed technique over all considered traces as a function of the number of frames B_T represented in each node of the trellis.

used in the experiments as a function of time. It can be clearly noticed a strong rate variability over time. In all simulations, unless otherwise noted, the following parameters have been used: $\beta = 1.0$, $P_T = 0.03$, $\sigma = 100$ kbit/s, $N = 3$, $B_T = 5$.

5. RESULTS

First, we compare the performance of the proposed framework with the technique in [4]. Figure 3 shows the average quality achieved by both techniques when used for transmission over the wireless channel realizations represented by the traces in Fig. 2. The values are shown as a function of the β parameter which models the user's annoyance due to the change in the quality of the representations. The performance of the proposed technique is shown for two N values, i.e., the optimization is performed by considering the freeze probability $P(b_k < 0)$ after the transmission of two or three segments in the future. For the technique in [4], two different buffer level thresholds have been considered, i.e., 2 and 4 s of media content in the buffer, that is, one or two DASH segments. The γ parameter, which influences the step down decision, has been set to 0.8. The proposed technique performs better than the reference one for the whole range of considered β values, hence showing its robustness regardless of the amount of annoyance that is assumed to be perceived by the user in case of transitions between different representations. Note that the central value $\beta = 1.0$ means that a change of one step in the representation basically removes the advantage of doing the change if the representation has higher bitrate, while it doubles the quality penalty when moving towards a lower bitrate representation. No freeze events were experienced in any simulation.

We now analyze the performance of the proposed framework as a function of two important parameters, the number of frames B_T represented by each single node in the trellis and the σ of the download bandwidth of the channel. The former is particularly important to reduce the complexity of the algorithm. Figure 4 shows that the performance is almost constant even if B_T is increased up to 5. Such a value allows a significant complexity reduction since, for an average case of 100 frames in the buffer, only 20 nodes instead of 100 need to be considered in the trellis. More importantly, also the number of edges between the nodes significantly reduces.

Figure 5 presents the performance of the proposed framework as a function of the assumed σ value, which is important for the reliability of the $P(b_k < 0)$ value. The graph shows that the value close to the one measured for the traces considered in this work (i.e., 86) yields a good performance. If the value is significantly changed the performance may degrade but it is still better or equal to the best one of the reference

The final publication is available at IEEE via: http://dx.doi.org/10.1109/ICME.2015.7177479

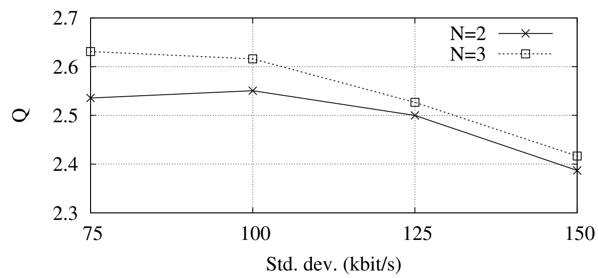

Fig. 5. Behavior of the proposed technique as a function of the σ assumed for the wireless channel.

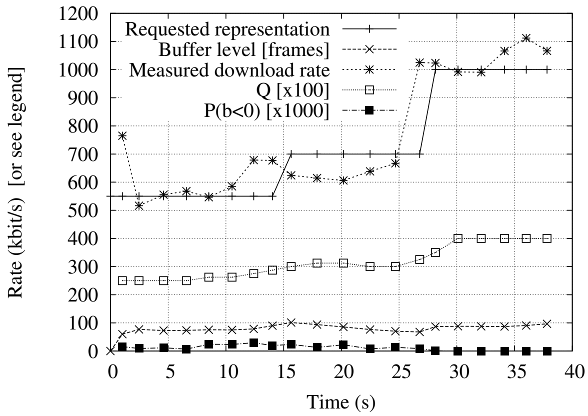

Fig. 6. Behavior of the proposed technique as a function of time.

technique (see Fig. 3 at $\beta = 1.0$.)

Finally, the behavior over time is shown in a typical case. Figure 6 shows, for the proposed technique and one of the traces in Fig. 2, the rate of the selected representation, the buffer level, the freeze probability, the download rate of the last segment and the quality Q. It can be noticed that the freeze probability $P(b_k < 0)$ is kept under the threshold level $P_T = 0.03$ while at the same time the formulation of the quality using $\beta = 1.0$ limits the number of representation changes. The reference technique [4] is also shown in Fig. 7 as a function of time when using the same trace. Comparing the two figures it is possible to notice the advantage of the proposed technique, which closely follows the download rate increase at the end of the trace due to a very low estimated freeze probability which in turn allows a more aggressive behavior, whereas the reference technique only moderately increases the rate in the same time interval.

6. CONCLUSIONS

This paper presented a new framework to improve the quality of the DASH-based streaming experience by proposing a method to tune the tradeoff between the quality of received content and the probability of playback freeze due to an empty buffer. An analytical formulation has been presented, as well as all the necessary elements to efficiently compute the playback freeze probability as a function of the representation request decisions. Results based on simulations using real download rate traces of 3G channels showed that the approach provides better performance compared to other bandwidth-adaptive algorithms. Finally, the robustness of the framework to its most important parameters has also been assessed.

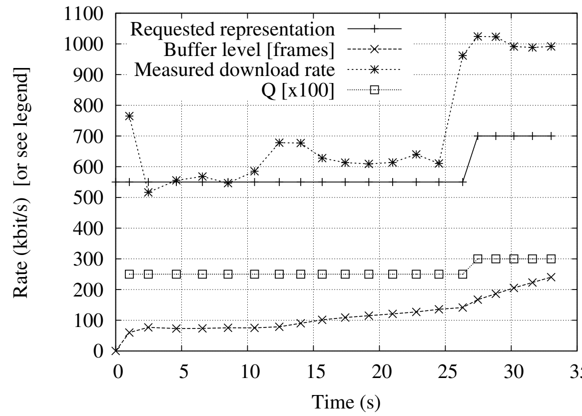

Fig. 7. Behavior of the technique proposed in [4] as a function of time.

7. REFERENCES

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," *RFC 3550*, July 2003.

[2] ISO/IEC 23009, "Dynamic adaptive streaming over HTTP (DASH)," 2012.

[3] P.A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[4] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. of the second annual ACM conference on Multimedia systems (MMSys)*, San Jose, CA, USA, Feb. 2011, pp. 169–174.

[5] S. Mehrotra and W. Zhao, "Rate-distortion optimized client side rate control for adaptive media streaming," in *Proc. of the IEEE Intl. Workshop on Multimedia Signal Processing*, Rio De Janeiro, Brazil, Oct. 2009, pp. 1–6.

[6] S. Xiang, L. Cai, and J. Pan, "Adaptive scalable video streaming in wireless networks," in *Proc. of the 3rd annual ACM conference on Multimedia systems (MMSys)*, Chapel Hill, NC, USA, Feb. 2012, pp. 167–172.

[7] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. de Veciana, "Video quality assessment on mobile devices: Subjective, behavioral and objective studies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 652–671, Oct. 2012.

The final publication is available at IEEE via: http://dx.doi.org/10.1109/ICME.2015.7177479