

# High Quality Network and Device Aware Multimedia Content Delivery for Mobile Cloud

**Muhammad Saleem<sup>1\*</sup> and Yasir Saleem<sup>1</sup>**

<sup>1</sup>Department of Computer Science & Engineering, University of Engineering and Technology, Lahore,  
Pakistan

[e-mail: saleemnawaz@gmail.com]

\*Corresponding author: Muhammad Saleem

*Received February 17, 2019; revised April 13, 2019; accepted May 19, 2019;  
published October 31, 2019*

---

## **Abstract**

The use of mobile devices is increasing in multimedia applications. The multimedia contents are delivered to mobile users over heterogeneous networks. Due to fluctuation in bandwidth and user mobility, the service providers are facing difficulties in providing Quality of Service (QoS) guaranteed delivery for multimedia applications. Multimedia applications depend on QoS parameters such as delay, bandwidth, and jitter to offer better user experience. The existing schemes use the single source and multisource delivery but are unable to balance between stream quality and network congestion for mobile users. We proposed a Quality Oriented Multimedia Content Delivery Scheme (QOMCDS) for the mobile cloud to deliver better quality multimedia contents for the mobile user. The multimedia contents are delivered to the mobile device based on the device's parameters and network environment. The objective video quality assessment models like Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Video Quality Measurement (VQM) are used to measure the quality of the video. The client side Quality of Experience metric such as Startup delay, Rebuffering events, and Bitrate switch count was used for evaluation. The proposed scheme is evaluated using dash.js and is compared to existing schemes. The results show significant improvement over existing multimedia content delivery schemes.

---

**Keywords:** Multi-source, mobile client, mobile cloud, quality of service, quality of experience.

## 1. Introduction

**T**HE usage of smartphones and tablets is increasing due to the widespread growth of mobile devices. Advances in mobile capabilities, together with Long Term Evolution networks, and Wi-Fi access have brought rich user experience for mobile users [1]. According to Cisco [2], by 2021, 78% of world video traffic will be mobile data traffic and 86% of the mobile data traffic would be generated by smartphones. Considering the increasing popularity of multimedia technology, people are using mobile devices for watching multimedia videos through streaming. Irrespective of the service provider, the users always expect stability in streaming service, which is an essential function for multimedia videos. The smooth watching of videos is the concern of viewers; no matter what issues are occurring in the network.

Likewise, the playback quality is inconsistent with existing videos platforms due to network quality fluctuation. This issue is frequent on mobile devices due to the limited hardware resources and bandwidth. With the rapid increase in the number of users, the limited bandwidth causes delays in the network and the multimedia services are affected. The increasing use of Facebook, Instagram, and Snapchat are forcing network service providers to expand the network capacity. Expanding network capacity will not be sufficient without considering the quality of experience (QoE) which is gaining popularity with growing user expectation for quality of service. The focus is now shifting to propose innovative solutions to enable QoE [3], [4].

The Dynamic Adaptive Streaming over HTTP (DASH) was proposed to further improve the video streaming quality. This technique has gained much attention in recent times. When the DASH is employed, the videos are divided into small chunks encoded with different bitrates. The DASH helps in providing enhanced video streaming by using the dynamic adaptive bitrate selection and current quality levels. On the basis of this research, several solutions and implementations are presented by using DASH [5]–[12] and all the research works confirm that video streaming and service quality is enhanced. However, there are still some limitations in current research for mobile devices due to limited mobile resources, mobility and bandwidth variations.

Several device and network aware techniques [10],[13] have been proposed which consider the device resolution, bandwidth, battery, and processing speed to deliver multimedia contents for mobile devices. The multisource schemes [14] use multiple video streams for the mobile user which overload the network and increase congestion on the network. An optimal method is needed which ensures smooth playback and avoids video freeze by considering mobility and network variations.

The multimedia packets need correctness, real time data, and sequence in order to maintain streaming quality as compared to general services [15],[16]. The service quality for multimedia applications declines while meeting the QoS demands due to mobility and bandwidth variations. The smooth playback of video with limited hardware resources and bandwidth is a challenging task for mobile cloud applications. The quality of experience is challenging for the mobile user because of different processing powers, resolution, mobility and available bandwidth. An optimal technique is proposed which considers device and network parameters to deliver the best quality contents for the mobile user. Specifically, the contributions of this research are summarized as follows:

- Mobile device and network parameters are estimated to deliver the best video for the mobile client.
- The Lyapunov optimization is used to select the optimal bitrate for the next video streaming.
- The proposed QOMCDS is validated through detailed simulations using real HSDPA and LTE datasets. The simulation results show QOMCDS is able to improve the video streaming quality in comparison with existing schemes.

This paper introduces an intelligent quality oriented multimedia content delivery scheme (QOMCDS). Based on the current traffic load and mobile resources, the multimedia contents are delivered intelligently to the mobile user. The scheme dynamically changes the bitrate for multimedia delivery to meet mobile QoS requirements and enhances Quality of experience. The paper is structured as follows. The related work is discussed for multimedia content delivery mechanisms in Section 2. Video Quality assessment techniques are also described in related work. Section 3 covers the proposed work along with algorithms. The experimental results are discussed in Section 4. Section 5 concludes the paper.

## 2. Related Work

This paper presents a Quality Oriented Multimedia content delivery scheme for mobile users. The existing research work on multimedia content delivery and quality assessment methods are discussed in this section.

### 2.1 Multimedia Content Delivery

Recently, more attention has been given to provide high quality video streaming for mobile users. The quality of service metrics is the sum of parameters for delay, jitter, loss, and throughput. The quality of experience depends on quality of service at server side, network layer and mobile user mobility. The authors of [15] worked on the improvement of mobile multimedia streaming services in cloud computing. They presented a network and device aware QoS scheme to deal with issues related to limited bandwidth and terminal power. The authors of [17] worked on cloud based mobile media environment. They highlighted the issues due to mobility as moving bulk of data over large distance results in congestion over the internet which degrades the Quality of Experience (QoE) for the user.

Adaptive video streaming is drawing attraction nowadays due to its flexible adaptation of video streaming quality according to communication requirements. The collaborative and cooperative techniques are discussed and implemented in [18] which highlight and resolve the issues regarding resource competing among various users. Another bitrate adaptation algorithm is introduced in [5], that performs on the basis of throughput estimation while adopting adaptive forgetting factor. A new virtual caching scheme is presented by the authors in [19] for the adaptive video streaming that performs on the basis of network function virtualization and helps in storing and saving the resources.

The evaluation of three adaptive HTTP streaming players Adobe OSMF, Microsoft Smooth Streaming and Netflix player is performed in [20]. Microsoft Streaming can adapt the highest sustained bitrate because it is useful under persistent and unrestricted bandwidth. It also accumulates larger playback buffer as compared to other adaptive streaming players. However, it reacts slower to the long and unnecessary adaptation and the short term bandwidth fluctuations. The Netflix is more committed and aggressive to provide high

quality video as compared to Microsoft Smooth Streaming, although both share the same shortcomings. The Adobe OSMF usually fails to adopt the appropriate bitrate even under the stable bandwidth. The authors also argued that the rate adaptation logics interact at TCP congestion layer and application layer. The transport layer is still challenging for smooth adaptive streaming.

A media edge cloud architecture was proposed for high QoS for multimedia applications in which storage, sharing, adaptation, delivery, and retrieval utilize cloud computing to achieve a better quality of experience [21]. In [22] a request scheduler is proposed to distribute request over different interfaces in parallel. The video is divided in smaller chunks with the constant duration, which enables segments to be requested over separate links to efficiently utilize bandwidth. Mobile Multi-source High Quality Multimedia Delivery Scheme (M3QD) introduces content distribution over infrastructure-based and hybrid ad-hoc networks [14].

Buffer-Based Approach to Rate Adaptation (BBA) [23] uses estimation and current buffer allocation to select the video bitrate. This algorithm starts with the low bitrate and with the increase in buffer allocation a higher bitrate is selected to smooth variation caused by varying capacity. The buffer-based algorithm is not suitable for short videos because it assumes the large buffer size. The BBA often provides the lowest QoE even under better network conditions.

A rate adaptation algorithm (AIMD) [24] detects bandwidth variations using HTTP throughput for adaptive HTTP streaming. The HTTP throughput is measured by using the segment fetch time. The bandwidth capacity of current media stream for end to end network is decided on the basis of smoothed HTTP throughput.

A Feedback Linearization Adaptive Streaming Controller (ELASTIC) [25] is a stream switching controller that avoids the network traffic variations using feedback control theory. Another method for throughput estimation is to find the harmonic mean of a certain number of previous measurements. The ELASTIC algorithm selects previous five and the FESTIVE algorithm [26] selects the previous twenty measurements to find out the harmonic mean. This harmonic mean method shows optimal performance when the environment has uniform measurements because it is able to distinguish a few outlier measurements. However, it experiences some short-term bandwidth variations while using the wireless environment which can result in underestimation or the overestimation of the available bandwidth.

A QoE-aware DASH system (QDASH) [27] is a probing methodology to measure network bandwidth. It is sensitive to the variations in available bandwidth and gives accurate measurements for video quality, supported by the current network conditions. The Rate-based [28] approach uses ABR controller for the DASH standard. It chooses the highest possible bitrate which is lower than throughput prediction based on arithmetic mean of last 5 chunks.

## 2.2 Video Quality Assessment

Video Quality Metric (VQM) measures the perceived video quality. It measures the blurring, global noise, unnatural motion and color distortion which are perceptual effects of video impairment. The results are combined into a single metric. It is an objective video quality assessment method and has been adopted by ANSI [29].

Peak Signal-to-Noise Ratio (PSNR) is an assessment method for images. It is calculated by finding the maximum power of the signal and the difference between the reference and test image [30], [31]. Lee et al. [32] define the PSNR as shown in equation 1.

$$PSNR_{db} = 20 \log_{10} \frac{MAX_{bitrate}}{\sqrt{EXP_{thr} - CRT_{thr}}} \quad (1)$$

The  $MAX_{bitrate}$  represent the bitrate after the encoding process.  $EXP_{thr}$  is the expected throughput of the multimedia stream over the network. The  $CRT_{thr}$  represent the actual throughput of the stream.

The Structural similarity (SSIM) is chosen to evaluate the QoE metric as it provides a more reliable representation of the perceived video quality. The SSIM uses distortion, luminance, and contrast to measure the correlation of the videos. The SSIM index values are varying within 0-1 and values near to 1 means better quality.

SSIM extracts the structural information using HVS and structural similarity is measured between the original and the image under observation [33]. The SSIM index is defined in equation 2.

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot s[(x, y)]^\gamma \quad (2)$$

The parameters  $\alpha$ ,  $\beta$ , and  $\gamma$  in equation 2 adjust three comparison functions.

We have used Bitrate switch count, Startup delay and Rebuffering events to evaluate QoE media metrics introduced by DASH industry forum. Bitrate switch count is an important QoE media metric. It is the number of switches due to quality variations while playing a video session that has multiple bitrates. It is used to quantify the flicker effects due to quality level variations. The Startup delay is a metric that represents a time duration to fill the sufficient buffer for playing the video after initiating a connection to the server. It is measured in seconds. The Rebuffer count represents the total number of interruptions during a video session.

### 3. Quality Oriented Multimedia Content Delivery Scheme

We start with introducing the multimedia contents delivery from a cloud server to the mobile user. The client sends a request for the multimedia contents and server responds with the data stream. The mobile client sends resolution class and quality level to the cloud server. The server delivers the multimedia stream to the mobile user based on the resolution class and quality level parameters received from the mobile client.

#### 3.1 System Architecture

The system architecture consists of a mobile client, cloud content management service and cloud media server as shown in Fig. 1. The cloud content management service deals with all the communication between the server and the mobile client. The cloud media server deals and stores all the multimedia data. The mobile client has Network and Device Parameters Estimation (NDPE) module to estimate device parameters and available bandwidth. The rate adaptation algorithm determines the desired bitrate for the next playback.

The cloud content management service is an important component in the content delivery system that is responsible for receiving a client request, quality monitoring, stream adaptation, and content delivery. The content management has Device Information Handler (DIM), Stream Manager (SM), Media Quality Monitor (MQM), Media Stream Controller (MSC) and Device Profile Manager (DPM).

The Mobile client has NDPE module and rate adaption module to measure the network conditions and desired quality level for the multimedia streaming respectively. The NDPE module determines the device parameters which are used by the bitrate adaptation module.

The mobile client sends media content request to the Device Information Handler. The device information handler is used to receive resolution class and quality level from the mobile client. When the mobile device using the cloud service for the first time, the request is initially sent to the DPM. The device parameters are stored and updated in the DPM. The MQM gets the device detail from the DPM to determine the optimal stream to be sent to the client. The Device Profile Manager shares information with Stream Manager. The Stream Manager is responsible to find sources of requested multimedia contents. It selects a file for Segmentation and Transcoding. The multimedia file is encoded in different bitrates and stored on the Cloud Server. The media stream controller fetches contents from a server and delivers to the mobile client.

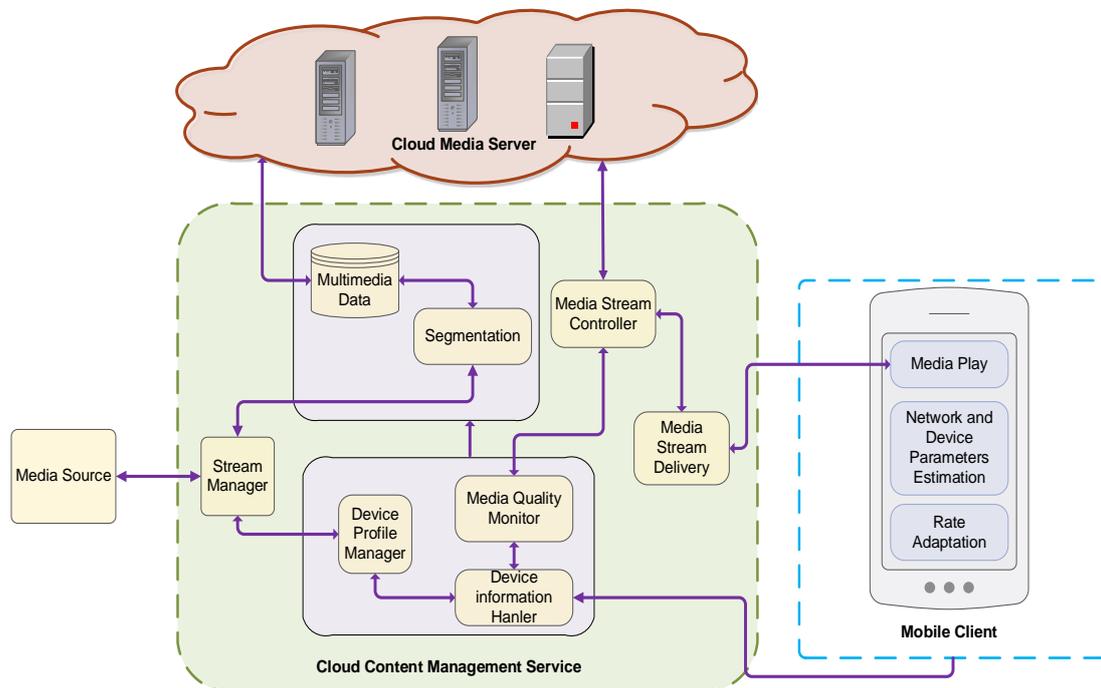


Fig. 1. Quality Oriented Multimedia Content Delivery Architecture

The client determines the estimated desired quality level based on available bandwidth, CPU usage, remaining battery, quality level fluctuation, and resolution. The users that have a good connection and can completely play the video will avail a high quality multimedia stream. The users with the poor connection will receive a lower quality video stream.

### 3.2 System Model and Design Requirements

A video is divided in small chunks in a DASH system and the chunk set is denoted by  $\mathcal{S} = \{1, 2, \dots, s\}$ . Each of the chunks is encoded at different bitrates and the duration of each chunk consists of 2 seconds. The length of chunk  $k$  that belongs to chunk set  $\mathcal{S}$  is  $t_l(k)$ . The chosen bitrate of chunk  $k$  for a mobile user is denoted by  $r^k(t)$ . The number of frames that are produced by  $k$  chunk at time  $t$  is calculated as in equation 3.

$$c^k(t) = \frac{r^k(t) \Delta t}{L} \quad (3)$$

The minimum bitrate is set to 235 kbps to ensure acceptable video quality for handheld devices.

**Table 1** shows the symbols used in this paper.

**Table 1.** Notations

Symbol	Explanation
$AB_{est}$	Estimated available bandwidth determined using round trip time
$C_n$	Chunk number of the multimedia packet
$S_l$	Large packet used to measure round trip time
$S_s$	Small packet used to measure round trip time
$RTT_l$	Round trip time measured with large packet sent to the server
$RTT_s$	Round trip time measured using small packet
$Q_{prev}$	Quality level of the previous chunk
$Q_{next}$	Quality level for next chunk
$\varphi$	Frequency of switches
$\mu$	Average quality level variation
$MOS_{pred}$	Predicted Mean Opinion Score
$D_t$	Battery Discharge Time
$U$	Available CPU resource
$V_p$	The time required to play the remaining video
$C_d$	The CPU resources needed to decode the chunk

The main goal of adaptive video streaming is to dynamically choose the video level  $l(t) \in L = \{l_0, \dots, l_{N-1}\}$  to enhance the Quality of Experience for each video chunk. The frequent quality switching could be annoying to the user. Reducing quality level switching for adaptive video streaming is a key requirement to increase QoE. A video level received by the client is an important factor that enhances the videos QoE. The higher the bitrate and resolution, the better the video quality. The design goals for the proposed QOMCDS is to reduce quality level switches, enhance the received video level and reduce the rebuffering events.

**Table 2** shows the different classes based on resolution, battery capacity and battery voltage for mobile devices.

**Table 2.** Classification of Mobile Devices Based on a Previous Study on 4914 Devices [22]

Device Classes	Class 1	Class 2	Class 3	Class 4	Class 5
Resolution Ranges	$\leq 1024 \times 768$	$1024 \times 768,$ $800 \times 600$	$800 \times 600,$ $480 \times 360$	$480 \times 360,$ $320 \times 240$	$< 320 \times 240$
Resolution	$720 \times 1280$	$540 \times 960$	$480 \times 800$	$320 \times 480$	$240 \times 320$
Battery Capacity	2100 mAh	1900 mAh	1650 mAh	1400 mAh	1200 mAh
Battery Voltage	3.8 V	3.8 V	3.7 V	3.7 V	3.7 V

The six quality levels from highest to lowest are 3000kbps, 1750kbps, 1050kbps, 750kbps, 375kbps, and 235kbps.

### 3.2.1 Network and Device Parameters Estimation

The NDPE module estimates the available bandwidth. GPING-Pair [34] is used for client side bandwidth estimation which uses RTT-based method. The GPING-Pair measures the available bandwidth between the first gateway and the client. This path is subject to ISP admission control and various interference. It can become the bottleneck for the entire connection to server. The Ping program is used to check the connectivity to a server and to find the ISP gateway. GPing-Pair method is modified to find a more accurate average available bandwidth for mobile clients. The average is calculated to achieve smooth video streaming for the mobile client. Two lowest RTT values are used by GPing-Pair at 64 bytes and 1064 bytes in order to determine the highest available bandwidth. We have used a notable difference in packet sizes to avoid fragmentation and achieve more precise RTT information. This GPing-Pair sends 10 ping pairs of 64 and 1064 bytes to calculate the lowest RTT by using echo replies.

The estimated available bandwidth is calculated using equation 4.

$$AB_{est} = \frac{4*(S_l - S_s)}{(\min(RTT_l) + \max(RTT_l)) - (\min(RTT_s) + \max(RTT_s))} \quad (4)$$

We consider that the path is symmetric and the value of RTT is divided by the factor 2 for computing one-side delay. Here, the  $S_s$  and  $S_l$  show packet sizes of a ping pair. Hence  $RTT_s$  and  $RTT_l$  define the RTT measurement of each ping pair.

The proposed scheme selects the highest bitrate of the video quality within the  $AB_{est}$ .

The bitrate of the next segment can be shown using equation 5.

$$Q_{i+1} = \max\{i: Q_i < AB_{est}\} \quad (5)$$

$Q_{i+1}$  is the bitrate for the next segment and  $Q_i$  denotes the i-th video quality bitrate.

The latest quality level is denoted by  $Q_l$ . The average video quality is taken by the average of all video chunks and calculated using equation 6.

$$Q_{avg} = \frac{\sum_{i=1}^{N-1} Q_i}{N-1} \quad (6)$$

The number of quality switches indicate the number of time the selected quality level is changed during time t and it is calculated by using a counter which is incremented when  $Q_l \neq Q_{l+1}$ .

The frequency of switches is denoted by  $\varphi$  and calculated using equation 7.

$$\varphi = \frac{\sum_{i=1}^{N-1} Q\{Q_i \neq Q_{i+1}\}}{N-1} \quad (7)$$

The average quality level variation is denoted by  $\mu$  and is calculated using equation 8.

$$\mu = \frac{\sum_{i=1}^{N-1} |Q_i - Q_{i-1}|}{N-1} \quad (8)$$

The standard deviation is denoted by  $\sigma$  and calculated using equation 9.

$$\sigma = \sqrt{\frac{\sum_{i=0}^n (Q_i - \mu)^2}{n-1}} \quad (9)$$

We can measure the MOS from the frequency of switches, average quality level variation and standard deviation of the video quality. The predicted MOS is calculated using equation 10.

$$MOS_{pred} = Q_{avg} - \alpha \cdot \mu - \beta \cdot \sigma - \gamma \cdot \varphi \quad (10)$$

The NDPE module measures the resolution, available battery and CPU load of the mobile device. The battery status API [35] is used to determine the discharge time of the battery. The battery manager interface describes the current battery information of the device. The *dischargingTime* attribute shows the time left in seconds till the battery is completely discharged. The discharge time in minutes is shown in equation 11.

$$D_t = \frac{dischargingTime}{60} \quad (11)$$

The CPU utilization is calculated from CPU busy time and idle time by using JavaScript code. The available CPU resource is denoted by U and calculated using equation 12.

$$U = 100 - \frac{Busy\_Time}{Busy\_Time + Idle\_Time} \quad (12)$$

### 3.2.2 Problem Formulation and Transformation

To design an effective bitrate adaptation scheme, we used the idea of the Lyapunov optimization framework [36]. The optimization of Lyapunov function can select the video rate according to the current state of the device and network environment. We have used resolution, available bandwidth, video quality fluctuation, available battery, and CPU utilization to select the bitrate of the next video chunk. The core idea of our approach is to use a Lyapunov function to minimize the output with a greedy algorithm.

We want to increase the bitrate for chunk  $c^k(t)$  as shown in equation 13.

$$\begin{aligned} & \max \quad MOS_{pred}, \quad \forall t \\ & s. t. \quad c^k(t) \leq AB_{est}(t) \\ & \quad \quad D_t \geq V_p \\ & \quad \quad U \geq C_d \end{aligned} \quad (13)$$

The solution of equation 13 can be found using dynamic programming methods. However, the computation of these methods is complex. Due to this Lyapunov optimization has become a better choice for mobile devices due to low computation complexity and less prior knowledge.

To satisfy equation 13, we need to consider various factors such as video streaming bitrate, video quality, quality level variation, CPU utilization, and battery usage. We can use Lyapunov optimization to transform this problem. The main idea is to use virtual queues to reduce the complexity of the problem for constraints of the optimization problem. The virtual queues are kept stable to satisfy the constraints.

The aggregate queue backlog is measured by the addition of the squares of the recent queue backlog. The output of Lyapunov function at time t is denoted by L(t) and defined in equation 14.

$$L(t) = \frac{1}{2} [Q_{avg}^2 + \mu^2 + \varphi^2 + \sigma^2] \quad (14)$$

$L(t)$  shows the overall status of the queue lengths. A large value of  $L(t)$  signifies at least one queue is congested. A small value of  $L(t)$  means that all queues are short. We assume all queues are empty initially at  $t = 0$ . The Lyapunov drift is formulated as shown in equation 15.

$$\Delta(t) \cong E\{L(t+1) - L(t)\} \quad (15)$$

$\Delta(t)$  shows the variation of Lyapunov function during a one-time slot. If  $\Delta(t)$  becomes smaller the queues will be stable. Therefore, improving queue stability and maximizing the quality of experience of video streaming is the goal of the optimization. The drift minus penalty is shown in equation 16.

$$\Delta(t) - V\mathbb{E}\{MOS_{pred}(t)\} \leq D_r \quad (16)$$

Where  $V$  is non-negative parameters used to represents a video streaming quality. It controls the drift  $D_r$  and the reward  $\mathbb{E}\{MOS_{pred}(t)\}$ . A large value of  $V$  indicates a higher priority assigned to maximize QoE but with larger queue lengths. According to the Lyapunov optimization technique, the decision should be chosen to reduce an upper bound during each time slot  $t$ . The upper bound of drift minus penalty is shown in equation 17.

$$\begin{aligned} & \mathbb{E} \left\{ \frac{\sum_{i=1}^{N-1} Q_i}{N-1} \right\} \\ & - \mathbb{E} \left\{ \frac{\sum_{i=1}^{N-1} |Q_i - Q_{i-1}|}{N-1} \right\} \\ & - \mathbb{E} \left\{ \frac{\sum_{i=1}^{N-1} Q_i \{Q_i \neq Q_{i+1}\}}{N-1} \right\} \\ & - \mathbb{E} \left\{ \sqrt{\frac{\sum_{i=0}^n (Q_i - \mu)^2}{n-1}} \right\} \end{aligned} \quad (17)$$

The optimization is achieved by taking the lowest value of the right side of the above inequality. The equation 17 provides insight into deciding the maximum value of  $c^k(t)$  to enhance the video quality experience for the mobile user. The problem in equation 13 is transformed into a mathematically tractable form. It provides optimization of adaptive video streaming by using video quality, frequency of quality switches and quality level variations. The purpose of the optimization is to select an appropriate bitrate for the next chunk and maintain the stability of the queues.

### 3.2.3 Rate adaptation algorithm

The quality level and resolution class are determined using the rate adaptation algorithm. The desired quality level is estimated from available bandwidth, available processing resource, available battery, resolution, frequency of switches and average quality level variation. The rate adaptation algorithm is performed by the mobile client using parameters. Initially, the lowest bitrate is chosen to be sent to the server. Later, the bitrate is chosen based on the parameters.

We determine the resolution class using the resolution of the device. We select the supported bitrate based on the device resolution and available bandwidth. The supported bitrate is chosen such that it doesn't exceed available bandwidth. We find the optimal bitrate for the next chunk using Lyapunov optimization by using equation 17. The Lyapunov function determines the bitrate for the next chunk by making upper bound of the drift plus penalty minimal. The optimal bitrate is selected based on the bitrate switching rate, average

quality level, and quality variations. The algorithm decides the maximum bitrate by considering the factors to reduce quality variations and bitrate switching rate. When the selected bitrate using equation 17 is greater than supported bitrate, the bitrate for the next chunk is chosen as supported bitrate. We find the CPU utilization using equation 12 and estimate CPU resources required to decode the next chunk. We find the battery draining time using equation 9. If the processing resource is not sufficient to decode the next chunk, then lower the quality level of the next chunk. In case the battery is not sufficient to play the selected video, lower the quality level of the video.  $R_c$  is the resolution class of the selected device.  $Q_{next}$  is the estimated quality level for next video chunk.

**Algorithm 1: Rate Adaptation Algorithm**

**Input:** R: Resolution of the device

$AB_{est}$ : Estimated available bandwidth

$D_t$ : Battery draining time

$Q_{avg}$ : Average quality level

$\mu$ : Average quality variations

$\phi$ : Bitrate switching rate

$U$ : CPU utilization

$Q_{prev}$ : Quality level of the previous chunk

**Output:**  $R_c$  and  $Q_{next}$

```

1:  $R_c = \text{ResolutionClassBasedOnDeviceResolution}(R)$ ;
2:  $Q_{supported} = \text{BitRateLevelSelectionBasedOnAvaivableBandwidth}(R_c, AB_{est})$ ;
3: // Determine bitrate of the next chunk by making the upper bound of the drift-plus
4: //penalty function minimal
5:  $Q_{next} = \text{FindOptimalBitrate}(Q_{avg}, \mu, \phi)$  // using equation 17
6: if ( $Q_{next} > Q_{supported}$  )
7:    $Q_{next} = Q_{supported}$ ;
8: endif
9: while ( $U < C_d$ ) // Check the CPU Utilization that can decode the selected chunk
10:    $Q_{next} = Q_{next} - I$ ;
11: end while
12: while ( $D_t < V_p$ ) // Battery drain time should be more than the time to play the video
13:    $Q_{next} = Q_{next} - I$ ;
14: end while
15: if ( $Q_{next} < I$ ) // I is the minimum available quality level
16:   Pause the Video
17: else
18:   Return  $Q_{next}$  and  $R_c$ 
19: endif

```

### 3.3 Media Stream Selection

The MSC changes the sending stream to the receiver based on the feedback from Media Quality Monitor. MSC is responsible for fetching contents from media server and delivery to the mobile client.

Based on device resolution class and quality level, the MQM module controls the multimedia stream by calculating the actual sending rate and desired stream rate to the receiver. The MQM module selects the most suitable stream for delivery to the mobile client. The stream is updated based on the device class and quality level at the output. We take device resolution class and quality level, calculate desired stream rate for the device and match the desired media stream with actual sending stream. If the sending stream matches to the desired stream, then keep sending the stream. Otherwise, update the sending stream based on the new desired rate calculated from the network environment.

**Algorithm 2: Media Stream Selection algorithm**

**Input:** Quality level and resolution class

**Output:** The stream is selected for next segment delivery

```

1:  Get_Desired_Stream (device resolution class, quality level); // quality level received from
2:    // client using algorithm 1
3:  If (Desired_media_stream == current_sending_stream)
4:    Return current_sending_stream
5:  else
6:    current_sending_stream = Desired_media_stream;
7:    Return current_sending_stream;
8:  endif

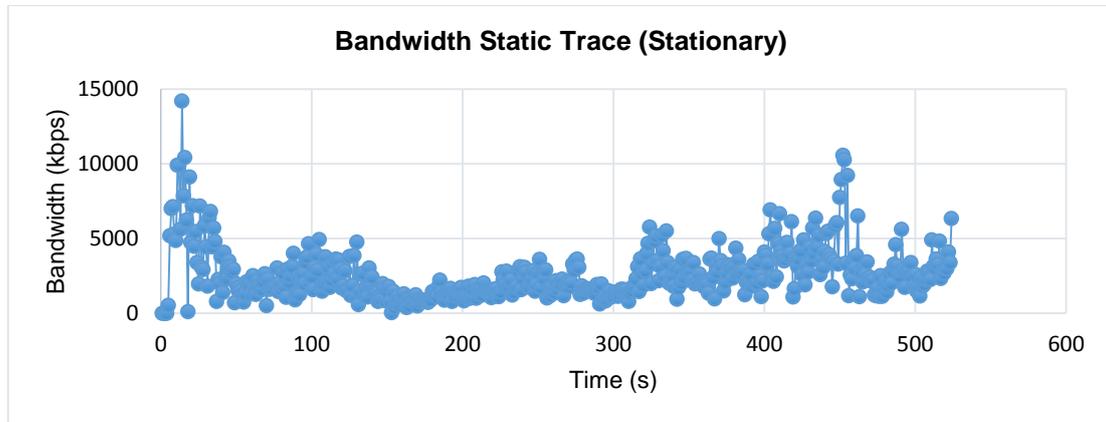
```

## 4. Experimental Evaluation

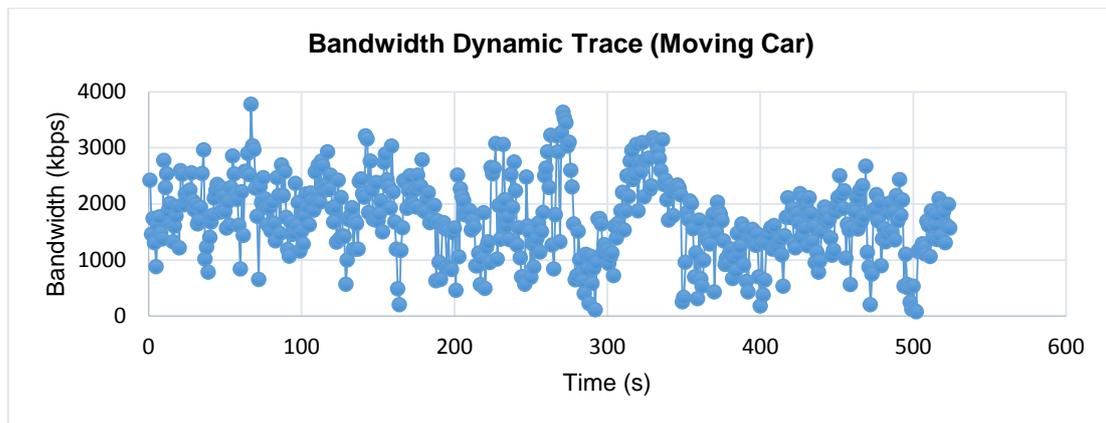
We modified dash.js (version 2.9) [37] to implement QOMCDS and existing ABR algorithms. DASH-JS is a smooth integration of DASH standard into the Web by adopting the video element of HTML5. This is developed on JavaScript that uses the Media Source API of Google's Chrome browser. This offers a flexible and most likely browser independent DASH player. DASH-JS is recently using media chunk of WebM and ISO Base format Media File.

The dash.js was configured to get stream based on the bitrate selection from a client. The playback buffer capacity of DASH player was configured for 60 seconds. The server machine used for experiments was XEON quad core 3.7 GHz Intel processor with 32 GB RAM. The APACHE Web Server was used for testing. We used the Samsung Galaxy S6 with Android version 5.0.2 for the client side to play videos for simulations. The Google Chrome browser was used to run simulations on the mobile device.

We used 4G LTE mobile dataset [38] for static bandwidth trace and 3G/HSDPA mobile dataset [39] for dynamic bandwidth trace. The dataset for the static trace is taken when the mobile user is stationary while playing a video and bandwidth trace is shown in Fig. 2. The dataset selected for the dynamic trace is the one taken while playing a video in a moving car and bandwidth trace is shown in Fig. 3. The bandwidth trace of the first 500 seconds is considered for our experiments.



**Fig. 2.** 4G LTE Dataset Bandwidth Trace in Static Scenario



**Fig. 3.** HSDPA Bandwidth Trace in Dynamic Scenario for Moving Car

To emulate network behavior similar to bandwidth logs using adaptive HTTP streaming, we used a bandwidth throttle module to play bandwidth trace using Apache Web server. This means that we can limit the bandwidth according to measured bandwidth logs for static and dynamic network condition.

We used the objective video quality assessment models and QoE feature metrics to measure the quality of video closer to the human visual assessment. The QOMCDS is evaluated against renowned bitrate adaptation schemes such as AIMD, BBA, Elastic, QDASH, Rate-based, FESTIVE and M3QD which we implemented within our player as an extension. The comparison is performed based on a performance metric that has a large impact on viewer experience. The objective Video Quality Assessment (VQA) models such as PSNR, VQM and SSIM are used for evaluation. The standard client side QoE metric such as Startup delay, Rebuffering events, and Bitrate switch count is used to evaluate QoE for the mobile user. The Startup delay is measured in seconds.

#### 4. 1 Experimental Setup

A video dataset Waterloo Streaming QoE Database III (SQoEIII) [40] is used for the evaluation purpose. The 3 video clips are selected from the dataset for quality assessment of mobile user as shown in Table 3. Each of the video clip represents different temporal and spatial motion contents such as high motion, smooth motion, and static motion. The duration of each video clip is 500 seconds. Each of the videos is encoded by FFmpeg [41] at six different bitrates. These video rates include 235, 375, 750, 1050, 1750, and 3000kbps.

We played 300 chunks for each video and the duration of one chunk is 2 seconds that are encoded in six distinct quality levels. We performed two experiments: one with static bandwidth trace and the other with dynamic bandwidth trace. The experiment to measure video quality for each metrics is repeated for three videos. The average results of three videos are used to cover videos in different scenarios to evaluate our proposed method. The total bitrate switch count is defined as the number of times bitrate quality level change occurs during the experiment. At the start of each experiment, the lowest bitrate level is set to be sent to the server for chunk delivery. Table 3 shows the spatial information, temporal information and frame rate of the selected videos.

**Table 3.** Spatial Information(SI), Temporal Information(TI), Frame rate(FPS), and Description of Reference Videos [40]

Video ID	FPS	SI	TI	Description
BigBuckBunny	30	96	97	Animation, high motion
RushHour	30	52	20	Human, smooth motion
TallBuildings	30	81	13	Architecture, static

The MSU Video Quality Measurement Tool [42] is used to measure PSNR, SSIM, and VQM. In total 6 tests were performed using 3 video clips for each scheme in a static and dynamic network. Fig. 4 to Fig. 9 shows a comparison between the proposed scheme QOMCDS and existing techniques under static and dynamic scenarios for a mobile user.

#### 4.2 Results and Discussion

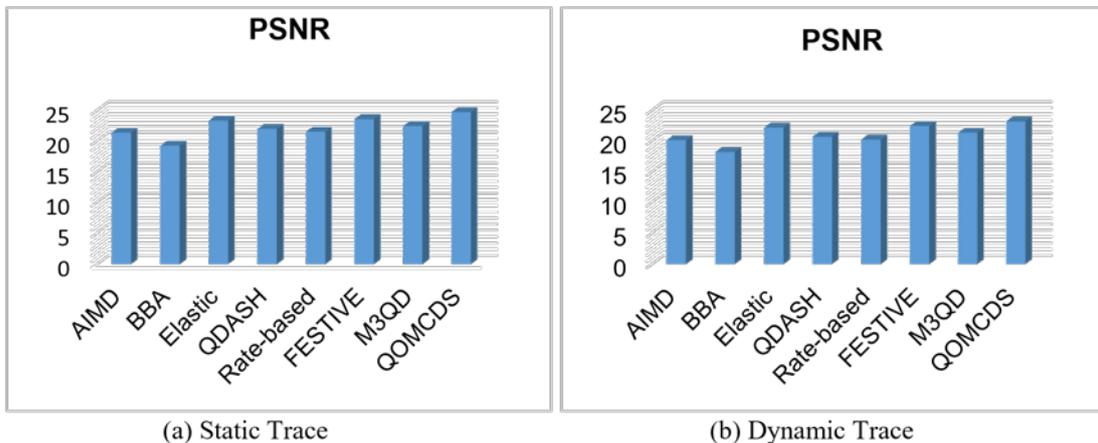
The performance analysis is performed at variable network bandwidth to evaluate the proposed QOMCDS model. Our proposed solution targets smartphones and tablets. Objective video quality assessment models are mathematical models to evaluate the quality of the video contents. There is no single metric that can measure accurate user perceived video quality. We used three metrics to have more accurate results. The higher PSNR and SSIM show better results. The lower VQM value gives better results. For simulations, parameter V is set to 15 to keep the queues stable.

Table 4 summarizes the evaluation of results and shows the values of all metrics against all models. Our proposed model shows the highest values of PSNR; 24.73, VQM; 3.91, SSIM; 0.91. The PSNR and SSIM of QOMCDS are higher than the AIMD, BBA, Elastic, QDASH, Rate-based, FESTIVE and M3QD. The results show that QOMCDS maintains high video quality in a static environment. In a dynamic environment, the approach tries to maintain multimedia quality effectively. This scenario shows the typical use of smart devices.

**Table 4.** Video Quality Assessment results

Method/Metrics	Static Trace			Dynamic Trace		
	PSNR	VQM	SSIM	PSNR	VQM	SSIM
AIMD	21.33	7.71	0.59	20.11	6.88	0.54
BBA	19.21	5.22	0.61	18.19	5.18	0.56
Elastic	23.33	5.34	0.83	22.21	5.23	0.75
QDASH	21.97	4.99	0.8	20.67	4.81	0.74
Rate-based	21.52	4.97	0.81	20.21	4.79	0.76
FESTIVE	23.60	5.40	0.85	22.45	5.31	0.76
M3QD	22.45	5.32	0.76	21.33	5.11	0.73
QOMCDS	24.73	3.91	0.91	23.21	3.61	0.82

**Fig. 4, Fig. 5 and Fig. 6** compare the results of all objective video quality assessment models with our proposed model QOMCDS. The average PSNR, VQM and SSIM values for streaming 3 video clip in both static and dynamic scenarios are calculated for evaluation. QOMCDS shows the value of PSNR around 24.73 dB for a static network that is a better score for video transmissions through wireless channels and BBA presents lowest PSNR value around 19.21dB. VQM has achieved 21.32% improvement over rate based approach in a static network. VQM has achieved 49 % improvement over AIMD approach in a static network. SSIM metric shows 6.5% better performance using QOMCDS over FESTIVE approach in a static environment. SSIM metric shows 35% better performance using QOMCDS over AIMD approach in a static environment. It's clear from the results that QOMCDS outperforms existing algorithms in static and dynamic network scenarios for all video quality assessment metrics.

**Fig. 4.** Video Quality Assessment using PSNR

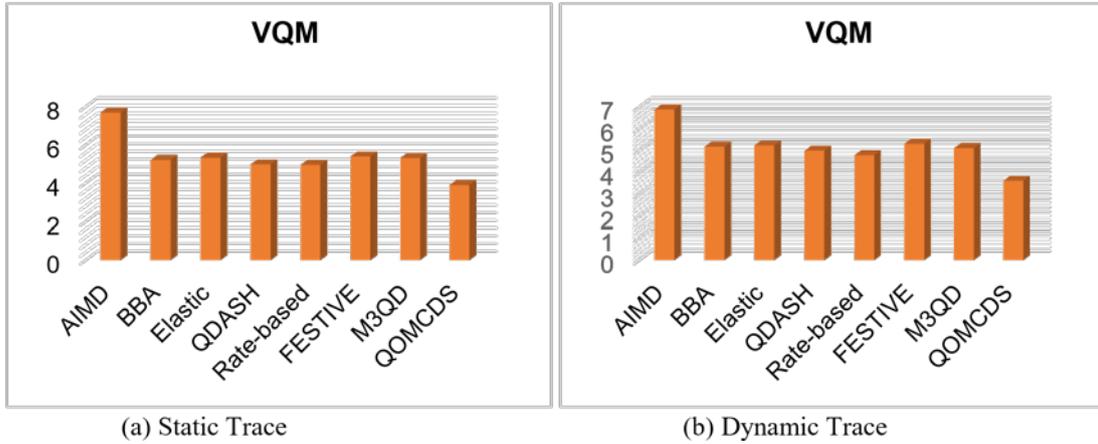


Fig. 5. Video Quality Assessment using VQM

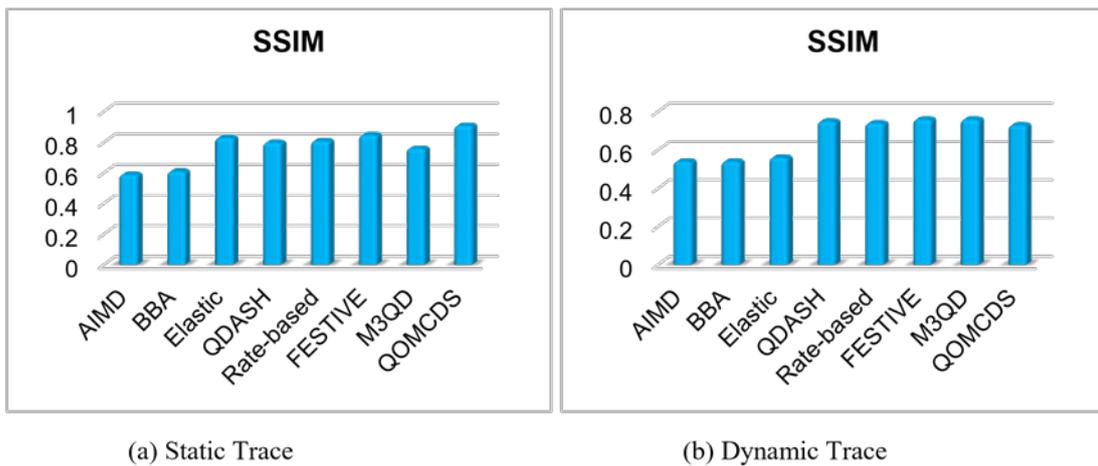


Fig. 6. Video Quality Assessment using SSIM

Fig. 6 shows the average SSIM over the static trace and dynamic trace. QOMCDS is able to maintain an average SSIM above 0.82 for static and dynamic network scenario. It performs better than Rate-based heuristic and FESTIVE. It is shown that FESTIVE results in higher SSIM as compared to the rate-based heuristic. The results confirm that the proposed algorithm achieves a higher SSIM. The QOMCDS outperform FESTIVE, Rate-based, BBA and QDASH algorithms in static and dynamic network scenarios.

The average quality metric for Startup delay, Rebuffering events and Bitrate switch count shown in Table 5. The results in Table 5 are taken using real traces of 3G/HSDPA and LTE dataset.

**Table 5.** Client Side QoE Metric results

Method/Metrics	Static Trace			Dynamic Trace		
	Startup Delay	Rebuffering Events	Bitrate Switch Count	Startup Delay	Rebuffering Events	Bitrate Switch Count
AIMD	6.2	12	41	6.9	18	66
BBA	6.6	5	32	6.9	7	55
Elastic	4.5	8	34	4.9	11	58
QDASH	12.2	10	33	13.2	16	59
Rate-based	3.3	11	31	4.5	18	53
FESTIVE	4.52	7	29	4.98	11	52
M3QD	11.43	11	36	14.28	15	61
QOMCDS	2.61	4	22	2.92	7	42

We evaluate the video quality in terms of Startup delay, Rebuffering events, and Bitrate switch count. The Fig. 7 to Fig. 9 shows a graphical representation of each quality metric. The startup delay shows 78 % improvement as compared to QDASH and 20.9 % improvement over rate-based in a static network. The QDASH shows the highest startup delay with 12.2 seconds. The test shows bitrate switch count 46.3 % improvement as compared to AIMD and 38% improvement over M3QD in a static network. The AIMD shows the highest bitrate oscillations with 41. The Rebuffering events show 66.7% improvement as compared to AIMD and 20% improvement over BBA in a static network. The AIMD shows the highest Rebuffering events with 12. We observed that QOMCDS achieve the highest stability as compared to existing adaptation techniques. The QOMCDS shows infrequent quality oscillations and stable throughput which has better viewer experience.

Fig. 8 plots the average number of Rebuffering events for the played videos. As the buffer based algorithm decides the next videos based on the buffer occupancy, we can expect the rate to fluctuate as the buffer occupancy changes. The rate based algorithm changes the level when the network bandwidth varies which results in fluctuations. Our algorithm takes frequency of rate switching into consideration when deciding the next quality level to reduce rate oscillations. Fig. 9 shows the average number of switches for the proposed and considered algorithms. ELASTIC almost produces the same number of switches per video flow in all considered algorithms. On average 34 and 58 video switches are observed for each video in static and dynamic trace respectively. The number of level switches for QOMCDS is 22 in a static network. In a dynamic network, the QOMCDS shows 42 bitrate switch count and AIMD shows 66 bitrate switch count. We can see that the total switch count for QOMCDS is always lower than that of other techniques. The results indicate that stable bitrate is experienced by the users using QOMCDS. The Startup delay, Rebuffering

events, and Bitrate switch value shows QOMCDS outperform the existing schemes. AIMD performs worst for rate switch metric as compared to QOMCDS.

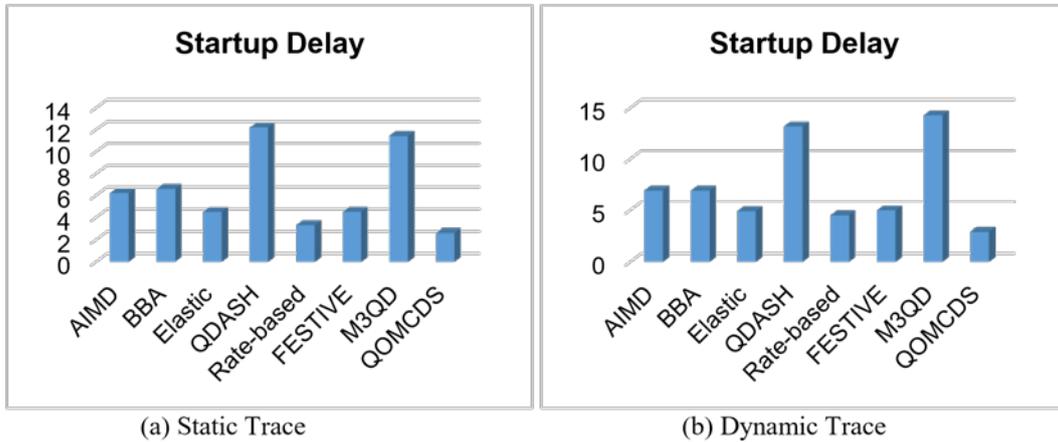


Fig. 7. Startup Delay

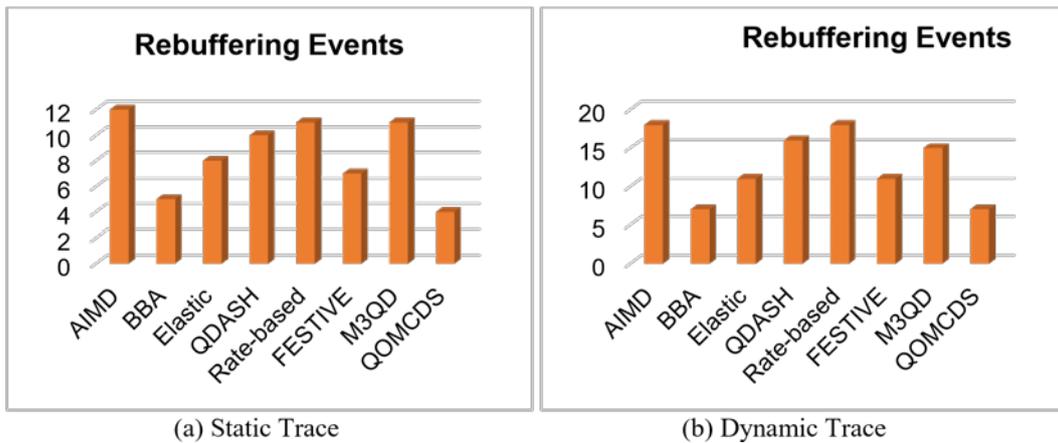


Fig. 8. Rebuffering Events

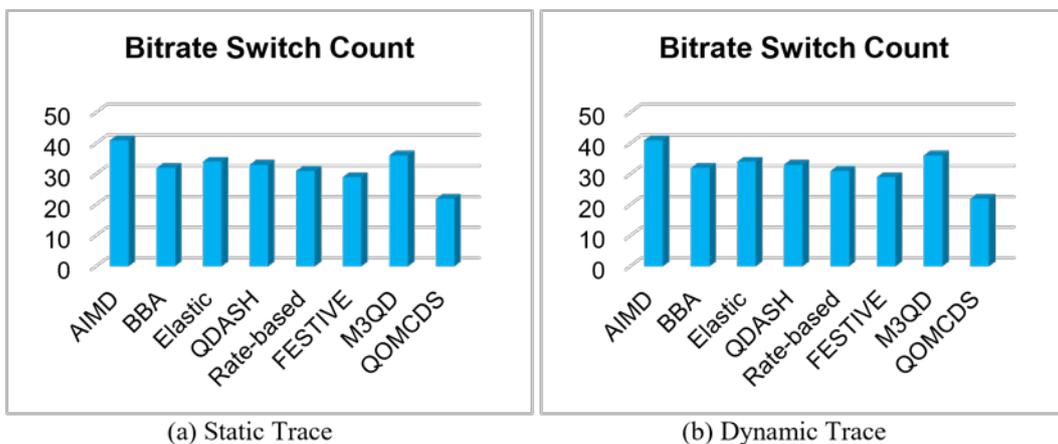


Fig. 9. Bitrate Switch Count

**Fig. 9** shows the results for bitrate oscillations. Buffer-based does not perform well in this metric because it tries to increase throughput when the buffer level is insufficient, but it results in increased oscillations and hence the performance decreased. QOMCDS perform better than the Elastic in excessive rebuffering.

From experimental results, we can conclude that QOMCDS performs more consistently across static and dynamic network scenario in comparison to existing techniques. One reason for QOMCDS consistency is its few design parameters. It uses  $V$  as a design parameter, the appropriate value is chosen to reduce bitrate oscillations. Other algorithms use more parameters and tuning the parameters for all scenarios is challenging.

Rate-based does not consider buffer level in deciding the next video level so it is the least affected by buffer size. The performance of Rate-based decreases because of more bitrate levels. The reason is that Rate-based start changing bitrate levels more frequently when there is a change in network bandwidth, leading to increased bitrate instability and a decrease in QoE for the user.

Our algorithm shows almost similar values in terms of SSIM and VQM which shows its flexibility to video content delivery for a mobile user. Although AIMD performs worst in a dynamic environment multimedia content delivery and shows a slight improvement in a static environment. The QOMCDS is very consistent for delivering multimedia content in a static and dynamic environment.

Q-DASH performs better during ramp down network condition because it performs smooth quality degradation. It confirms that smooth transitions are preferred over abrupt transitions.

The Rate-based maintains a stable video quality but it faces rebuffering events due to optimistic throughput prediction. QOMCDS provide better stability and fewer rebuffering events. The QOMCDS performs better than M3QD and BBA with very fewer rebuffering events.

The quality variations for QOMCDS are fewer so it maintains a stable video quality. The rate based approach faces large quality variations which significantly affect the user QoE. QOMCDS performs significantly better than FESTIVE and Rate-based approach when the bandwidth variations are large.

The impact of mobility on the bitrate adaptation is shown in a dynamic network scenario. When the mobile user is moving in a car, the number of quality switches are increased. When the mobile user is stationary, a higher bitrate can be selected based on the available network bandwidth.

## 5. Conclusion

We have presented a QOMCDS scheme for a mobile user to enhance user quality of experience and effectively select multimedia stream. The QOMCDS scheme is evaluated using simulations. The performance is compared to AIMD, BBA, Elastic, QDASH, Rate-based, FESTIVE and M3QD by delivering multimedia content streams. The objective video quality methods PSNR, VQM and SSIM, and QoE metrics show improvement for QOMCDS. The results show significant improvement using the proposed scheme QOMCDS over existing models using objective quality measurement and QoE metrics.

## References

- [1] Fangming Liu *et al.*, “Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications,” *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–22, Jun. 2013. [Article \(CrossRef Link\)](#)
- [2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, Cisco, San Jose, CA, USA, 2021-2016.
- [3] L. Zhou, X. Wang, W. Tu, G. Muntean, and B. Geller, “Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 3, pp. 409–419, Apr. 2010. [Article \(CrossRef Link\)](#)
- [4] R. Trestian, I.-S. Comsa, and M. F. Tuysuz, “Seamless Multimedia Delivery Within a Heterogeneous Wireless Networks Environment: Are We There Yet?,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 945–977, 2018. [Article \(CrossRef Link\)](#)
- [5] M. Aguayo, L. Bellido, C. M. Lentisco, and E. Pastor, “DASH Adaptation Algorithm Based on Adaptive Forgetting Factor Estimation,” *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1224–1232, May 2018. [Article \(CrossRef Link\)](#)
- [6] H.-B. Chang, I. Rubin, S. Colonnese, F. Cuomo, and O. Hadar, “Joint Adaptive Rate and Scheduling for Unicasting Video Streams in Cellular Wireless Networks,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8398–8412, Sep. 2017. [Article \(CrossRef Link\)](#)
- [7] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, “SDNHAS: An SDN-Enabled Architecture to Optimize QoE in HTTP Adaptive Streaming,” *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2136–2151, Oct. 2017. [Article \(CrossRef Link\)](#)
- [8] H. Yuan, X. Wei, F. Yang, J. Xiao, and S. Kwong, “Cooperative Bargaining Game-Based Multiuser Bandwidth Allocation for Dynamic Adaptive Streaming Over HTTP,” *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 183–197, Jan. 2018. [Article \(CrossRef Link\)](#)
- [9] Waqas ur Rahman and Kwangsue Chung, “Buffer-Based Adaptive Bitrate Algorithm for Streaming over HTTP,” *KSII Transactions on Internet and Information Systems*, vol. 9, no. 11, pp. 4585–4603, Nov. 2015. [Article \(CrossRef Link\)](#)
- [10] Sunghye Lee and Kwangsue Chung, “A Multipath Congestion Control Scheme for High-Quality Multimedia Streaming,” *KSII Transactions on Internet and Information Systems*, vol. 11, no. 1, pp. 414-435, 2017. [Article \(CrossRef Link\)](#)
- [11] Sangwook Kim and Kwangsue Chung, “A Bandwidth Estimation Scheme to Improve the QoE of HTTP Adaptive Streaming in the Multiple Client Environment,” *KSII Transactions on Internet and Information Systems*, vol. 12, no. 1, pp. 308–324, Jan. 2018. [Article \(CrossRef Link\)](#)
- [12] A. Biernacki, “Improving Video Quality by Diversification of Adaptive Streaming Strategies,” *KSII Transactions on Internet and Information Systems*, vol. 11, no. 1, pp. 374–395, Jan. 2017. [Article \(CrossRef Link\)](#)
- [13] Longhao Zou, R. Trestian, and G.-M. Muntean, “DOAS: Device-Oriented Adaptive Multimedia Scheme for 3GPP LTE systems,” in *Proc. of 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, London, 2013, pp. 2180–2184. [Article \(CrossRef Link\)](#)
- [14] B. Ciubotaru, C. H. Muntean, and G.-M. Muntean, “Mobile Multi-Source High Quality Multimedia Delivery Scheme,” *IEEE Transactions on Broadcasting*, vol. 63, no. 2, pp. 391–403, Jun. 2017. [Article \(CrossRef Link\)](#)
- [15] C.-F. Lai, H. Wang, H.-C. Chao, and G. Nan, “A Network and Device Aware QoS Approach for Cloud-Based Mobile Streaming,” *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 747–757, Jun. 2013. [Article \(CrossRef Link\)](#)
- [16] Xiaoke Zhu, Xiao-Yuan Jing and Di Wu *et al.*, “Similarity-maintaining Privacy Preservation and Location-aware Low-rank Matrix Factorization for QoS Prediction based Web Service Recommendation,” *IEEE Transactions on Services Computing*, pp. 1–1, 2018. [Article \(CrossRef Link\)](#)

- [17] F. Sardis, G. Mapp, J. Loo, M. Aiash, and A. Vinel, "On the Investigation of Cloud-Based Mobile Media Environments With Service-Populating and QoS-Aware Mechanisms," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 769–777, Jun. 2013. [Article \(CrossRef Link\)](#)
- [18] K. T. Bagci, K. E. Sahin, and A. M. Tekalp, "Compete or Collaborate: Architectures for Collaborative DASH Video Over Future Networks," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2152–2165, Oct. 2017. [Article \(CrossRef Link\)](#)
- [19] G. Gao, Y. Wen, and J. Cai, "vCache: Supporting Cost-Efficient Adaptive Bitrate Streaming," *IEEE MultiMedia*, vol. 24, no. 3, pp. 19–27, 2017. [Article \(CrossRef Link\)](#)
- [20] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. of the second annual ACM conference on Multimedia systems - MMSys '11, San Jose, CA, USA*, pp. 157-168, 2011. [Article \(CrossRef Link\)](#)
- [21] N. Usha, "Mobile Applications: Delivery Technologies In Multimedia Cloud Computing," *International Journal of Engineering Research*, vol. 2, no. 1, p. 7, 2013.
- [22] K. Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A. Hansen, and P. Engelstad, "Improving the performance of quality-adaptive video streaming over multiple heterogeneous access networks," in *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11, San Jose, CA, USA*, pp. 57-68, 2011. [Article \(CrossRef Link\)](#)
- [23] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: evidence from a large video streaming service," in *Proc. of the 2014 ACM conference on SIGCOMM - SIGCOMM '14, Chicago, Illinois, USA*, pp. 187–198, 2014. [Article \(CrossRef Link\)](#)
- [24] 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 '11, San Jose, CA, USA*, pp. 169-174, 2011. [Article \(CrossRef Link\)](#)
- [25] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH)," in *Proc. of 2013 20th International Packet Video Workshop, San Jose, CA*, pp. 1–8, 2013. [Article \(CrossRef Link\)](#)
- [26] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326-340, 2014. [Article \(CrossRef Link\)](#)
- [27] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: A QoE-aware DASH system," in *Proc. of the 3rd Multimedia Systems Conference*, pp. 11-22, 2012. [Article \(CrossRef Link\)](#)
- [28] "DASH Industry Forum | Catalyzing the adoption of MPEG-DASH." [Online]. Available: <https://dashif.org/>. [Accessed: 30-Dec-2018].
- [29] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in *Proc. of 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey*, pp. 2366–2369, 2010. [Article \(CrossRef Link\)](#)
- [30] M. Hammoudeh and R. Newman, "Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance," *Information Fusion*, vol. 22, pp. 3–15, Mar. 2015. [Article \(CrossRef Link\)](#)
- [31] Xiao-Yuan Jing, Xiaoke Zhu, Fei Wu and Ruimin Hu *et al.*, "Super-Resolution Person Re-Identification With Semi-Coupled Low-Rank Discriminant Dictionary Learning," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1363–1378, Mar. 2017. [Article \(CrossRef Link\)](#)
- [32] Seung-Bum Lee, G.-M. Muntean, and A. F. Smeaton, "Performance-Aware Replication of Distributed Pre-Recorded IPTV Content," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 516–526, Jun. 2009. [Article \(CrossRef Link\)](#)
- [33] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. of The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Pacific Grove, CA, USA, pp. 1398–1402, 2003. [Article \(CrossRef Link\)](#)

- [34] M. Zhong, P. Hu, and J. Indulska, "Revisited: Bandwidth estimation methods for mobile networks," in *Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, Sydney, Australia, pp. 1–6, 2014. [Article \(CrossRef Link\)](#)
- [35] "Battery Status API." [Online]. Available: <https://www.w3.org/TR/battery-status/>. [Accessed: 27-Jan-2019].
- [36] Michael J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems Synthesis Lectures on Communication Networks*, Communication Networks, 2010. [Article \(CrossRef Link\)](#)
- [37] Akamai. 2016. *dash.js*, Dash Industry Forum, 2018.
- [38] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond throughput: a 4G LTE dataset with channel and context metrics," in *Proc. of the 9th ACM Multimedia Systems Conference - MMSys '18, Amsterdam, Netherlands*, pp. 460–465, 2018. [Article \(CrossRef Link\)](#)
- [39] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications," in *Proc. of the 4th ACM Multimedia Systems Conference*, pp. 114–118, 2013. [Article \(CrossRef Link\)](#)
- [40] Z. Duanmu, A. Rehman, and Z. Wang, "A Quality-of-Experience Database for Adaptive Video Streaming," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 474–487, Jun. 2018. [Article \(CrossRef Link\)](#)
- [41] "FFmpeg," [Online]. Available: <https://www.ffmpeg.org/>. [Accessed: 16-Jan-2019].
- [42] "MSU Video Quality Measurement Tool: Download Page," [Online]. Available: [http://www.compression.ru/video/quality\\_measure/vqmt\\_download.html](http://www.compression.ru/video/quality_measure/vqmt_download.html). [Accessed: 17-Feb-2019].



**MUHAMMAD SALEEM** was born in Lahore, Pakistan, 1982. He received the B.S. and M.S. degrees in Computer engineering from the University of Engineering and Technology, Lahore in 2012 and currently enrolled in Ph.D. degree program in Computer engineering from University of engineering and Technology Lahore, Pakistan. His research interests include multimedia streaming, Internet of Things, mobile cloud computing and embedded system.



**YASIR SALEEM** is currently serving as an Associate Professor in University of engineering and Technology (UET) Lahore, Pakistan. His research interest includes Computer networks, cloud Computing and Embedded System. He completed his secondary education from England. He had achieved his B.S., M.S., and Ph.D. degree from Electrical Engineering department of UET in 2002, 2004 and 2011 respectively. He has authored and co-authored journal and conference papers at national and international level in field of Computer Science and Engineering.