

RESEARCH

Open Access

Moving QoE for monitoring DASH video streaming: models and a study of multiple mobile clients



Sheyda Kiani Mehr¹, Prasad Jogalekar² and Deep Medhi^{1*}

Abstract

Objective Quality of Experience (QoE) for Dynamic Adaptive Streaming over HTTP (DASH) video streaming has received considerable attention in recent years. While there are a number of objective QoE models, a limitation of the current models is that the QoE is provided after the entire video is delivered; also, the models are on a per client basis. For content service providers, QoE observed is important to monitor to understand ensemble performance *during* streaming such as for live events or concurrent streaming when multiple clients are streaming. For this purpose, we propose *Moving QoE* (MQoE, in short) models to measure QoE *during* periodically during video streaming for multiple simultaneous clients. Our first model MQoE_RF is a nonlinear model considering the bitrate gain and sensitivity from bitrate switching frequency. Our second model MQoE_SD is a linear model that focuses on capturing the standard deviation in the bitrate switching magnitude among segments along with the bitrate gain. We then study the effectiveness of both models in a multi-user mobile client environment, with the mobility patterns being based on traces from a train, a car, or a ferry. We implemented the study on the GENI testbed. Our study shows that our MQoE models are more accurate in capturing the QoE behavior during transmission than static QoE models. Furthermore, our MQoE_RF model captures the sensitivity due to bitrate switching frequency more effectively while MQoE_SD captures the sensitivity due to the magnitude of the bitrate switching. Either models are suitable for content service providers for monitoring video streaming based on their preference.

Keywords: Moving QoE, DASH video, Bitrate switching, Multi-client, Mobility, Service monitoring

1 Introduction

The increasing demand for videos over the Internet with the advent of the ubiquitous mobile devices has led to video streaming being a significant part of Internet traffic. By 2022 this volume is projected to be 82% of all the Internet traffic [1].

Dynamic Adaptive Streaming over HTTP (DASH) for video streaming has been standardized by MPEG [2]. Video content providers have adopted DASH for video streaming to the end users. Briefly, a video is first made available in multiple video bitrate codecs or

representations in DASH. Secondly, the video is divided into small segments with a playback duration that varies from 2 to 15 sec, where a segment is available in each of the multiple bitrate representations. This information is captured in a Media Presentation Description (MPD) file, which is sent to the end user's device at the beginning of video streaming. Adaptive bitrate (ABR) algorithms at the client-side video player are the primary means to optimize video quality [3, 4] from the representations available through the MPD file. The ABR algorithm at the client side attempts to select the best possible bitrates for future segments given the network condition while minimizing bitrate switching from one segment to the next too often [5].

*Correspondence: dmedhi@umkc.edu

¹Department of Computer Science & Electrical Engineering, University of Missouri–Kansas City, Kansas City, USA

Full list of author information is available at the end of the article

A major challenge that video content service providers face is to understand whether the users are adequately receiving satisfactory video quality experience video streaming. Since the content providers serve numerous users at any instance, the interest is to understand on *ensemble* how their services are being received by the users. Furthermore, they want to monitor this periodically *during* streaming. In other words, ensemble quality of video streaming on the fly is an important challenge for content service providers. To our knowledge, this problem from a content providers' perspective has received little attention in the literature, although QoE for video streaming has been an active research topic for a while.

In this work, we propose quantitative QoE models that help to capture the QoE *during* video streaming on the fly. Our models are targeted for QoE monitoring by content providers for video streaming management. We refer to our models broadly as *moving* QoE (MQoE) models. Note that quantitative QoE is commonly used in video QoE literature (see [6, 7] for two extensive surveys), although no previous work has addressed moving QoE models.

There has been a number of works to develop *static* QoE models such as the one proposed with the Model Predictive Control (MPC) approach [5]. Such QoE models are used to calculate a quantitative value *once* the video delivery is completed and for each individual client. On the other hand, such QoE models are not readily usable by video content providers that would want to monitor QoE snapshots in real-time to capture end user experience for numerous videos being streamed.

We present two MQoE models for video delivery monitoring by content service providers when multiple clients watch videos at the same time. Two important metrics that impact quality of video streaming is bitrates selected by clients and the frequency of bitrate switching. Our first model considers the bitrate of segments as well as the sensitivity due to the nonlinear behavior from video quality switching frequencies between segments. We refer to this moving QoE model as MQoE_RF (Moving QoE with rate and frequency). Our second MQoE model captures bitrate changes through the standard deviation of bitrate switching, which we refer to as MQoE_SD. Our moving QoE models are designed to take periodic snapshots for video streaming monitoring. Our MQoE models are particularly suitable for multiple clients watching videos that the content providers would like to assess service delivery.

In order to study our MQoE models and to mimic the perspective of a content provider to monitor the video streaming service, we implemented a multi-client environment accessing video streaming service using the GENI testbed [8]. Our goal was to consider two aspects in our study: first one is related to the specific video being streamed and the second one on user mobility.

For the first aspect, we consider the same video being streamed to multiple clients to conduct a controlled study. Interestingly, the situation of multiple clients accessing the same video is also common in practical situations such as when a video lecture is delivered by an instructor watched by students, or live events when multiple users want to watch the same event such as a sport event. It may be noted that for delivering live contents, YouTube also provides an API that uses DASH [9]. Considering the same video being watched by multiple clients lets us observe whether QoE is being equitably allocated among clients.

For the second aspect in our study, we conducted our study of the MQoE models with mobile clients for users in transit to fully capture the moving QoE behavior. For this, our work considers three different mobility patterns for the clients to emulate traveling in a car, a ferry, or a train.

Considering these two study factors together, our study focuses on multiple clients watching the same video stream while traveling on a ferry or a train, or cars. It is easy to image multiple users on a train or a ferry watching the same live events at the same time. In the case of cars, we envision that more than one user watching the same event on their own mobile devices in the back seat of a car, or users from a fleet of cars traveling in close proximity watching the same event. We point out that our moving QoE model is not dependent on the live scenarios studied in the paper. We considered the live scenario with the same video being watched for a controlled study. Such a study allows us to monitor QoE observed by different clients since they are all watching the same video.

We clarify that the focus of this work is not to devise a new ABR algorithm; rather, for a given ABR algorithm, we present moving QoE models that can be used by content providers for video streaming monitoring and management. For content service providers, it is first important to establish moving QoE models and collect data from monitoring to do a detailed analysis on the service impact; this is the scope and the focus of our paper. Note that seeking to improve quality for end users is a separate problem in itself that could mean changing ABR, increasing bandwidth on outgoing links (from content providers' locations), and possibly making additional business agreements with access providers where the clients are accessing services from; exploration of these issues is outside the scope of our present work.

The novelty of our work is that, to our knowledge, we are the first one to present moving QoE models for video stream monitoring from a content service provider's perspective. Furthermore, our work helps content providers to apply moving QoE models to monitor performance on a continual basis. Since in our experience, content providers give different priorities to different factors, we present two QoE models so that content providers have the freedom to choose one of the models that suit their needs based on

their preferences. As we show through our work, a static QoE model such as MPC QoE model is not applicable for use in a content monitoring environment.

The rest of the paper is as follows. We present our moving QoE models in Section 2. We then present our study environment in Section 3. Initial analysis on parameter selection is discussed in Section 4 before presenting our comparative study on the models and mobility scenarios in Section 5. In Section 6, we discuss the related work. Finally, we present a summary in Section 7.

2 Moving QoE models

Our first moving QoE model, MQoE_RF, considers two QoE metrics: bitrate gain and bitrate switching frequency. Bitrate gain reflects increase in quality to the end users. On the other hand, if bitrate switching frequency happens frequently, the end user may be displeased; that is, this factor negatively impacts overall quality of experience. Thus, our model rewards the bitrate gain while it somewhat penalizes bitrate switching frequency metric. Since our model focuses on a multi-client environment, both these metrics are considered in terms of aggregation among all the clients. Secondly, to address for in-flight QoE estimation, we take a window-based approach. To consider the nonlinear relationship between the two metrics over multiple clients and to modestly penalize for bitrate switching, our model considers adding the average bitrates received by all the clients, which is divided by the sum of the exponential smoothing values of the bitrate switching frequency of all clients in each window adjusted by a weight. If we set Δt to be the window and the number of active clients to be C , then our MQoE_RF model can be written as (see Table 1 for the complete list of notations):

$$\text{MQoE_RF}_{C,\Delta t} = \frac{\frac{1}{C} \left(\sum_{c=1}^C \overline{B_{c,\Delta t}} \right)}{1 + \frac{\frac{1}{C} \left(\sum_{c=1}^C \delta_{c,\Delta t} \right)}{\gamma}}. \quad (1)$$

Table 1 Notations used in various models

Variable	Description
B	Bitrate
\bar{B}	Average bitrate
K	Number of segments
C	Number of active clients
Δt	Time window
δ	Exponential smoothing of the switching frequency
N	Number of switching frequencies
σ	Standard deviation
ν	Exponential smoothing weight ($0 \leq \nu \leq 1$)
α, β, γ	Weight parameters

Here, $\overline{B_{c,\Delta t}}$ represents the average bitrate for client c during the window Δt , and $\delta_{c,\Delta t}$ represents exponential moving average on bitrate switching frequency, given by

$$\delta_{c,\Delta t} = (1 - \nu) * \delta_{c,\Delta t-1} + \nu N_{c,\Delta t}. \quad (2)$$

We now explain further the rationale behind our choices. Our assumption is that window Δt is a reasonable time window for measurements (see Section 4 for further discussion). Then, $\overline{B_{c,\Delta t}}$ represents the average of all the DASH segments for client c in Δt . The switching frequency factor in the denominator is exponentially smoothed to even out any large oscillatory behavior during window Δt . The parameter γ acts as a scaling (damping) parameter on this bitrate switching frequency. Finally, to account for the possibility of no switching frequency, especially over multiple windows, that could lead to $\delta_{c,\Delta t}$ being nearly zero, we have added one in the denominator in (1) as the final stabilization factor in our MQoE_RF model. In this case, the QoE model can still be computed based on the average bitrate.

Our second moving QoE model, MQoE_SD, differs from MQoE_RF (1) in that it is a linear model that relates the bitrate magnitude with the change arising in bitrates. That is, the first term in MQoE_SD is the same as the numerator in MQoE_RF, which reflects the average of bitrates among all the clients. The standard deviation of the bitrates of all the clients in a window can be an aggregated value to capture the switching magnitude during the window; this term is then subtracted from the bitrate term based on a weight. Thus, our second model, MQoE_SD, can be written as:

$$\text{MQoE_SD}_{C,\Delta t} = \frac{1}{C} \left(\sum_{c=1}^C \overline{B_{c,\Delta t}} \right) - \alpha \cdot \frac{1}{C} \left(\sum_{c=1}^C \sigma(B_{c,\Delta t}) \right), \quad (3)$$

where $\sigma(B_{c,\Delta t})$ represents the standard deviation on bitrates in window Δt for client c .

To contrast our moving QoE models, consider next a static QoE model such as the original MPC QoE model [5]:

$$\text{QoE_MPC} = \sum_{k=1}^K B_k - \beta \cdot \sum_{k=1}^{K-1} |B_k - B_{k-1}|, \quad (4)$$

where B_k is the bitrate of segment k while K is the total number of segments in a video. Note that such static QoE models consider all segments being delivered in their QoE calculation. Before we discuss how a static model could be adapted for moving QoE determination, we point out that we kept the two most dominant terms from [5]: bitrates of segments and differences in bitrate from one segment to the next from the original MPC QoE model.

We do not consider two terms from the original MPC QoE model: the startup delay and rebuffering. The startup delay is an issue only at the beginning of a video session. Since we are considering a windowed scenario for moving QoE, this term is relevant only at the beginning in the initial window, but no longer in any other windows and is, thus, irrelevant for moving QoE and is ignored. The other term, rebuffering, was found to have very minimal effects in a recent extensive study with video streaming [10]; thus, we also do not consider this term (we will comment on rebuffering later in Section 5.3 based on our study and point out how rebuffering is indirectly captured by our MQoE models). Since (4) for observing QoE is for an entire video, we adapt it for use in each window by computing for the segments transferred in each window. Assume that in the window Δt , the number of segments is $K_{\Delta t}$, the MPC QoE model in the window Δt for client c can be rewritten as:

$$\text{QoE_MPC}_{c,\Delta t} = \sum_{k=1}^{K_{\Delta t}} B_k^c - \beta \cdot \sum_{k=1}^{K_{\Delta t}-1} |B_k^c - B_{k-1}^c|. \quad (5)$$

Thus, over the set of all clients in window Δt , we get the following moving MPC-based QoE model, which we refer to as MQoE_MO:

$$\text{MQoE_MO}_{C,\Delta t} = \frac{1}{C} \sum_{c=1}^C \left(\sum_{k=1}^{K_{\Delta t}} B_k^c - \beta \cdot \sum_{k=1}^{K_{\Delta t}-1} |B_k^c - B_{k-1}^c| \right). \quad (6)$$

3 Study environment

To study the MQoE models, we implemented our study environment on the GENI testbed [8], in which clients access a video from a DASH video server. In this environment, we allow multiple clients to access simultaneously the same video to emulate watching the same live event by a number of users. The raw link bandwidth was set to 10 Mbps. The DASH client code was implemented in Python and was first used in [11]; the video server was based on Apache HTTP server. For the ABR scheme used by the clients, we used the commonly used hybrid ABR algorithm that applies both throughput and buffer signal for bitrate selection based on [12].

For our study, we used *Big Buck Bunny* (BBB) and *Elephants Dream* (ED) [13], two well-known DASH video datasets, which consist of 150 and 164 segments, respectively. Each of these videos has 20 bitrate representations, ranging from 0.045 Mbps at the lowest resolution for both of the datasets to 3.936 Mbps and 4.066 Mbps at the highest resolution for *Big Buck Bunny* and *Elephants Dream*, respectively. The twenty representations and the gaps between the representations sequence in both datasets are very similar. Each segment was of 4 sec playback duration. Thus, the entire *Big Buck Bunny* video is 10 min long and the entire *Elephants Dream* video is 11 min. While the two datasets have very similar bitrate representation, the sizes in terms of bytes of each segment in a dataset with a specific representation is not the same as the other dataset. In Fig. 1, we show the segment sizes (in MB) of the highest bitrate codec representation for each dataset to illustrate this point. For example, for first few segments of *Big Buck Bunny* with highest bitrate of 3.936 Mbps has

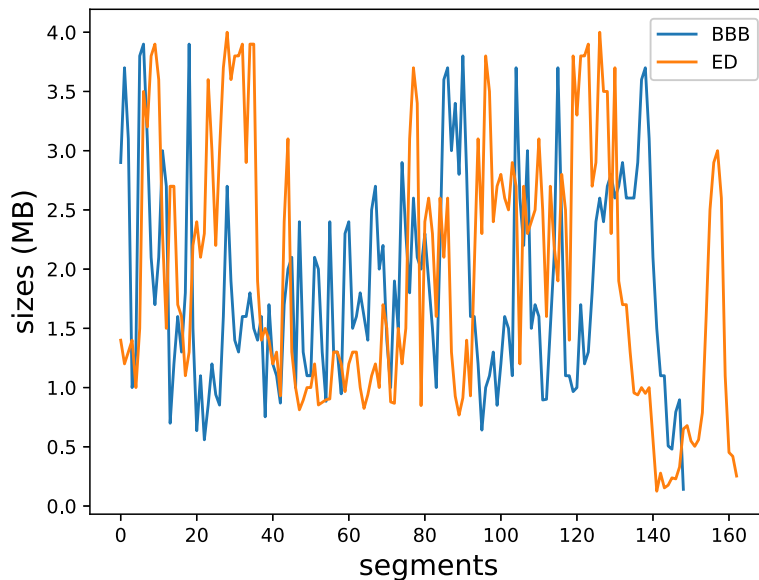


Fig. 1 Segments in megabytes: BBB vs. ED for the highest representation

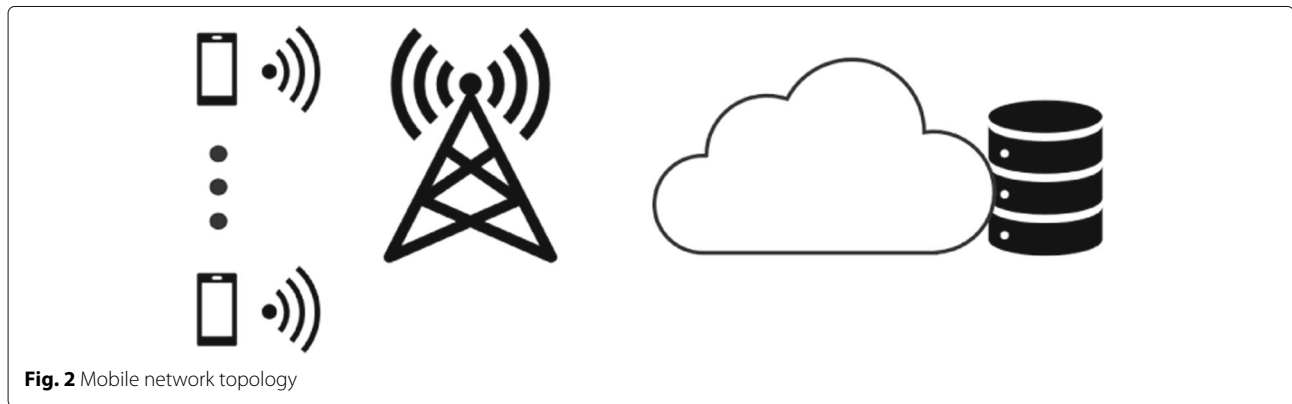


Fig. 2 Mobile network topology

a significant larger size than the first few segments of *Elephants Dream* highest bitrate of 4.066 Mbps. We observe similar differences in the other segments multiple times. The main implication of this observation is that even if each segment received is from the highest representation, the bitrates are not related to the segment sizes in bytes. Consequently, MQoE observed for user watching *Big Buck Bunny* would be different than for *Elephants Dream* since the ABR algorithm depends on the throughput as a factor, assuming all other factors being equal.

To emulate the effect on the clients to experience mobility while traveling and connected to a wireless network environment as shown in Fig. 2, we used traces on path behavior dataset provided by Riser et al. [14] while traveling by a car, a train, or a ferry. We installed Wondershaper [15] on the server machine to throttle the link with these traffic traces. As noted in [14], with a car, there are frequent fluctuations on bandwidth availability and also dead time when the signal is not available. On the other hand, due to length of the videos we studied, our study faces the frequent fluctuation part, but never reached the dead time mentioned in [14]. With the train trace, there is a large drop in bandwidth availability at one point during our study, which captures essentially the dead time. Finally, for the ferry trace, the ferry was going from one shore to another shore; thus, the bandwidth availability continued to gradually drop as the ferry moved away from the departing shore. Due to the duration of the videos we studied, the bandwidth availability was just about to start to ramp up due to signal strength improving from the other shore when our videos ended. Thus, the three traces gave us different perspectives on mobility patterns during the video streaming period and it is instructive to keep this in mind. We conducted our study in this setting with multiple clients ranging from three clients to ten clients. Finally, MQoE models for multiple clients are presented by normalizing based on the maximum value of the MQoE model for a single client for the associated traffic trace.

4 Initial analysis and adjustment

Our initial analysis centered around determining window Δt , weights associated with our QoE models, and the limitation of MQoE_MO for use in moving QoE monitoring.

Our first experimentation was to determine the window duration Δt by varying the window size. We observed that with a small window size, there are many windows that have no switching. Sometimes with a higher traffic that causes latency in the client's request, no segment might be transmitted if it is a very tiny window. Based on our initial trials, the window size of 60 sec was found to be a good window size that has a reasonable number of segments and switching in each window. The window size was kept at a constant time duration for the rest of our study. Then, in all our study, we consider the first window for Δt to be the ramp-up window; thus, we will focus on results from window 2 to window 10 for both the videos. In addition, we can observe ramp-up of the frequency in the second window as we used exponential smoothing with previous value from the first window.

For use in (1), exponential smoothing in switching frequency was used as shown in (2). We found that setting $\nu = 0.75$, which gives more weight to the newest value of switching frequency, allows us a level of relative stability while capturing the changes. Thus, this value was used in the rest of our study.

We considered a number of different values for both γ and α used in (1) and (3), respectively. The combination of γ and α are categorized into three sets: set0: $\gamma = 25$ and $\alpha = 0.5$; set1: $\gamma = 10$ and $\alpha = 1$; set2: $\gamma = 5$ and $\alpha = 1.5$. This is summarized in Table 2. We found

Table 2 Sets of weight parameters for MQoE models

Sets	γ	α
Set0	25	0.5
Set1	10	1.0
Set2	5	1.5

that when $\gamma = 25$ (from set0), the MQoE_RF behavior is similar to the bitrates received by the clients; i.e., the bitrate term in the numerator is the dominant term in (1) for MQoE_RF at this value of γ . As we reduced γ from 25 to 5, we noticed that the MQoE behavior changes to the point of being more fluctuating, while giving higher weights to the denominator in (1); Thus, we show the graphs for three values of γ at 25, 10, and 5 in Fig. 3a with three clients in car. Similarly, we found that when $\alpha = 0.5$ (from set0), then the MQoE_SD behavior in (3) is similar to the bitrates received by the clients. When α increases to 1.5, the MQoE behavior changes to the point where QoE has higher fluctuation because it is giving more weights to the second term in (3). Thus, for α , three values 0.5, 1.0, 1.5 are shown in Fig. 3b for three clients in car. From this discussion, it is clear that set0, i.e., $\alpha = 0.5$ and $\gamma = 25$ essentially reflects the same behavior as bitrates. Thus, we exclude set0 in the rest of the discussions in this paper.

The parameters, γ and α , are self-learned parameters. We note that γ between 5 and 25 and α between 0.5 and 1.5 are the useful ranges for these parameters to account for the associated term. Below or above these ranges, we observe asymptotic behavior that would not give us any new information.

Consider next the MQoE_MO model (6). We found that the straightforward extension of the MPC QoE model to moving QoE model (6) is problematic at times. Consider again Fig. 4, which also includes MQoE_MO on the graph. From window 8 to 10, the MQoE_MO value dropped by 57.0% while the bitrate drop was only 20.03% with set1 (see Fig. 5a). On further investigation, we found that since the number of segments transmitted during a window can vary (depending on the network condition), it may also be possible that this number can be quite low as it so happened in window 10. With (6), a large drop is possible in the moving QoE in a particular window. This result also illustrates that a static QoE model is not readily usable as a moving QoE model. Thus, for moving QoE, new models

as we proposed here are necessary. In the rest of the paper, we simply focus on our MQoE models (1) and (3), and for two sets of α and γ parameters values: set1 and set2.

5 Comparative study

By considering three mobility traces based on car, train and ferry, our study focuses on three dimensions: 1) to establish the behavior of the MQoE models as the number of clients is varied, 2) to understand how the models are impacted as we consider two different videos, and 3) perceived QoE by clients in a multi-client scenario. As noted earlier, the environment for this study mimics as if the clients are watching a live event.

Of the two videos, most of our discussions from our study centers around scenarios for the *Big Buck Bunny* video. We also discuss results for *Elephants Dream*, in certain cases with cars, to show the difference in behavior that is a manifestation of the difference between the two videos in terms of bitrates and segment sizes that we discussed earlier in Section 3.

For each scenario, we show a set of figures that represents the average bitrate, the bitrate exponential switching frequency, the standard deviation of the bitrate switching magnitude, and the MQoE values for both our models as Δt changes. For some of the scenarios, we present the mean segment size of each window along with maximum and minimum values of each window during the session for all the clients. This will show the main reason of difference in bitrate representation selection for the two datasets and finally the MQoE values while having similar configurations. Then we provide segment-based size and bitrate comparison for the first client of a scenario for the two datasets.

Next the results are discussed for each of the three mobility trace scenarios: car, train, and ferry, with more detailed discussions for 'car'. For all trace scenarios, we studied three situations in terms of the number of simultaneous clients: three, five and ten clients.

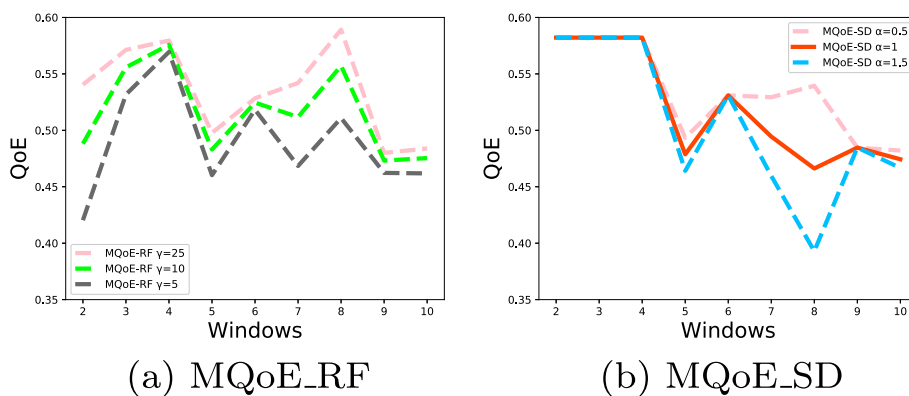
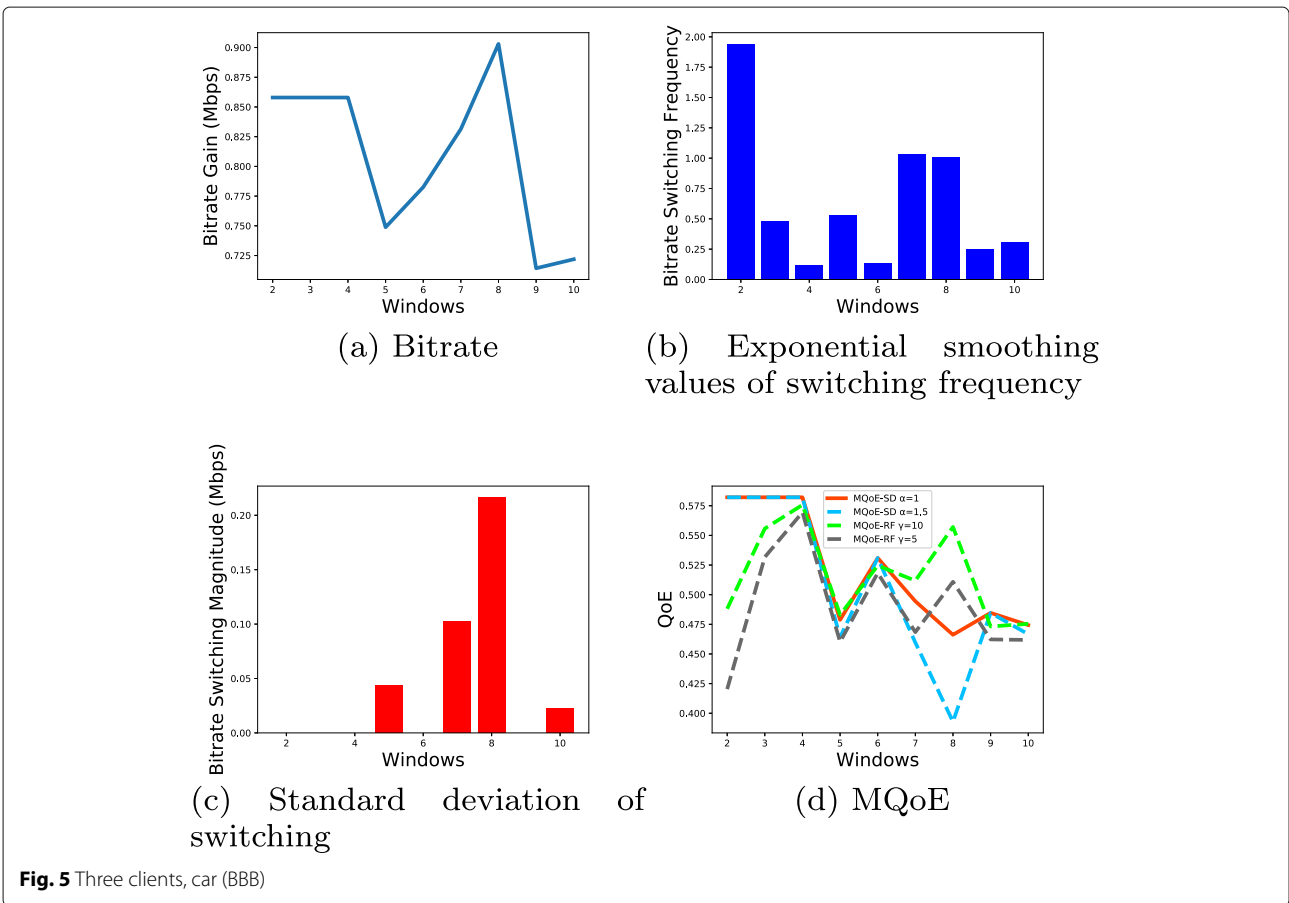
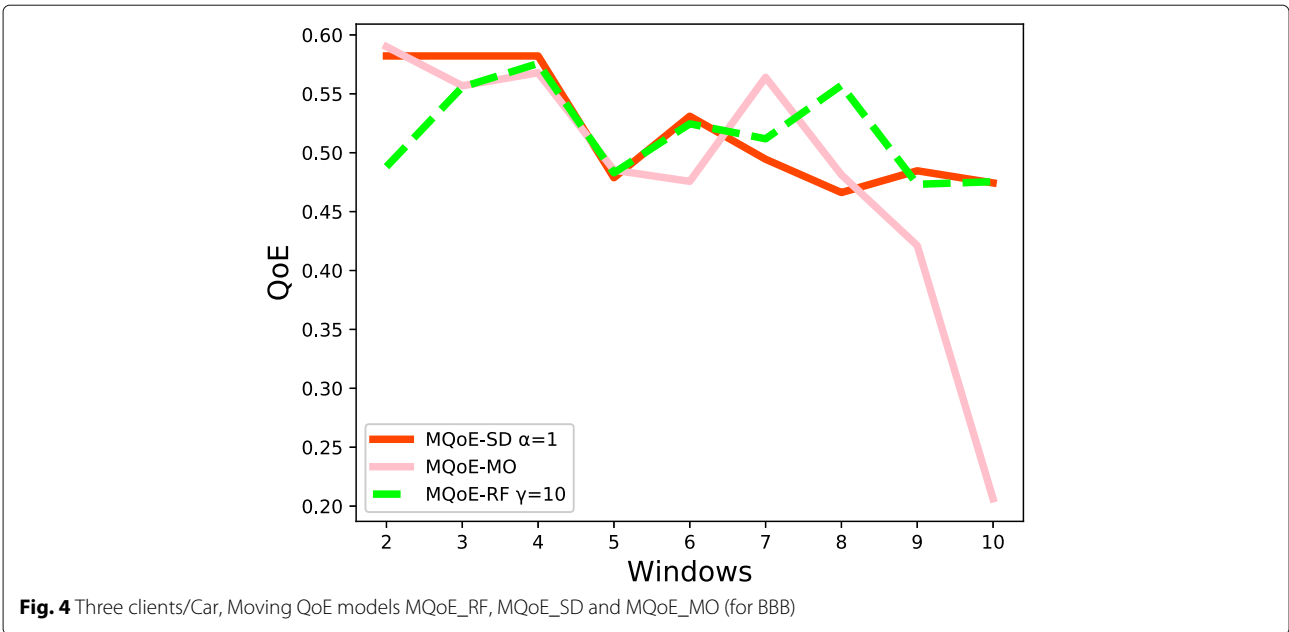
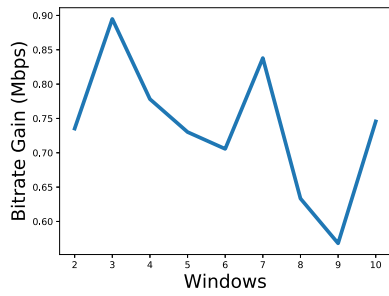
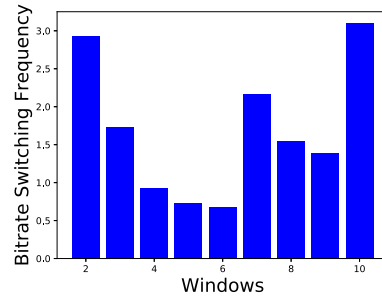


Fig. 3 Three clients (car) with different values of α and γ (for BBB)

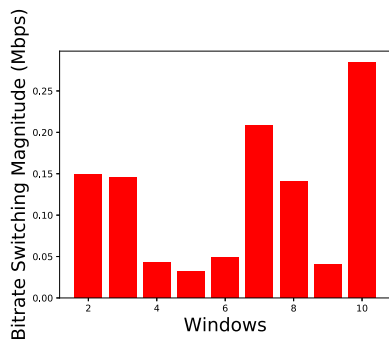




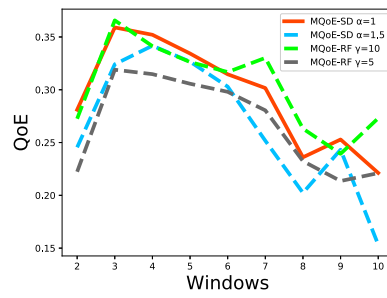
(a) Bitrate



(b) Exponential smoothing values of switching frequency



(c) Standard deviation of switching



(d) MQoE

Fig. 6 Three clients, car (ED)

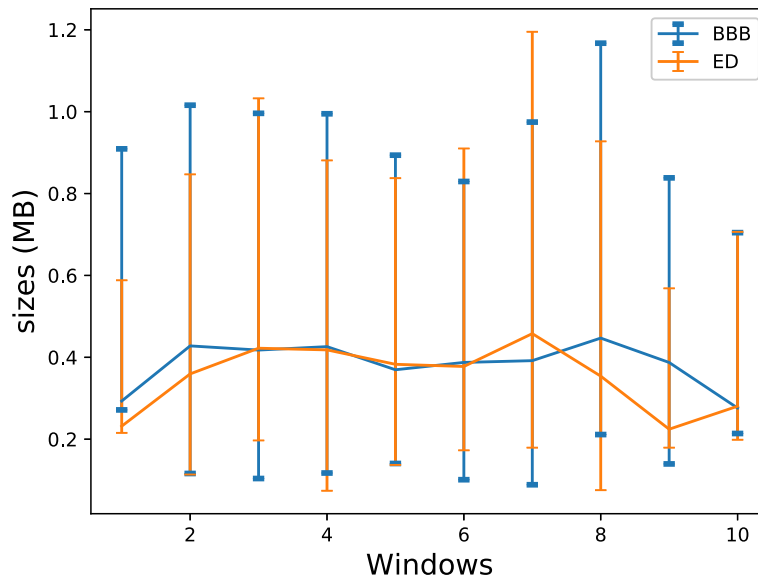


Fig. 7 Three clients (with car), bytes range in each window for BBB and ED

Table 3 Average values of segment sizes (in MB)

Scenario	Average value BBB	Average value ED
Car (3 clients)	0.38	0.35
Car (5 clients)	0.26	0.21
Car (10 clients)	0.13	0.11
Train (3 clients)	0.27	0.34
Train (5 clients)	0.17	0.19
Train (10 clients)	0.12	0.10
Ferry (3 clients)	0.33	0.34
Ferry (5 clients)	0.24	0.22
Ferry (10 clients)	0.13	0.11

5.1 Mobility trace: car

5.1.1 Three clients

Consider first three simultaneous clients watching when the *Big Buck Bunny* video is streamed. From Fig. 5a, we see that as we go from window 6 to window 7, the bitrate increases. Along with that, the bitrate switching frequency and bitrate switching magnitude also increase (see Fig. 5b and c), which affect the QoE models (see Fig. 5d) in different ways. With set1, the bitrate switching frequency and bitrate switching magnitude impact on both models which cause QoE value drop by 2.43% for MQoE_RF and with a larger drop by 6.88% for MQoE_SD. The impact of the magnitude on the linear model is larger than the impact of frequency on the non-linear model for window 7. These changes become more severe when the value of α

increase and γ decrease where MQoE_RF and MQoE_SD drop by 9.6% and 13.44% for set2, respectively.

From window 7 to window 8 MQoE_RF with either value of γ shows an increasing trend when the bitrate reaches the peak value. It means that for that value of bitrate, the bitrate switching frequency cannot be much significant for the model to take a different trend than bitrate. The value of bitrate switching frequency Fig. 5b did not change notably. However, MQoE_SD decreases and with larger values of α from set2, a larger drop occurs, which is caused by the high value of the bitrate switching magnitude with a higher weight in (3).

Note that in the plots on the standard deviation of the bitrate switching magnitude (see Fig. 5c and also in later figures), some windows have values equal to zero. This happens when in a specific window, no bitrate switching occurs. This may also happen if the latency due to congestion on a link is too high that the number of segments in these windows is one or less.

Now if we look at the results for the *Elephants Dream* video (Fig. 6), the metrics' value and behavior on each window were found to be different than with the *Big Buck Bunny* video. Bitrate (see Fig. 6a), unlike the *Big Buck Bunny* (see Fig. 5a), in the first three windows has an increasing trend as the standard deviation on bitrate switching shows some values on those windows. From window 8 to window 9, while the bitrate decreases, MQoE_SD shows an increasing trend when the standard deviation for bitrate switching decreases. MQoE_RF shows a closer trend to the bitrate as the bitrate

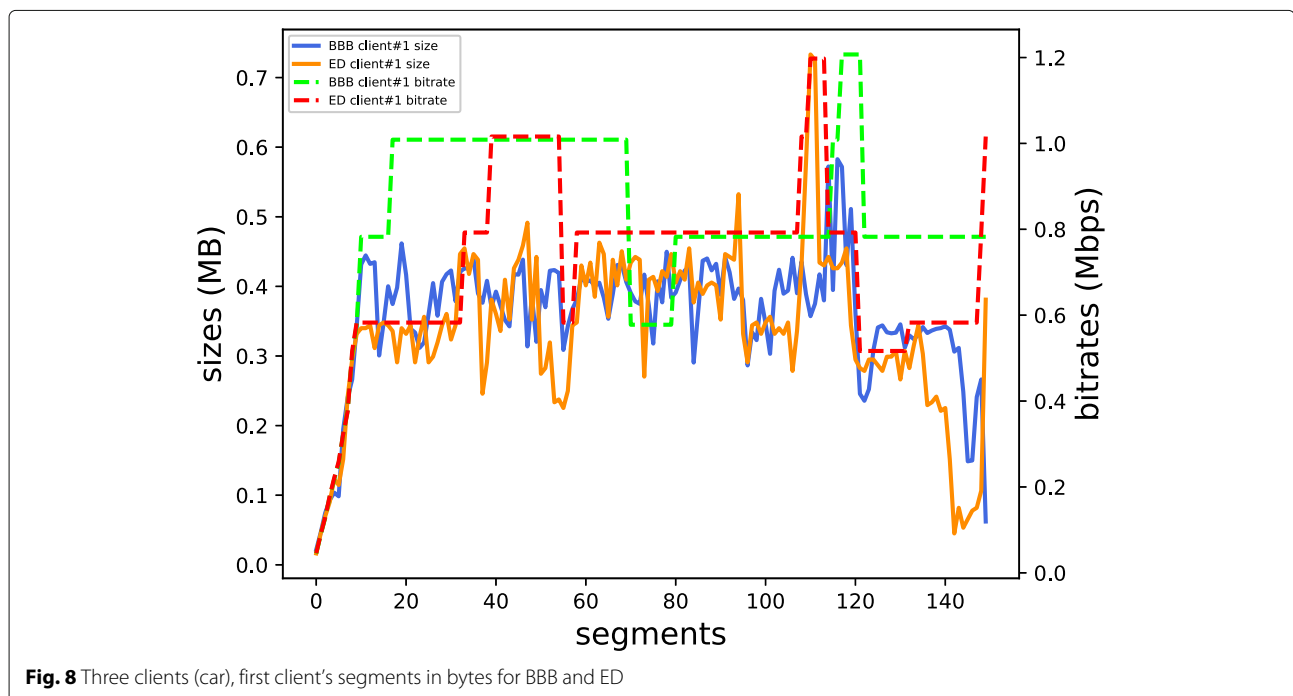


Fig. 8 Three clients (car), first client's segments in bytes for BBB and ED

switching frequency has almost the same value for these two windows. From window 9 to window 10, the bitrate has an increasing trend; however, MQoE_SD decreases due to the increase in bitrate switching standard deviation by almost six times. The MQoE_RF still shows rise for both sets, although the jump is lower than bitrate jump, as the bitrate switching frequency has been doubled.

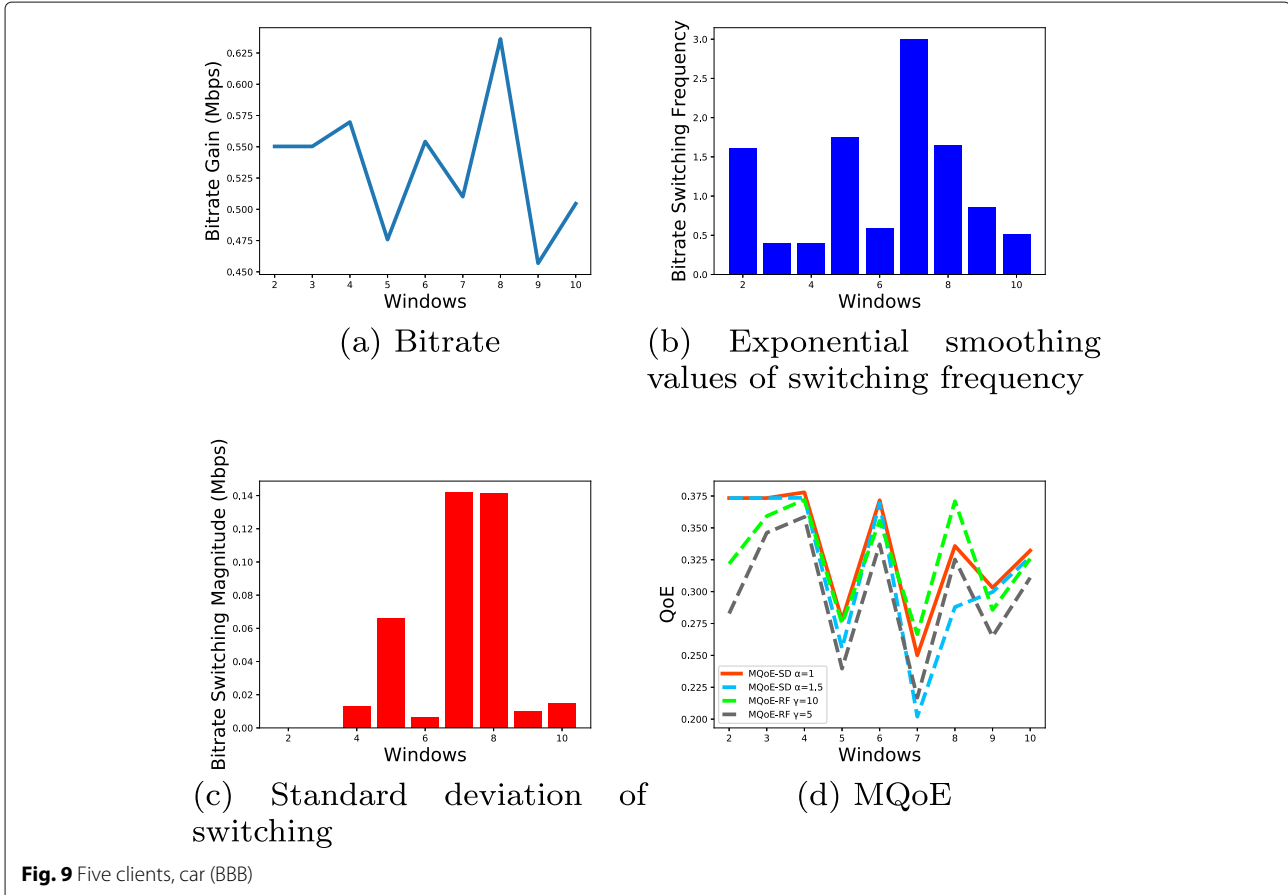
We next take a comparative view of the observations from the two different videos. As we mentioned in the previous section, the bitrate codecs of the two videos are similar; however, segments at the same level of the bitrate codec vary in sizes (in terms of MB). In other words, similar codecs do not mean similar sizes in bytes. The average, maximum and minimum on the sizes in MB for segments transmitted in each window for all clients are shown in Fig. 7. For *Big Buck Bunny*, the average sizes (in MB) from window 3 to window 6 are close to *Elephants Dream*. On the other hand, from the window 1 to window 3 and from the window 7 to window 10, the average sizes for *Big Buck Bunny* is higher than *Elephants Dream*. Table 3 shows the average values for all studied cases. This presents the average size of the 10 window for each dataset wherein *Big Buck Bunny* is found to be greater than that for *Elephants Dreams* in this for three clients (in car). It is

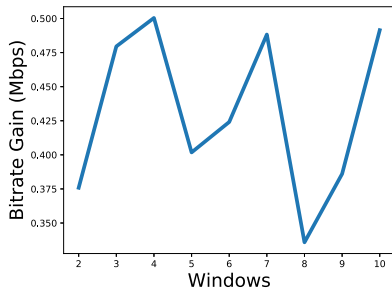
instructive to compare bitrates for each video, shown here for the first client along with segment byte sizes chosen; see Fig. 8. This is to illustrate that the measured MQoE for each video can be noticeably different, which is possible since the ABR algorithm uses throughput (that depends on the bytes transferred) as a factor in deciding the bitrate to choose for the next segment.

5.1.2 Five clients

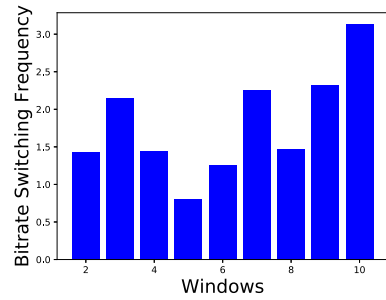
Going from three clients to five clients (see Fig. 9) for *Big Buck Bunny*, we see that bitrates have more swings (compare Figs. 9a to 5a). Both models follow the pattern of bitrate changes with set1. Same trend can be observed for set2 with MQoE_RF (where $\gamma = 5$), but from window 8 to 9 this trend changes temporally for MQoE_SD (where $\alpha = 1.5$). The MQoE_RF has higher rise compared to MQoE_SD for both sets of weight parameters as the effect of bitrate switching frequency is smaller than the bitrate switching magnitude. With a larger α , we see a shorter rise for MQoE_SD as the effect of bitrate switching magnitude is too high that MQoE_SD does not show a similar trend as bitrate for these windows.

We note that the bitrate value on the window 8 spikes for *Big Buck Bunny* (see Fig. 9a) while for *Elephants Dreams*, it

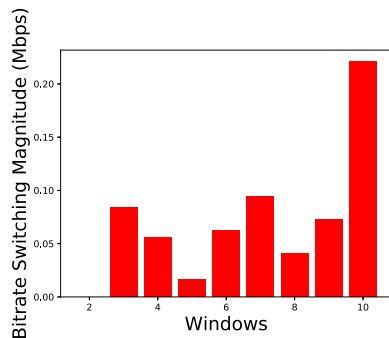




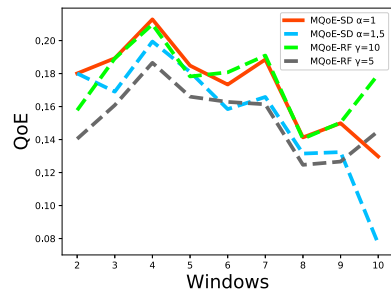
(a) Bitrate



(b) Exponential smoothing values of switching frequency



(c) Standard deviation of switching



(d) MQoE

Fig. 10 Five clients, car (ED)

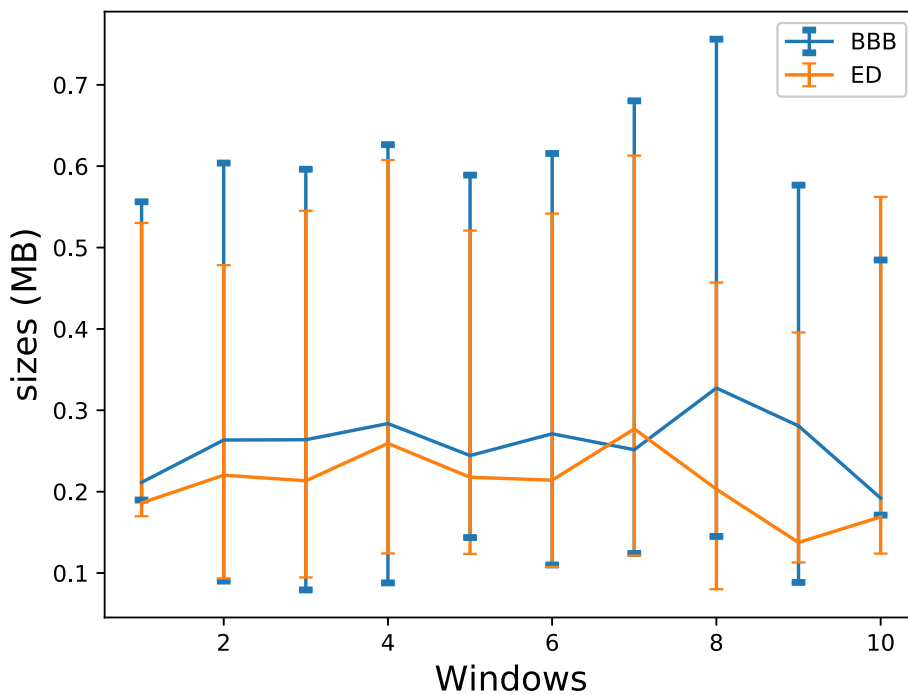
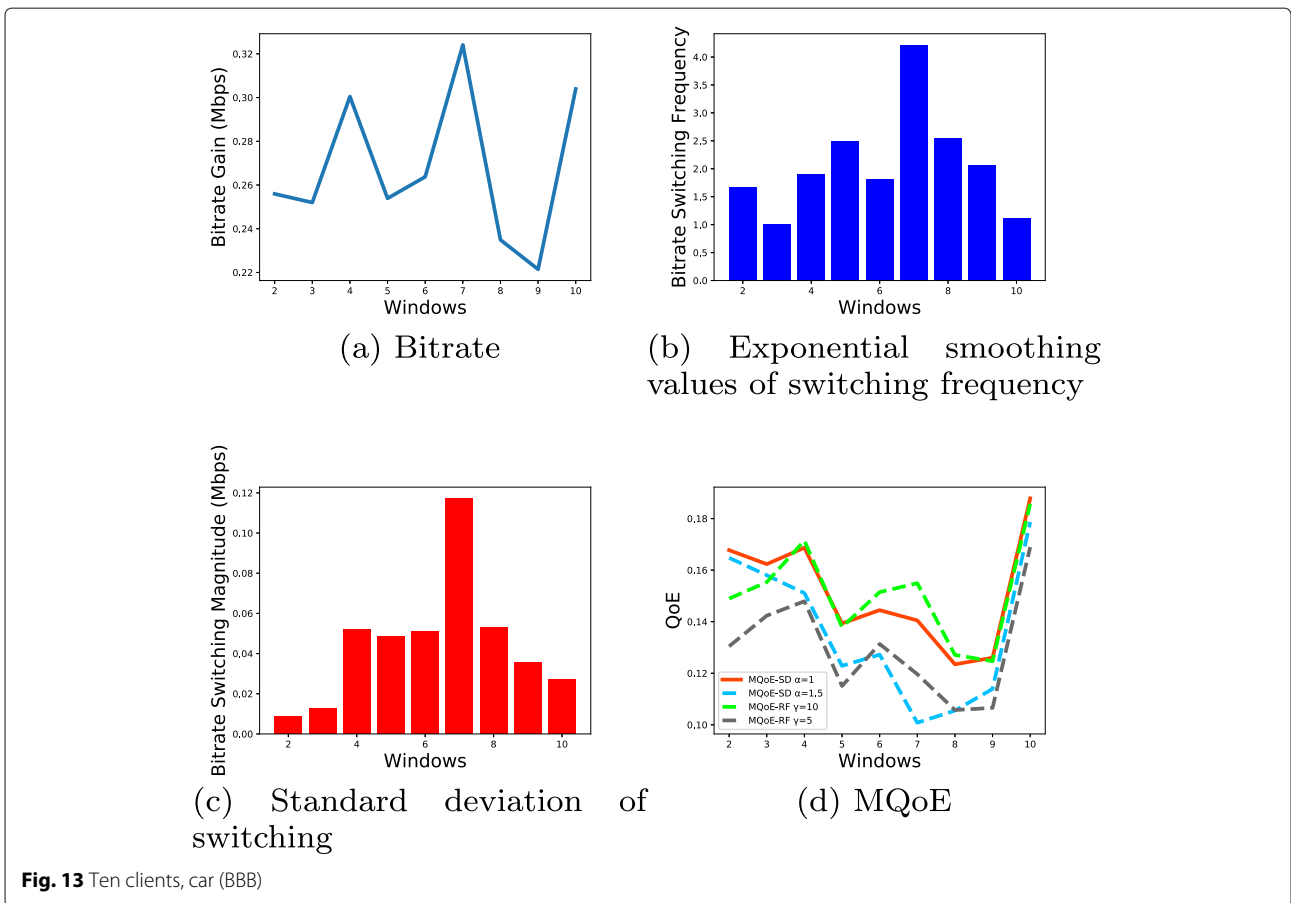
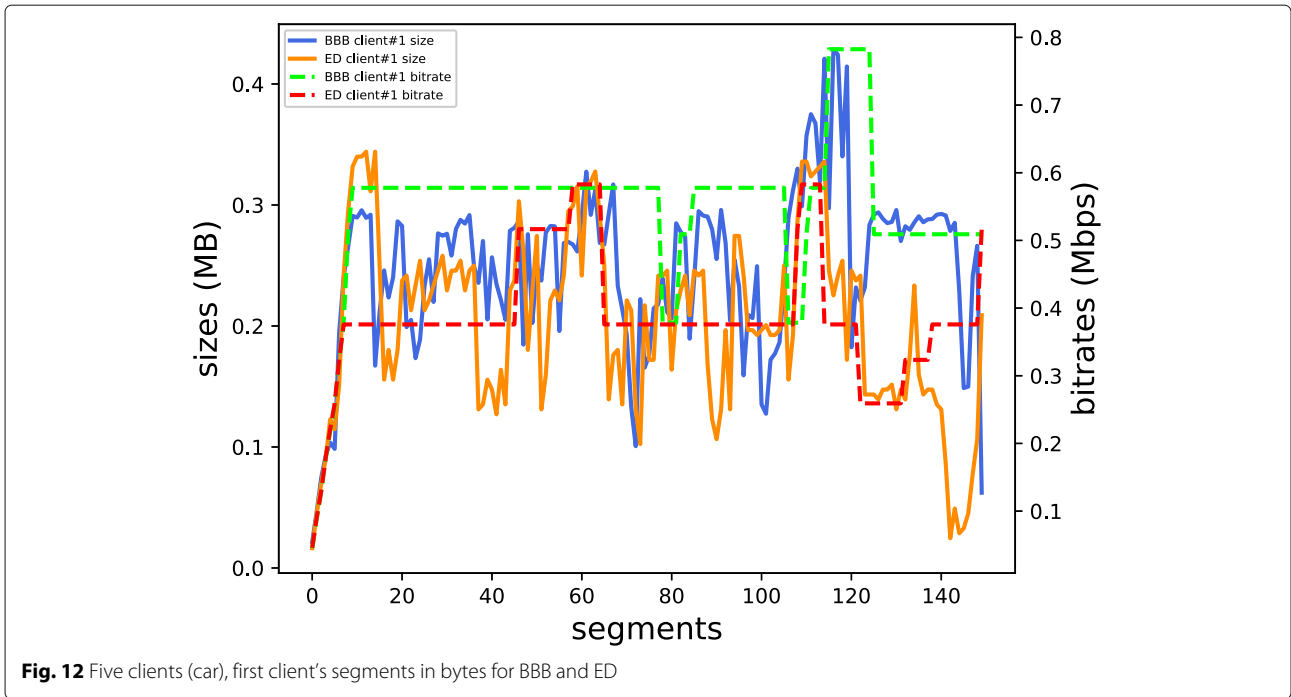
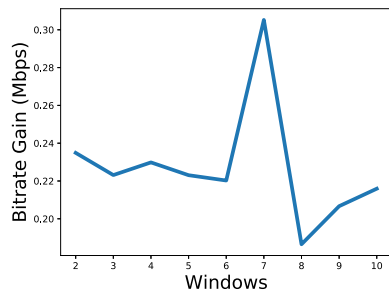
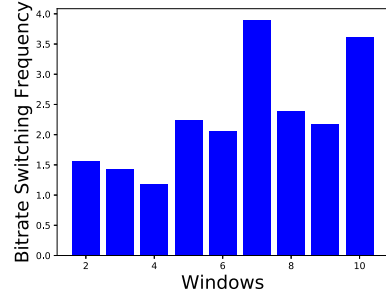


Fig. 11 Five clients (with car), bytes range in each window for BBB and ED

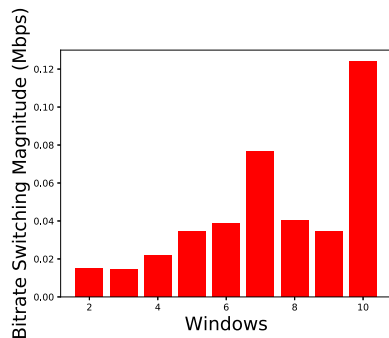




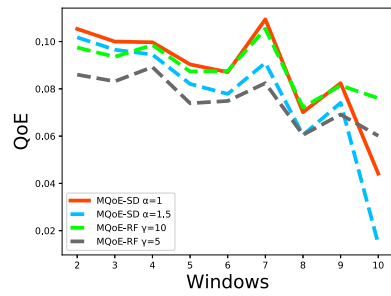
(a) Bitrate



(b) Exponential smoothing values of switching frequency



(c) Standard deviation of switching



(d) MQoE

Fig. 14 Ten clients, car (ED)

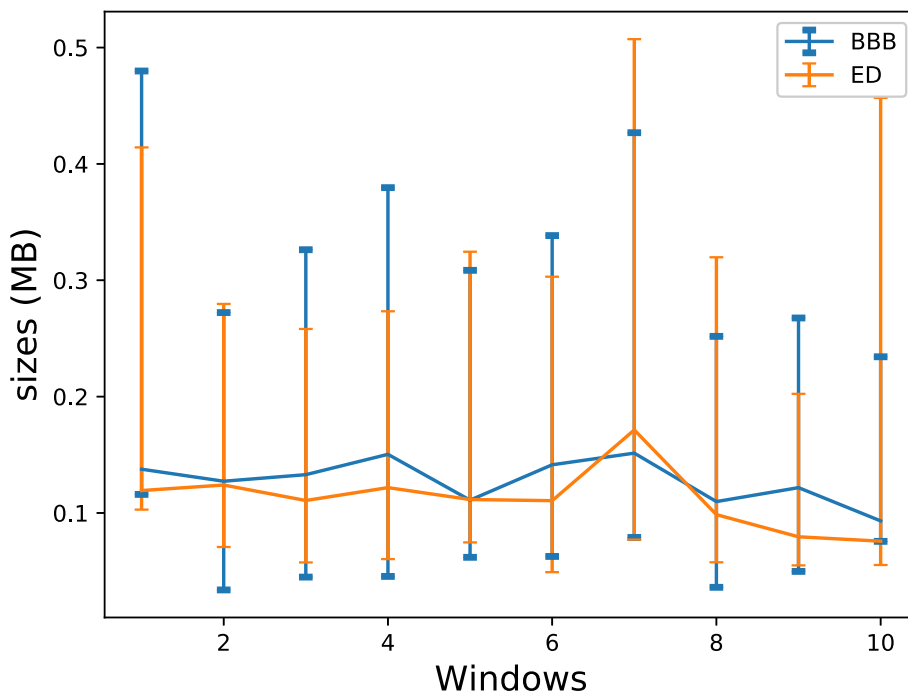
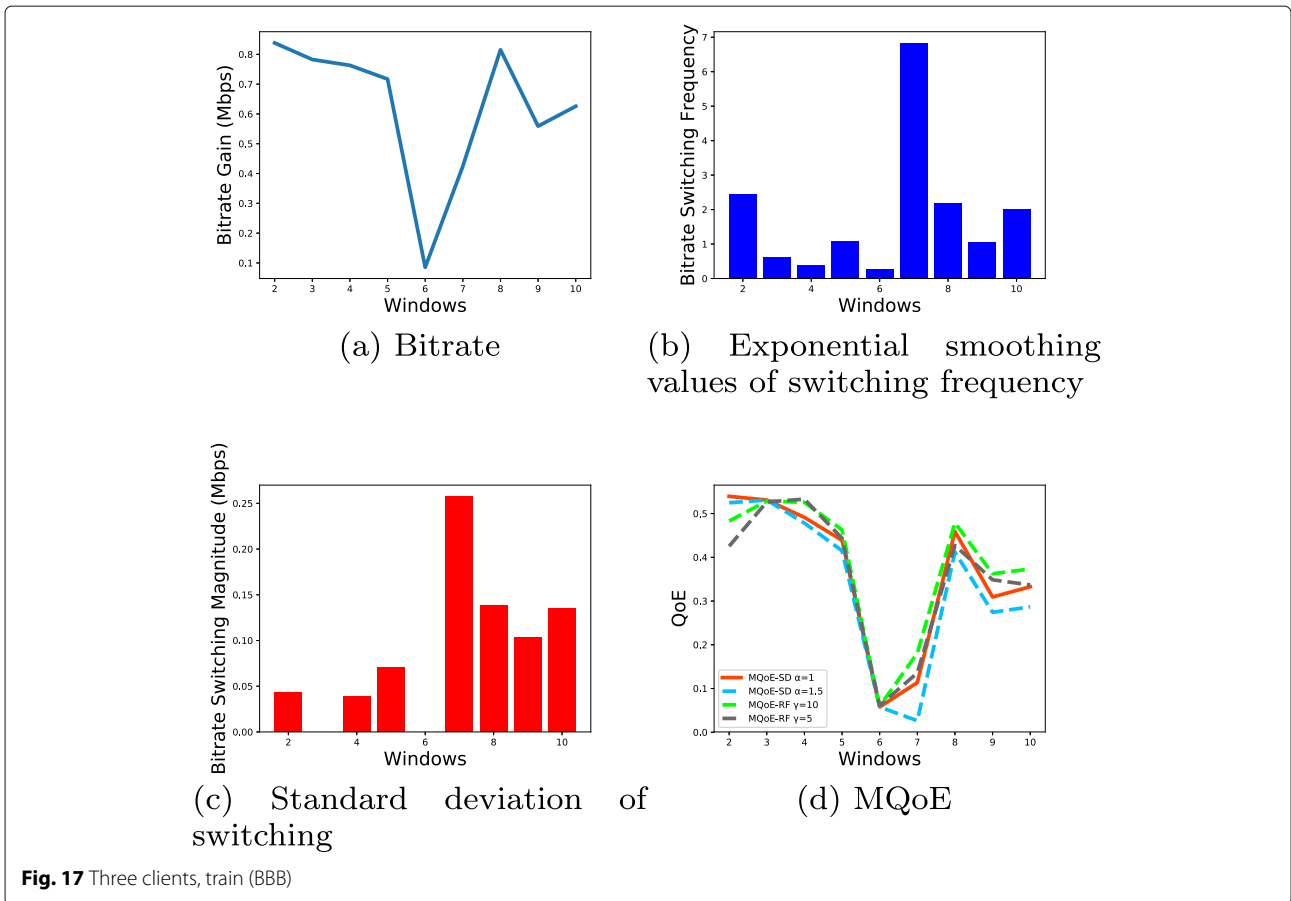
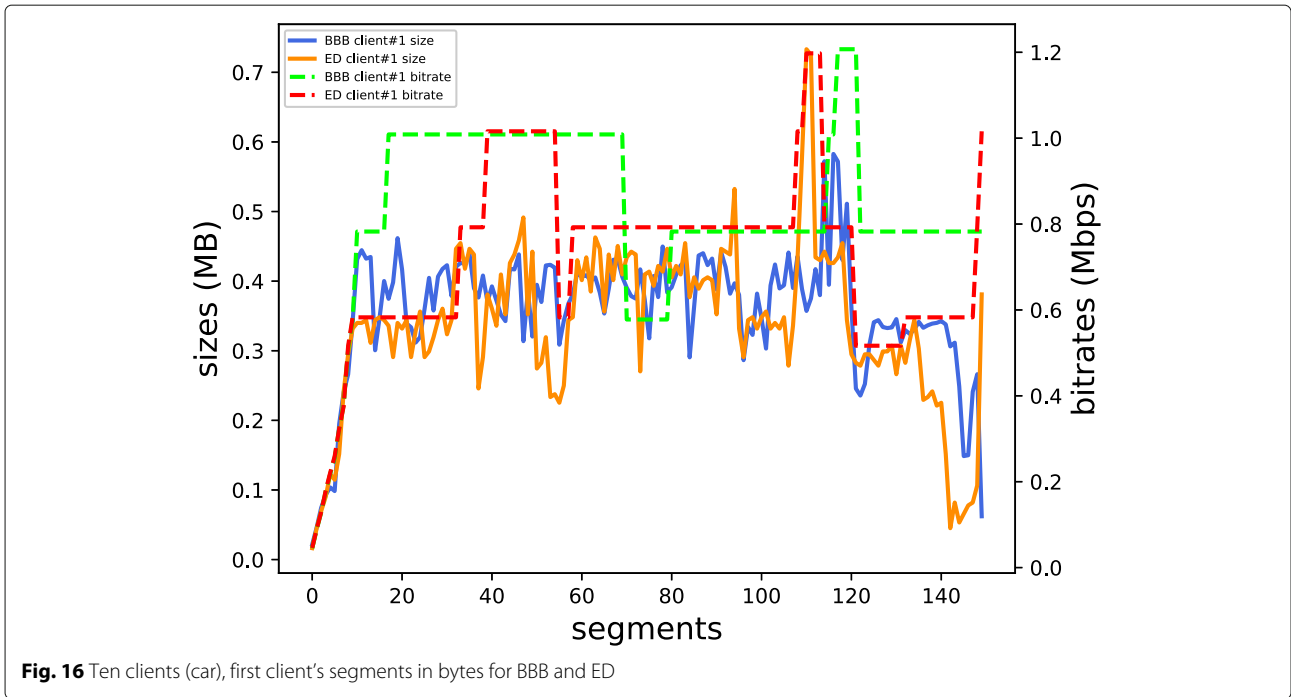


Fig. 15 Ten clients (with car), bytes range in each window for BBB and ED



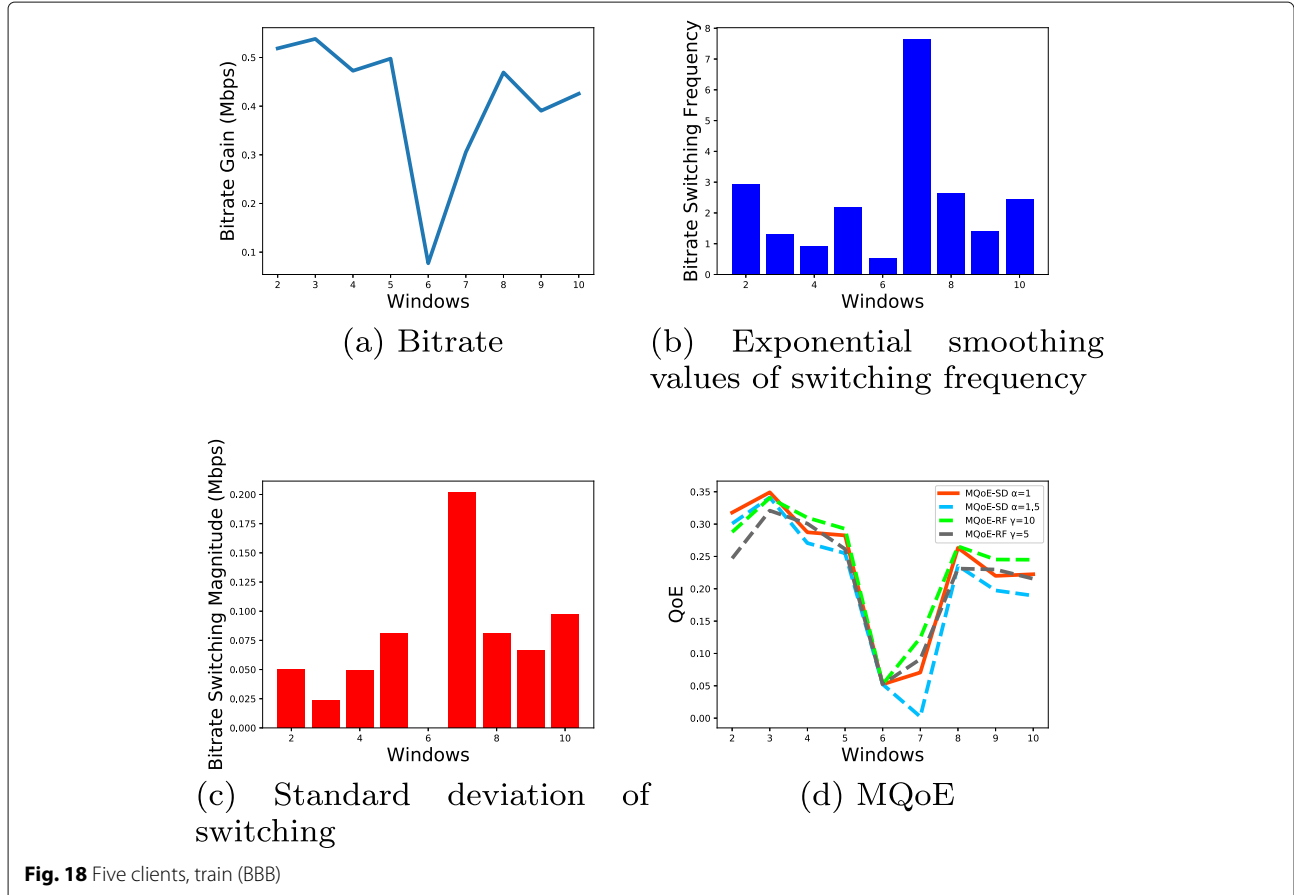
shows a drop (see Fig. 10a). From window 9 to window 10, there is a rise in the bitrate and bitrate switching frequency and magnitude; however MQoE_SD decreased with both sets of parameters. The reason is that from window 9 to window 10 there is a very large rise of bitrate switching magnitude. The QoE value (see Fig. 10d) show smaller drop and rise compared to *Big Buck Bunny* as the value of bitrate switching frequency and standard deviation for *Elephants Dream* change less for sequential windows than the same metrics in *Big Buck Bunny*. The average segment size per window for *Elephants Dream* is less than that for *Big Buck Bunny* (see Fig. 11 and Table 3). Figure 12 shows first client's segments in bytes for the two videos when there are five clients.

5.1.3 Ten clients

When we go from five clients to ten clients for *Big Buck Bunny* (see Fig. 13), the shape of the bitrate swings is notably different while the range of values are smaller due to higher number of clients. There is a 12.32% rise for MQoE_RF from window 5 to window 7 with set1. On the other hand, from window 5 to 6, the MQoE_RF with set2 and MQoE_SD with set1 and set2, there is an increase in the MQoE value by 14.17%, 3.69%, 3.55%, respectively;

then, from window 6 to window 7, these models decrease by 8.91%, 2.75%, 20.76%, respectively. The reason for the drop in MQoE_SD is that the bitrate switching magnitude is much larger in this window compared to previous windows and this value has a significant effect on the linear model (3). At window 7, the bitrate switching frequency is also large. However, its effect on the nonlinear model (1) is not as significant as the linear model (3). This is an illustration of how our QoE models are amenable to capturing the sensitivity due to frequency and the magnitude of the switching.

For *Elephants Dream* video, the bitrate has a large rise on window 7 compared to other windows. The two models (see Fig. 14d), within any set of parameters, show a very similar behavior while on window 10 they all drop while the bitrate shows a rise. This drop is moderate for MQoE_RF and severe for MQoE_SD. The increase in bitrate switching frequency on window 10 is almost twice that in the window 9 and the increase of bitrate switching standard deviation is tripled the value of window 9. Figure 15 shows that *Big Buck Bunny* has larger size segments in each window than for *Elephants Dream*. In general, the 10-client situation leads to a congested environment, and thus, the differences are minimized due to



competing for link resources by all clients. Figure 16 shows first client's segments in bytes for the two videos when there are ten clients.

5.2 Mobility trace: train

5.2.1 Three clients

We next consider our study for mobility being from a train with three clients for *Big Buck Bunny*. For this mobility scenario, a large drop in bitrate occurs in the middle due to a drop in bandwidth availability (see Fig. 17a).

More specifically, in window 6, the operable bitrate drops significantly; during this window, there is no switching to improve the bitrate. Naturally, QoE for each model also drops in this window. Going from window 6 to window 7, there is a spike in the bitrate, which increases the values for both MQoE models with set1. More specifically, we see a larger increase in MQoE_RF than MQoE_SD. When α increases with set2, MQoE_SD takes a downward trend. In window 7, both bitrate switching frequency and bitrate switching magnitude have large value. But, by increasing α for set2, the effect of bitrate switching magnitude on the MQoE_SD model is higher. MQoE_RF and MQoE_SD, were both able to capture the penalty of the bitrate switching frequency and the bitrate switching magnitude.

5.2.2 Five and ten clients

For five clients (Fig. 18 for *Big Buck Bunny*), the overall behavior is similar to that of three clients in most windows. However, with ten clients (Fig. 19), we observe a different shape than three and five clients. From window 6 to window 7, for both sets of weight parameters, there is a rise for both models along with the rise represented for the bitrate. This increase is larger for MQoE_RF.

From window 7 to window 8, the bitrate is decreasing slightly. For the model MQoE_RF with set1, we can see that the QoE value decreases while the bitrate frequency switching decreases from window 7 to window 8. The reason is that the model receives the impact of bitrate trend more than the bitrate switching frequency. However, for set2, the MQoE_RF is almost flat as the model receive more impact by bitrate switching frequency. On the other hand, MQoE_SD behaves differently than bitrate. For both sets of the weight parameters, MQoE_SD increases, unlike the bitrate and MQoE_RF. This rise means that the magnitude size is not significant compared to the previous window.

5.2.3 Comparison between two videos

For three, five and ten clients on a train watching the *Elephants Dream* video, MQoE models generally show

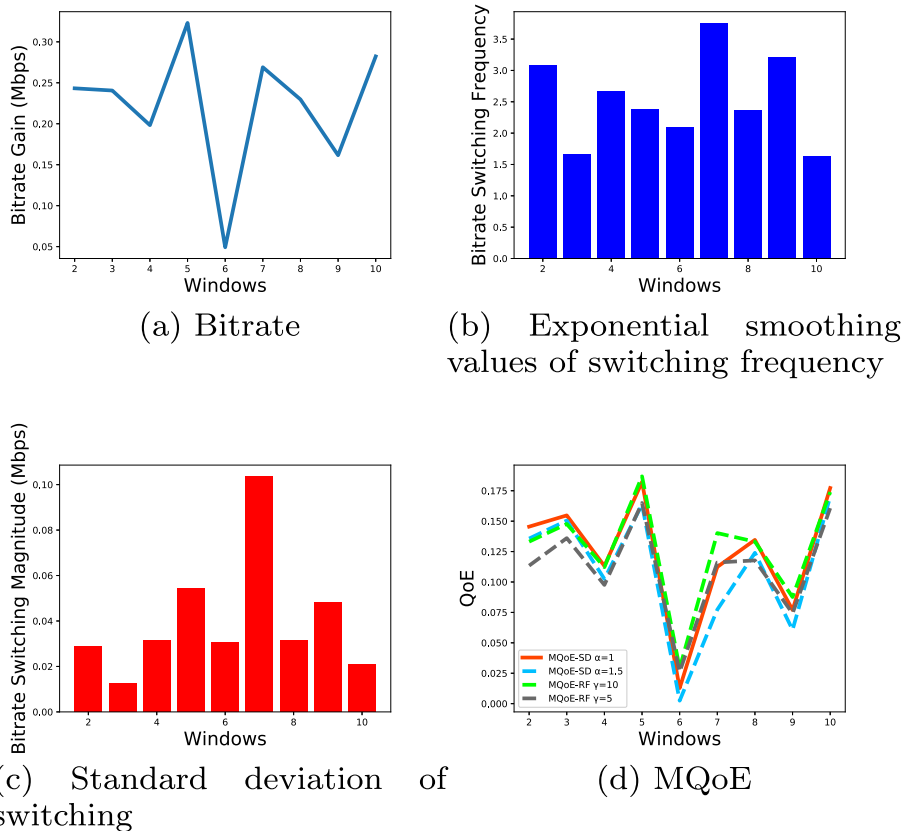
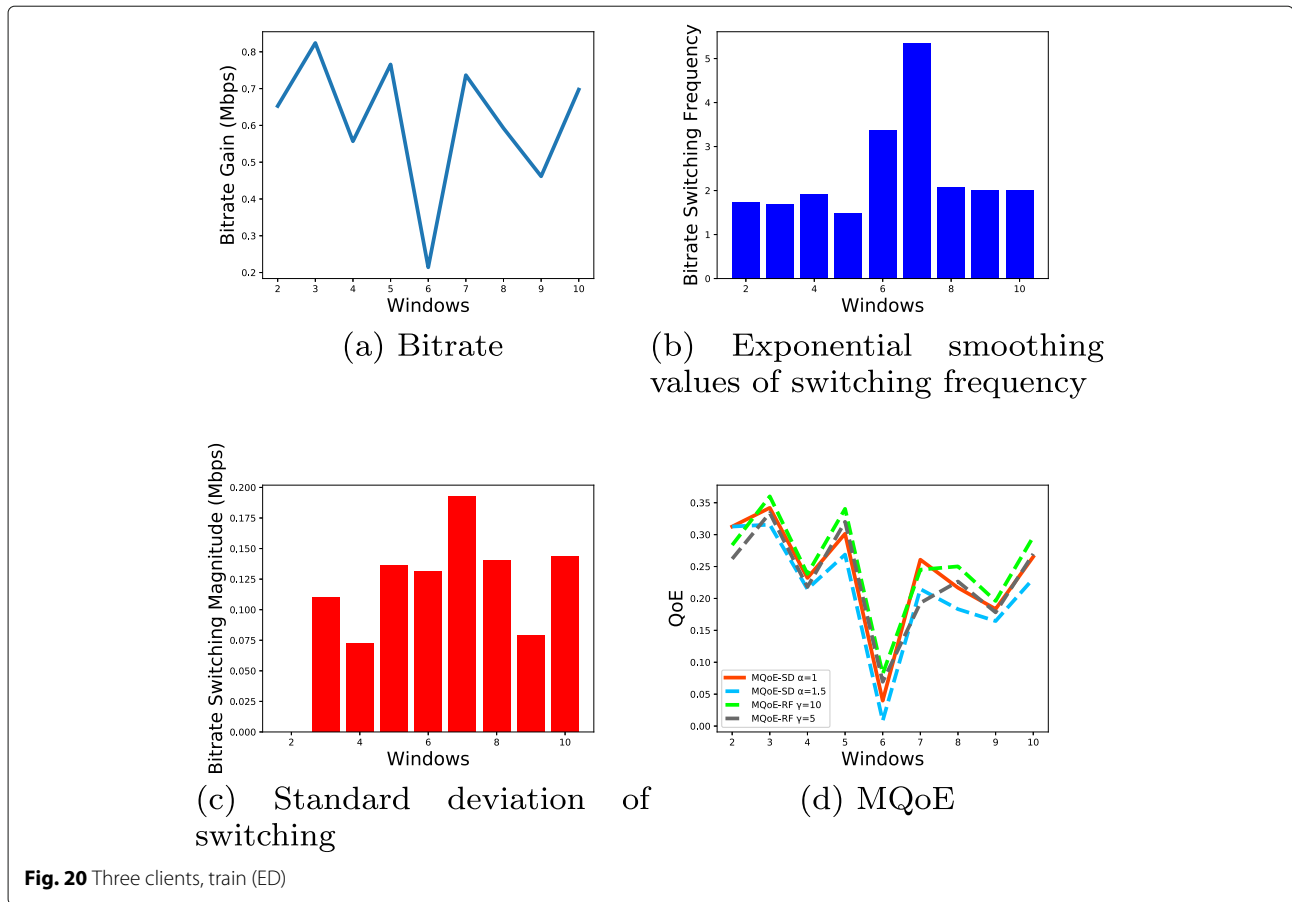


Fig. 19 Ten clients, train (BBB)



somewhat similar patterns (see Figs. 20, 21, and 22). On the other hand, for *Elephants Dream* compared to *Big Buck Bunny*, there is more fluctuation of bitrates and, thus, for MQoEs. Based on Figs. 23, 24 and Fig. 25 that show sizes on windows, for three clients train the area under the curve for *Elephants Dream* is larger than that for *Big Buck Bunny*. For five clients, the two plots of *Elephants Dream* and *Big Buck Bunny* are going through many changes, however, the overall sizes are not significantly different. For ten clients train the area under the curve for *Big Buck Bunny* is slightly larger than that for *Elephants Dream* (see Table 3 for average values).

5.3 Mobility trace: ferry

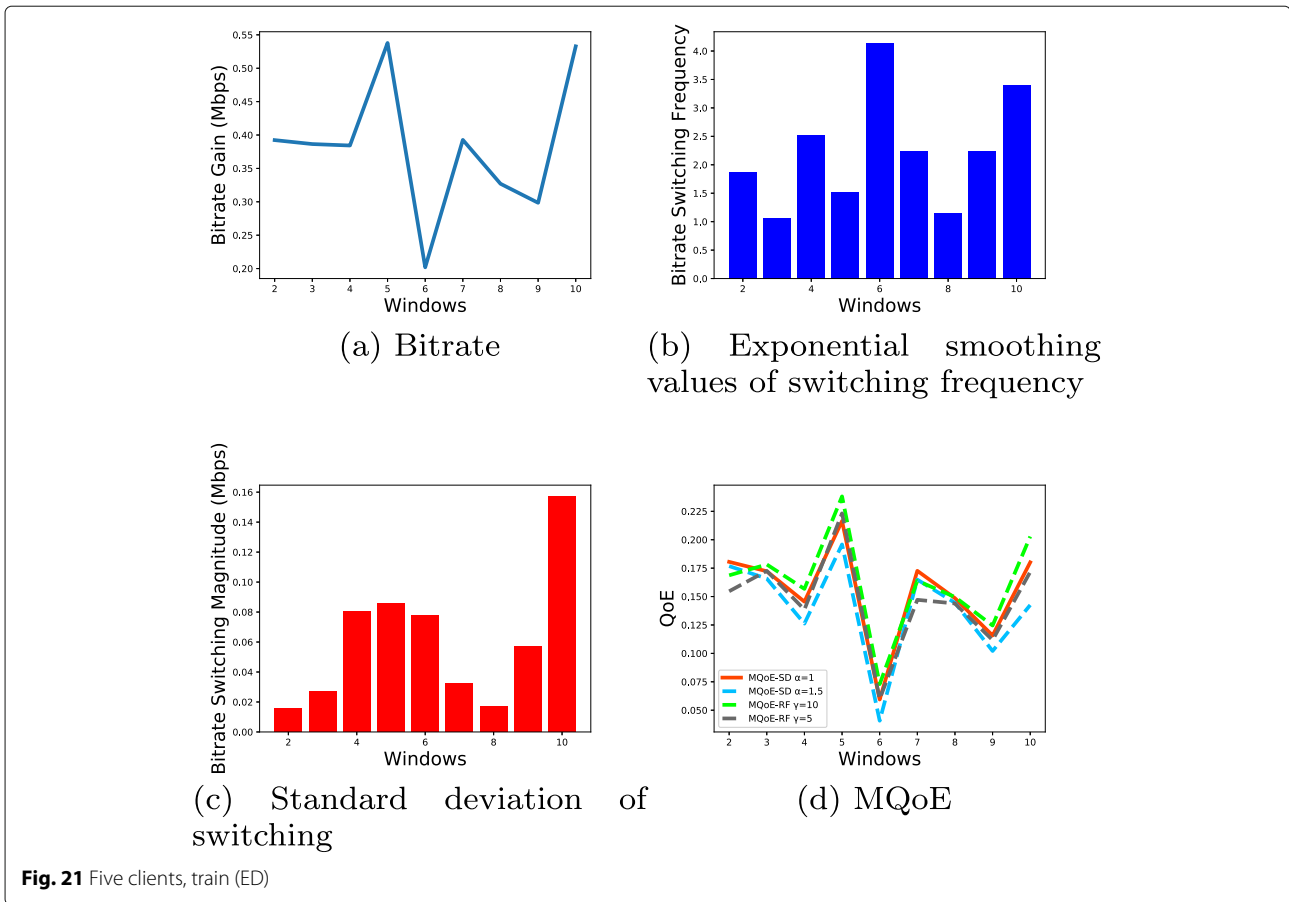
For the case of traveling in a ferry, the signal drops gradually as the ferry moves away from the departing shore, which impacts the available bandwidth. Thus, with all ferry scenarios with three, five and ten clients, we see the QoE value gradually drop for both MQoE models along with the bitrate (see Figs. 26, Fig. 27, and Fig. 28) for *Big Buck Bunny*. We do note a small difference in window 10 for all the scenarios as the bitrate shows a small increase from window 9 to 10.

With three clients, there is a small difference in MQoE_RF compared to MQoE_SD on each Δt

(MQoE_RF is higher than MQoE_SD). With MQoE_RF when γ decreases in set2, it does not reflect the small peaks of bitrate significantly. However, MQoE_SD is more sensitive on the magnitude of bitrate switching and shows bumps and dents even more than the bitrate.

For the five and ten clients also MQoE_RF is higher than MQoE_SD. For MQoE_RF the jumps on window 4 and window 7 are significantly higher than MQoE_SD. This is mostly affected by increase of the bitrate on those windows when there are more number of clients to compete and the value of bitrate switching frequency decreases. There are high values of bitrate switching magnitude in these windows, which does not let MQoE_SD to rise as much as MQoE_RF. For ten clients, the peaks get smaller when the α increases as the bitrate switching magnitude is large for these windows; by increasing α (in set2), its impact on the linear model MQoE_SD is noticeable. The trend in the case of *Elephants Dream* is very similar to *Big Buck Bunny* and, thus, is not shown here.

Recall that our model does not explicitly consider rebuffering. We found that only in the case of a ferry, we observed some rebuffering, which occurred as it reached the lowest point of the wireless signal. Certainly, rebuffering is a factor when there is a dead time. On the other hand, our QoE models capture this drop indirectly by



reporting a lower QoE value and assigning a zero if there are no segments transmitted at all in a window. Thus, as we postulated early on, rebuffering does not need to be explicitly captures in the MQoE models.

5.4 MQoE comparison with multi-client and fairness

We now discuss how increasing the number of clients impacts MQoE. With a higher number of the clients, the competition for the resources increases, which causes each client to get a smaller share of the resources. Smaller shares also causes a QoE degradation (see Fig. 29, shown for *Big Buck Bunny*). In the case of cars, MQoE_RF with $\gamma = 10$ (for set1), three clients scenario has higher MQoE value than for five and ten clients. With five clients, the QoE decreases about 36.86% on average compared to three clients. With ten clients, the QoE shows a decrease of 53.71% on average compared to five clients.

In the case of the train, regardless of the number of clients, the QoE drops to the lowest point at window 6. For the rest of the windows, QoE behavior with the train is showing almost the same pattern as with the car. On average, with five clients, the QoE decreases by 37% compared to three clients. With ten clients, the QoE shows a decrease of 47% compared to five clients.

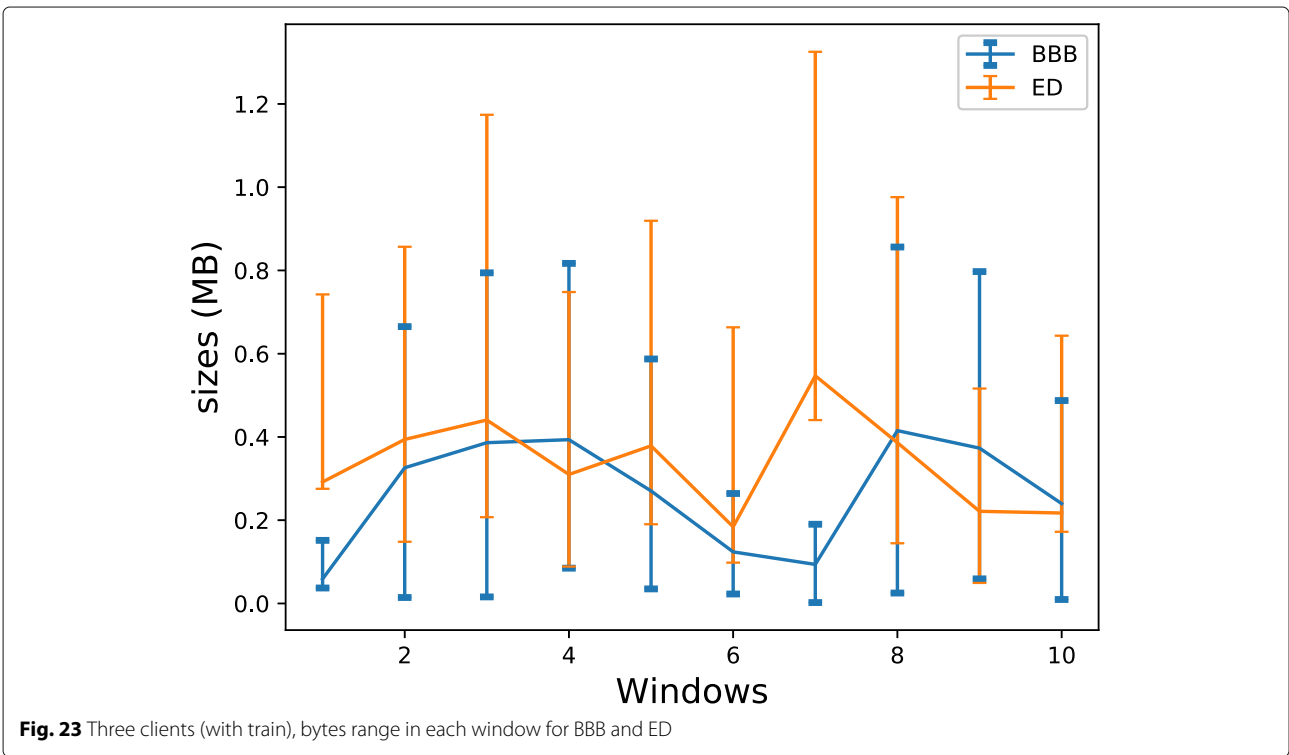
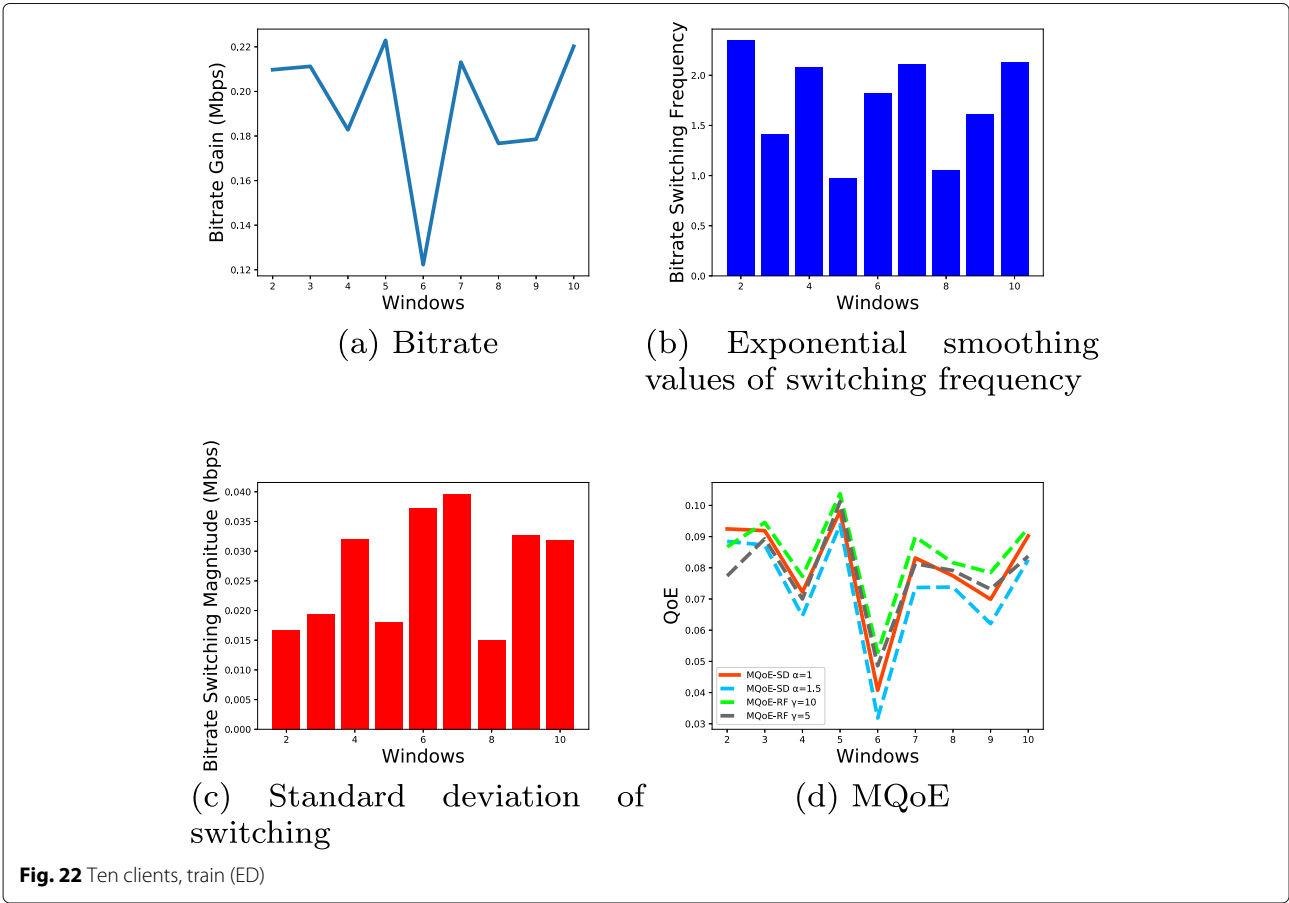
With the ferry, we see the decreasing trend from the starting window to the final window. But the QoE drop for the case with three clients is much more significant since the QoE was higher to start at the beginning compared to five and ten clients scenarios.

We show the results for three, five and ten client scenarios for MQoE_SD with $\alpha = 1$ in Fig. 30. We observed a similar trend as with MQoE_RF.

We also analyze how fairly each client is treated in terms of MQoE. In the case of three clients, for MQoE_RF of each client with $\gamma = 10$, the standard deviation was 6.89% compared to the average among them. With the same weight parameter, for ten clients this value decreased to 2.52%. The lower deviation indicates that clients are getting fairer share in a congested environment, although none are getting very high QoE. Recall that all clients watched the same video in our study. This shows that the ABR does not treat each client equally fairly except for in a congested environment.

6 Related work

There have been a number of works, which proposed QoE models with objective QoE metrics.



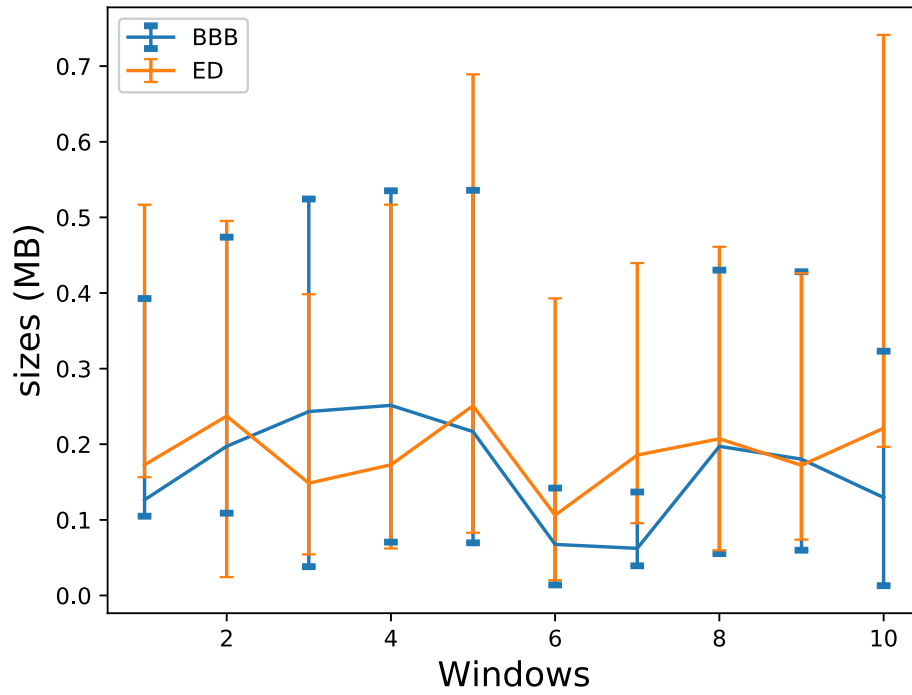


Fig. 24 Five clients (with train), bytes range in each window for BBB and ED

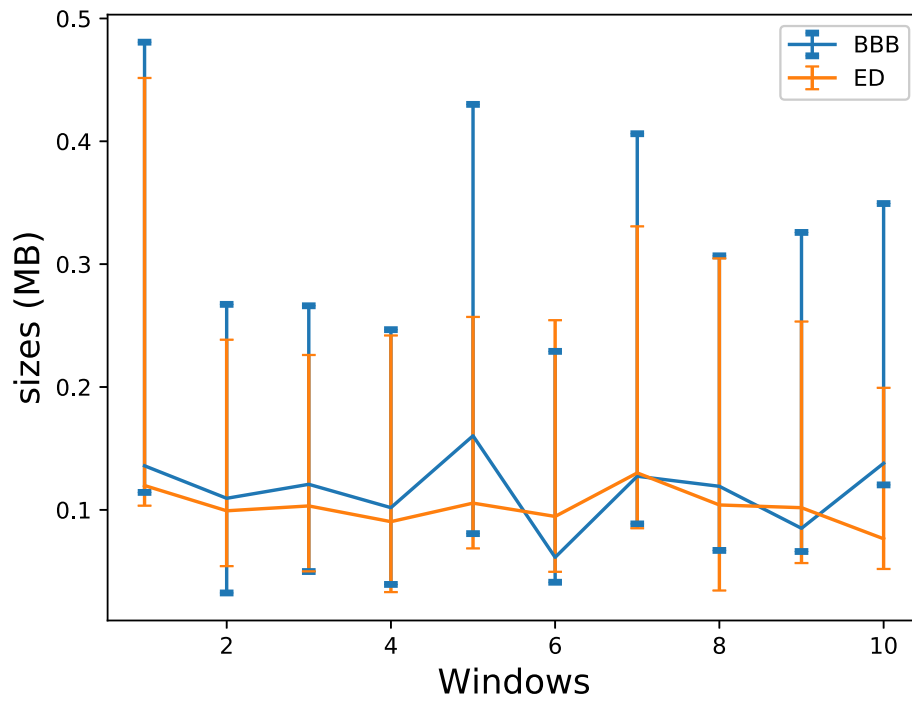
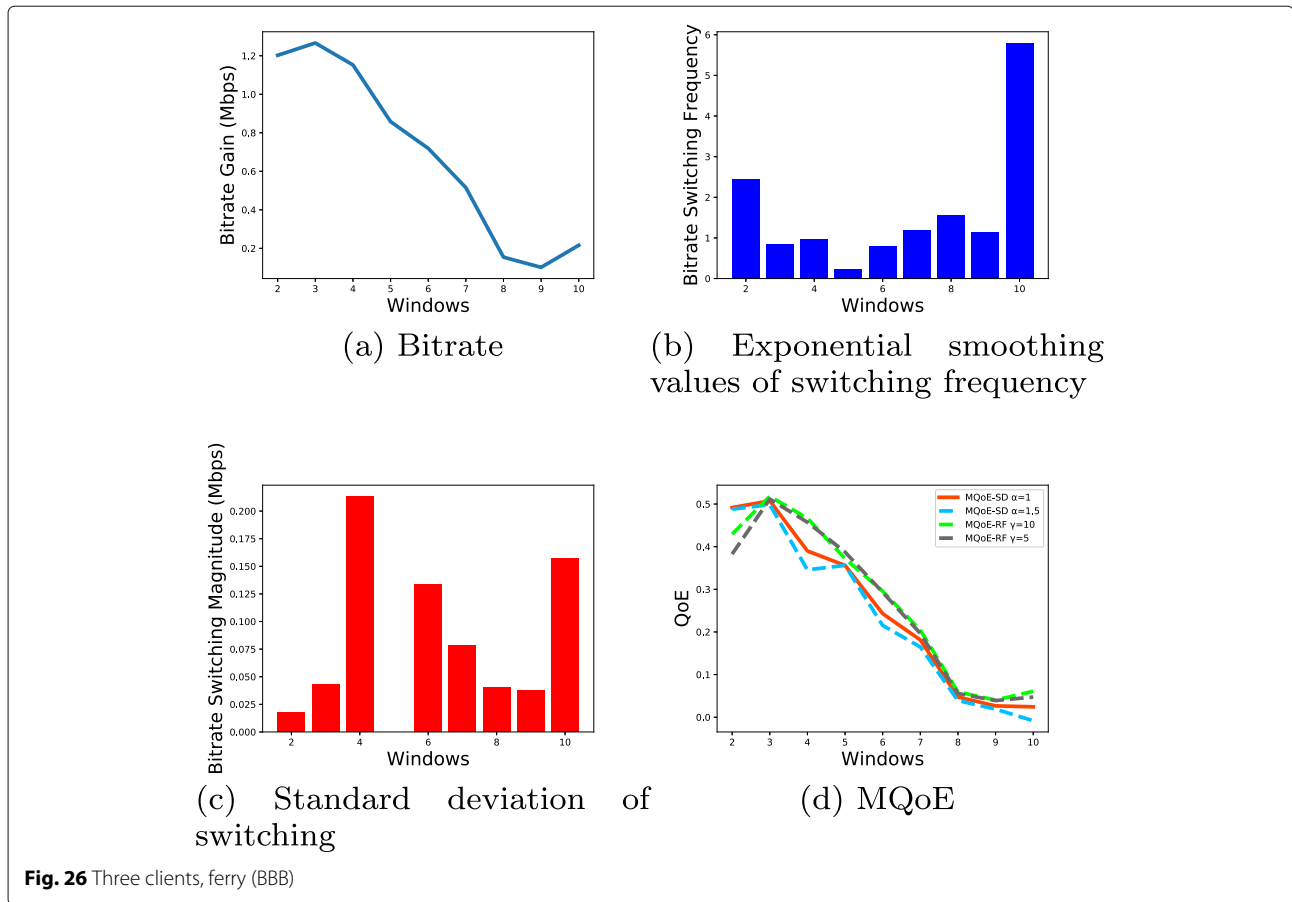


Fig. 25 Ten clients (with train), bytes range in each window for BBB and ED



Some work proposed models to capture the exponential relation between the QoE and QoS parameters [16–21]. However, QoS metrics are not sufficient to measure the satisfaction of the users. Due to the nonlinear relationship between these metrics, it is not easy to construct a simple model [7].

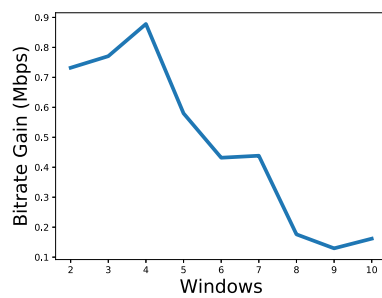
Most approaches formulated a linear parametric model with QoE objective metrics. Yin et al. [5] proposed a QoE model in the Model Predictive Control (MPC) approach; we refer to this model as the MPC QoE model. This QoE model was used in assessing QoE for video streaming [10]. They considered QoE metrics such as bitrate gain, rebuffering and the difference between the quality level of consecutive chunks (switching amplitude).

Hoßfeld et al. [22] discussed factors that influence a QoE model. Yarnagula et al. [23] formulated a complex parametric QoE model over a number of metrics. De Vriendt et al. [24] addressed the problem of how to assess QoE of an end user under the form of a prediction for the MOS. For surveys on QoE models for DASH, see [7, 25].

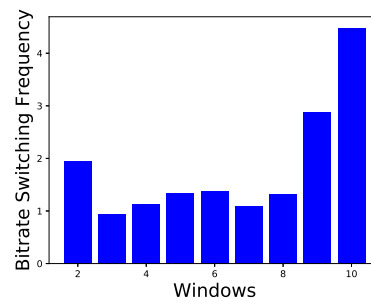
Wang et al. [26] proposed a model to maximize the QoE by considering the average video bitrate, frequency of variations and the amplitude of variations. The variation

metric is a centralized measure for the variation of the video quality around the average quality that is denoted as spectrum equation in [27]. Moldovan et al. [28] proposed a quadratic problem formulation which maximize both service quality and fairness. They define the objective as being to maximize the average quality, minimize the number of quality switches, and ensure equal utility (QoE) among users.

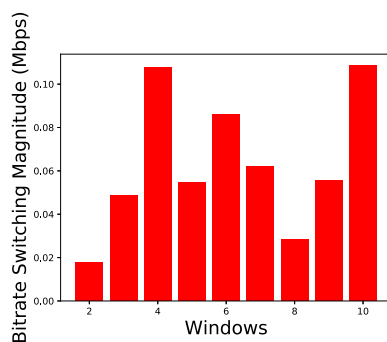
Xue et al. [29] proposed a model which combines instantaneous qualities and cumulative quality taking into account video segment quality. The instantaneous quality was obtained using a linear model using Quantization parameter (QP) values and instantaneous rebuffering. Guo et al. [30] proposed a model which estimates the overall quality using a linear combination of median and minimum of the instantaneous quality. The instantaneous quality was obtained from QP values using the normalized quality vs. inverted normalized quantization stepsize (NQQ) model. Tran et al. [31] presented a model considering encoded video quality and quality variation. The quality of the encoded video is calculated for each segment considering the average QP.



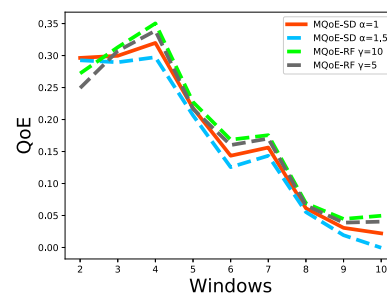
(a) Bitrate



(b) Exponential smoothing values of switching frequency



(c) Standard deviation of switching



(d) MQoE

Fig. 27 Five clients, ferry (BBB)

Manga et al. [32] presented a measurement study on QoE performance. However, their work do not present any moving QoE models.

Live streaming has also been investigated in a number of work on video streaming [33–35]. But these works focused on the QoE for a single client, and none are from the perspective of a content service provider.

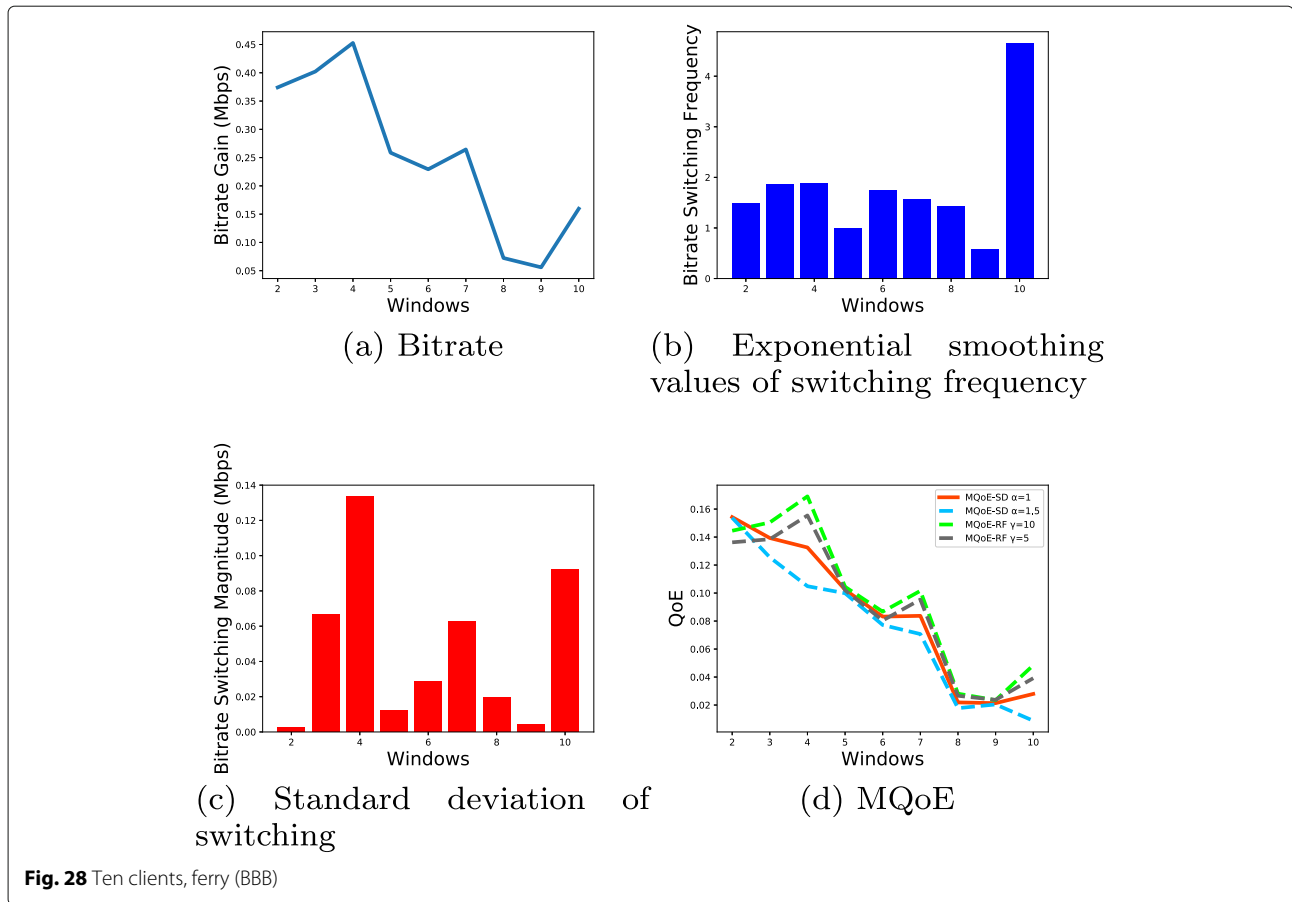
Note that our work focuses on quantitative moving QoE models. On the other hand, subjective QoE, measured using Mean Opinion Score (MOS), is studied for video delivery [23, 36]. It is also pointed out in [36] that subjective assessments are costly, time-consuming, and not scalable. First, we note that no previous work has studied subjective QoE from a content service provider's perspective. Secondly, subjective QoE assessment, for moving QoE snapshots, is impractical from a content provider's perspective for video quality monitoring. Consider a content service provider streaming a live event to thousands of end users. If subjects were to be employed to mimic understanding this situation, a significant number of subjects would be necessary. If monitoring is to be captured, say every minute for video stream monitoring, then this would require each subject to be reminded every minute

to record the MOS score, which may potentially require additional 10 to 15 seconds to record, the user would be distracted from continuing to watch the video for the next minute. Furthermore, fatigue could quickly set in even after a few minutes of scoring, resulting in noisy measures. Thus, as we proposed here, a quantitative MQoE approach is a more viable approach for monitoring video streaming for content service providers.

Most of the quantitative models discussed above were formulated for assessing QoE for an individual user or the QoE is reported at the end of a video session. Secondly, they were not readily adaptable in a multi-client scenario, especially in a scenario like a live event and on a rolling basis. Our proposed QoE models fill the void in the current literature. Our QoE model is the first work to address moving QoE (collectively for multiple clients) for video streaming which is more suited for a service provider point of view monitoring instead of individual client perspective.

7 Summary and future work

We presented two moving QoE models that can report ensemble QoE in review windows for multiple clients



streaming on a periodic basis. To our knowledge, we are the first to propose MQoE models for video streaming performance monitoring that can be used by content service providers. Secondly, the multi-client scenario has rarely been studied before, which is important to consider for content providers. Our study shows that such models can be used to understand the QoE behavior of multiple clients during streaming, especially for a video transmission such as for a live event. We also found out that a static QoE model such as MQoE_MO is not suitable for moving QoE. Our nonlinear model MQoE_RF is preferable when a content provider wants to capture the bitrate switching frequency in the QoE model. Our linear model MQoE_SD, in general, is useful at capturing the standard deviation of bitrate switching. Based on our observation, For weight parameters, $\gamma = 10$ with MQoE_RF and $\alpha = 1$ with MQoE_SD (i.e., set1) were found to be the best values to use to adequately capture the bitrate switching frequency and bitrate switching magnitude impact while keeping quality due to bitrates. Our work is expected to be useful to content providers (stakeholders) to observe variations for different conditions and fairness on QoE received by different clients so that they can take appropriate actions.

There are a number of additional studies that could be pursued. First, the window we used was as we observed for our study environment. When there is a much higher number of clients simultaneously streaming, the desirable window could be different than the one used in our study. In other words, a further study is needed to provide better recommendation on the optimal window size. Secondly, our study was limited to a maximum of ten simultaneous users. In real life, a very larger number of users may watch live events, coming from different devices and locations. In such a situation, videos are also not served from a single server since the content providers use a data center for video hosting. Thus, even with a very large number of clients streaming, they would be divided to be served by many servers. A monitoring analysis can be done based on each server as well as for a cluster of servers such as a rack of servers at a data center to understand MQoE behavior. In addition, this raises a scheduling need on which clients should be served from a specific server. For instance, a specific group of clients could be served from a particular server, either based on the clients' geographic locations or based on service level agreements to provide prioritized service to certain clients. It would be worthwhile to look into such associated problems in future research.

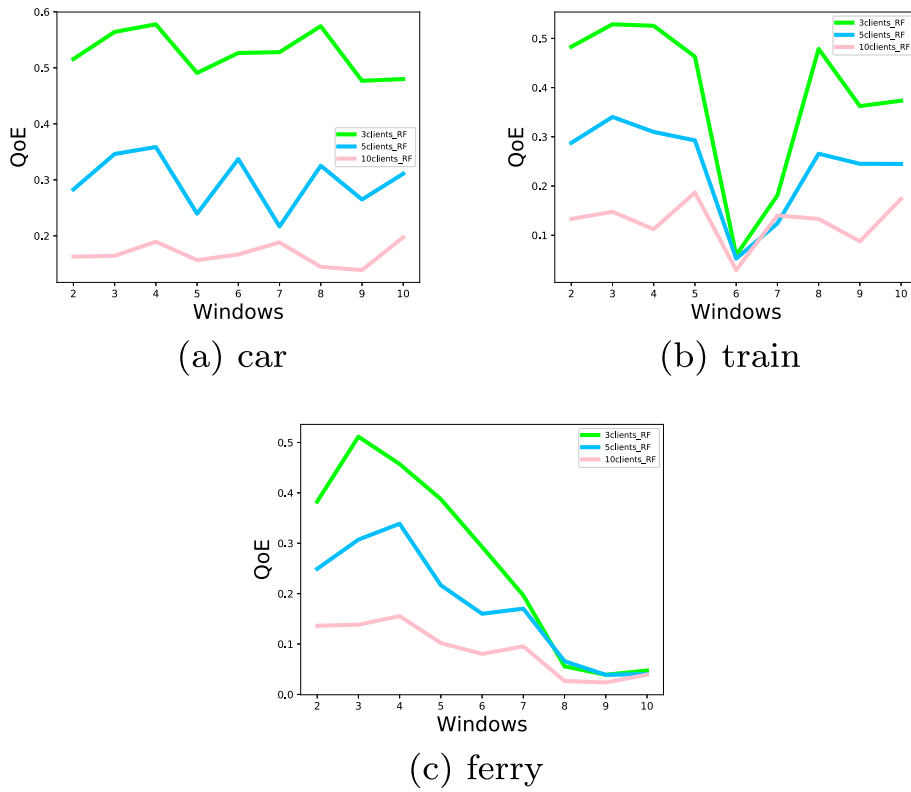


Fig. 29 Multi-client comparison, MQoE_{RF} $\gamma = 10$ (BBB)

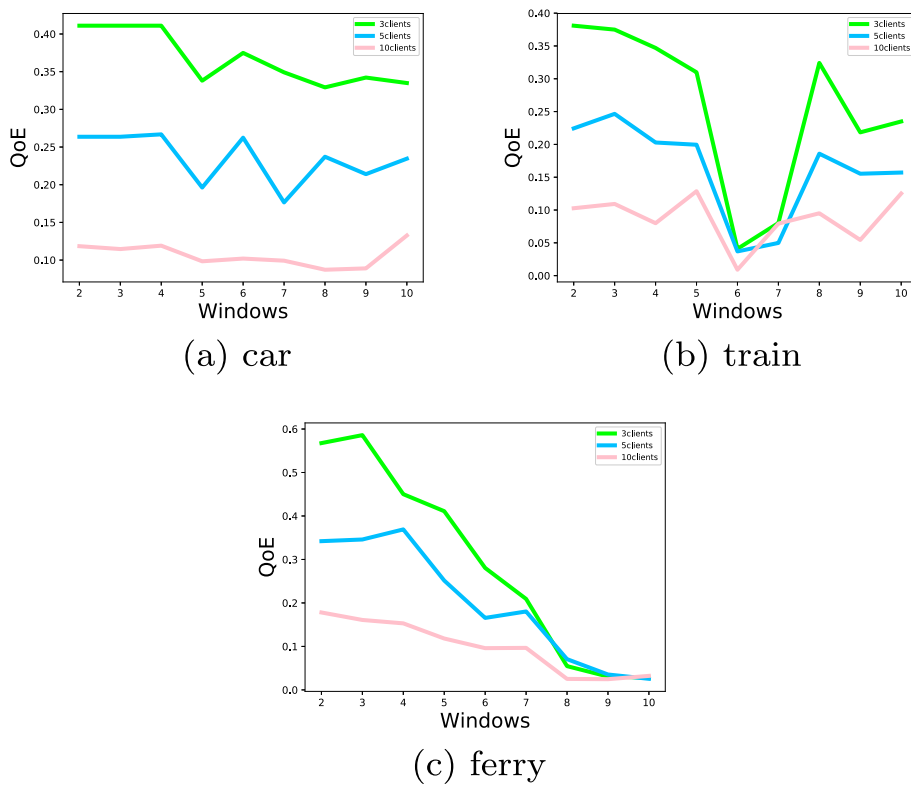


Fig. 30 Multi-client comparison, MQoE_{SD} $\alpha = 1$ (BBB)

Abbreviations

ABR: Adaptive bitrate; DASH: Dynamic adaptive streaming over HTTP; GENI: Global environment for network innovations; HTTP: Hypertext transfer protocol; ISP: Internet service provider; MOS: Mean opinion score; MPC: Model predictive control; MPD: Media presentation description; MPEG: Moving pictures experts group; MQoE: Moving quality of experience; QoE: Quality of experience; QoS: Quality of service

Acknowledgements

We thank the authors of [14] for making their dataset publicly available. Without their work, our work would not have been able to capture realistic mobility patterns.

Authors' contributions

SKM initially developed MQoE_RF and MQoE_SD with input from PJ, which was then fine-tuned by all three authors. DM initially suggested the MPC QoE extension for MQoE (MQoE_MO), which was finalized in discussion with SKM and PJ. SKM implemented the client and the server code on the GENI platform and ran all the experiments. All authors participated in the analysis of the experiment results. SKM and DM wrote the initial draft of the manuscript to which PJ provided his feedback towards the final draft. All three authors have read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The measured data from our experiments are made available at <https://github.com/Sheydakm/MQoE>.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they do not have competing interests.

Author details

¹Department of Computer Science & Electrical Engineering, University of Missouri-Kansas City, Kansas City, USA. ²GAIA, Ericsson, Santa Clara, USA.

Received: 17 November 2020 Accepted: 5 April 2021

Published online: 26 April 2021

References

1. Cisco Visual Networking. Forecast and methodology, 2016–2021. White Paper. 2017. <http://web.archive.org/web/20200214054830/https://www.reinvention.be/webhdfs/v1/docs/complete-white-paper-c11-481360.pdf>.
2. ISO/IEC. ISO/IEC 23009-1: 2019: Information technology–Dynamic adaptive streaming over HTTP (DASH)–Part 1: Media presentation description and segment formats: International Organization for Standardization Geneva; 2019. <https://www.iso.org/standard/75485.html>.
3. Mao H, Netravali R, Alizadeh M. Neural adaptive video streaming with Pensieve. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication. New York: ACM; 2017. p. 197–210.
4. Sodagar I. The MPEG-DASH standard for multimedia streaming over the internet. *IEEE Multimed*. 2011;18(4):62–7.
5. Yin X, Jindal A, Sekar V, Sinopoli B. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. New York: ACM; 2015. p. 325–38.
6. Seufert M, Egger S, Slanina M, Zinner T, Hoßfeld T, Tran-Gia P. A survey on quality of experience of HTTP adaptive streaming. *IEEE Commun Surv Tutor*. 2014;17(1):469–92.
7. Juluri P, Tamarapalli V, Medhi D. Measurement of quality of experience of video-on-demand services: A survey. *IEEE Commun Surv Tutor*. 2015;18(1):401–18.
8. GENI testbed. <https://portal.geni.net/>.
9. Delivering live YouTube content via DASH. <https://developers.google.com/youtube/v3/live/guides/encoding-with-dash>.
10. Yan FY, Ayers H, Zhu C, Fouladi S, Hong J, Zhang K, Levis P, Winstead K. Learning in situ: a randomized experiment in video streaming. In: 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20). Berkeley: Usenix; 2020. p. 495–511.
11. Kiani Mehr S, Medhi D. QoE performance for DASH videos in a smart cache environment. In: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). New York: IEEE; 2019. p. 388–94.
12. Dash.js reference client implementation. <https://github.com/Dash-Industry-Forum/dash.js>.
13. Kreuzberger C, Posch D, Hellwagner H. A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP. In: Proceedings of the 6th ACM Multimedia Systems Conference. New York: ACM; 2015. p. 213–8.
14. Riiser H, Vigmostad P, Griwodz C, Halvorsen P. Commute path bandwidth traces from 3G networks: analysis and applications. In: Proceedings of the 4th ACM Multimedia Systems Conference. New York: ACM; 2013. p. 114–8.
15. WonderShaper – A tool to limit network bandwidth in Linux. <https://www.tecmint.com/wondershaper-limit-network-bandwidth-in-linux/>.
16. Khirman S, Henriksen P. Relationship between quality-of-service and quality-of-experience for public internet service. In: Proc. of the 3rd workshop on passive and active measurement, vol. 1; 2002.
17. Shaikh J, Fiedler M, Collange D. Quality of experience from user and network perspectives. *Ann Telecommun*. 2010;65(1-2):47–57.
18. Alberti C, Renzi D, Timmerer C, Mueller C, Lederer S, Battista S, Mattavelli M. Automated QoE evaluation of dynamic adaptive streaming over HTTP. In: 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX). New York: IEEE; 2013. p. 58–63.
19. Shao B, Renzi D, Amon P, Xilouris G, Zotos N, Battista S, Kourtis A, Mattavelli M. An adaptive system for real-time scalable video streaming with end-to-end QoS control. In: 11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10. New York: IEEE; 2010. p. 1–4.
20. Singh KD, Hadjadj-Aoul Y, Rubino G. Quality of experience estimation for adaptive HTTP/TCP video streaming using H. 264/AVC. In: 2012 IEEE Consumer Communications and Networking Conference (CCNC). New York: IEEE; 2012. p. 127–31.
21. Balachandran A, Sekar V, Akella A, Seshan S, Stoica I, Zhang H. Developing a predictive model of quality of experience for internet video. *ACM SIGCOMM Comput Commun Rev*. 2013;43(4):339–50.
22. Hoßfeld T, Seufert M, Sieber C, Zinner T. Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming. In: 2014 sixth international workshop on quality of multimedia experience (qomex). New York: IEEE; 2014. p. 111–6.
23. Yarnagula HK, Juluri P, Kiani Mehr S, Tamarapalli V, Medhi D. QoE for mobile clients with segment-aware rate adaptation algorithm (SARA) for DASH video streaming. *ACM Trans Multimed Comput Commun Appl (TOMM)*. 2019;15(2):1–23.
24. De Vriendt J, De Vleeschauwer D, Robinson D. Model for estimating QoE of video delivered using HTTP adaptive streaming. In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013). New York: IEEE; 2013. p. 1288–93.
25. Barman N, Martini MG. QoE modeling for HTTP adaptive video streaming—a survey and open challenges. *IEEE Access*. 2019;7:30831–59.
26. Wang C, Bhat D, Rizk A, Zink M. Design and analysis of QoE-aware quality adaptation for DASH: A spectrum-based approach. *ACM Trans Multimed Comput Commun Appl (TOMM)*. 2017;13(3s):1–24.
27. Zink M, Schmitt J, Steinmetz R. Layer-encoded video in scalable adaptive streaming. *IEEE Trans Multimed*. 2005;7(1):75–84.
28. Moldovan C, Skorin-Kapov L, Heegaard PE, Hoßfeld T. Optimal fairness and quality in video streaming with multiple users. In: 2018 30th International Teletraffic Congress (ITC 30), vol. 1. New York: IEEE; 2018. p. 73–8.
29. Xue J, Zhang D-Q, Yu H, Chen CW. Assessing quality of experience for adaptive HTTP video streaming. In: 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW). New York: IEEE; 2014. p. 1–6.

30. Guo Z, Wang Y, Zhu X. Assessing the visual effect of non-periodic temporal variation of quantization stepsize in compressed video. In: 2015 IEEE International Conference on Image Processing (ICIP). New York: IEEE; 2015. p. 3121–5.
31. Tran HT, Vu T, Ngoc NP, Thang TC. A novel quality model for HTTP Adaptive Streaming. In: 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE). New York: IEEE; 2016. p. 423–8.
32. Mangla T, Halepovic E, Ammar M, Zegura E. Drop the Packets: Using Coarse-grained Data to detect Video Performance Issues. In: Proceedings of the ACM CoNEXT. New York: ACM; 2020.
33. Wei S, Swaminathan V. Low latency live video streaming over HTTP 2.0. In: Proceedings of Network and Operating System Support on Digital Audio and Video Workshop. New York: ACM; 2014. p. 37–42.
34. Timmerer C, Weinberger D, Smole M, Grandl R, Müller C, Lederer S. Live transcoding and streaming-as-a-service with MPEG-DASH. In: 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). New York: IEEE; 2015. p. 1–4.
35. Lohmar T, Einarsson T, Fröjdh P, Gabin F, Kampmann M. Dynamic adaptive HTTP streaming of live content. In: 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks. New York: IEEE; 2011. p. 1–8.
36. Mok RK, Chan EW, Luo X, Chang RK. Inferring the QoE of HTTP video streaming from user-viewing activities. In: Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack. New York: ACM; 2011. p. 31–36.

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] 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](https://www.springeropen.com)
