Analysis of QoE for Adaptive Video Streaming over Wireless Networks with User Abandonment Behavior

Rachid El-Azouzi*, Krishna V Acharya†, Sudheer Poojary*, Albert Sunny‡
Majed Haddad*, Eitan Altman§ and Dimitrios Tsilimantos¶ and Stefan Valentin‖
*CERI/LIA, University of Avignon, Avignon, France; §INRIA, Sophia Antipolis, France
‡Indian Institute of Technology, Palakkad, India; †BITS Pilani KK Birla Goa Campus, India
¶Lab France Research Center, Huawei Technologies Co. Ltd. France
‖Darmstadt University, Darmstadt, Hessen, Germany

***Abstract*—In this paper, we develop an analytical framework to compute the Quality-of-Experience (QoE) metrics of video streaming in wireless networks. Our framework takes into account the system dynamics that arises due to the arrival and departure of flows. We also consider the possibility of users abandoning the system on account of poor QoE. Considering the coexistence of multiple services such as video streaming and elastic flows, we use a Markov chain based analysis to compute the user QoE metrics: probability of starvation, prefetching delay, average video quality and bitrate switching. Our simulation results validate the accuracy of our model and describe the impact of scheduler at eNB on the QoE metrics.**

I. Introduction

The proliferation of high-definition ephemeral video content has made video streaming the dominant contributor of modern-day mobile traffic. A recent study claims that video traffic (e.g., TV videos, video streaming, live video services) will represent 90% of the Internet traffic by 2021 [1]. While significant progress has been made in recent years towards increasing the capacity of cellular networks, users' *Quality of Experience (QoE)* has become a challenging and prominent issue. To efficiently trade-off network performance and QoE, researchers are exploring *HTTP Adaptive Streaming (HAS)* enabled architectures [2]–[4]. *Dynamic Adaptive Streaming over HTTP (MPEG-DASH)* has become a very popular HAS standard in recent years. With DASH, videos are streamed over HTTP from a video server where multiple versions of the source video are pre-encoded at different bitrates. Due to the highly variable nature of wireless channels, video interruptions are more likely to occur when the video quality is kept constant throughout the duration of the video session. The key concept in DASH is to dynamically adapt the video quality to match the network bandwidth. This, in turn, will reduce the number of playback interruptions. The design goal of DASH is to simultaneously obtain superior performance over different key metrics such as buffering delay, playback interruptions, average bitrate (video quality), and temporal variability of streaming quality. However, in an environment with highly variable throughput, simultaneously attaining good performance across all these metrics is an extremely challenging task.

While the quality of experience is a subjective quantity, there are metrics which help us quantify a user's QoE. The probability of starvation (stall/rebuffering) and the average video quality are obvious QoE metrics. Video clients typically prefetch some content before playout and also wait before playout after a starvation event. This causes a delay before playout which affects user experience. The average initial startup delay and rebuffering delay (delay post a starvation event) quantify this aspect of user QoE. With adaptive streaming, the user's video quality could change over time. Frequent quality switches are found to be detrimental to the user's QoE. Thus, the frequency of quality switching is another QoE metric.

Recent works have developed approaches to understand how some metrics such as probability of starvation, start-up delay, video quality and quality switches, influence the users' abandonment rate. Different metrics vary in the degree in which they impact the user abandonment rate. Authors in [5]–[7] showed that the time spent on rebuffering during a video session can significantly reduce the user abandonment rate. They found that the percentage of time spent in buffering (buffering ratio) has the largest impact on the user engagement across all types of content. For example, a 1% increase in buffering ratio can reduce the audience retention by more than *three minutes for a 90-minute video*. The impact of video quality on user engagement was investigated in [5], where it was found that watching time decreased between 1 and 3 minutes for every 1% increase in the buffering time.

A. Related work

Due to the increasing popularity of streaming video over wireless networks, there is a need to understand, both qualitatively and quantitatively, QoE performance metrics such as buffering delay, playback interruptions, average bitrate (video quality), and temporal variability of streaming quality. QoE issues, in different context, has been studied earlier [8], [9]. From the network operators' perspective, the ability to predict QoE of adaptive video flows is crucial towards designing efficient resource allocation mechanism. Network operators typically classify services as either real-time or elastic traffic to guarantee their quality of service requirements. However, they face difficulty when it comes to video streaming services because these services encompass attributes of both real-time and elastic traffic. In [10], the authors have modeled the

starvation probability and the moment generating function of starvation events in the scenario where the bandwidth is evenly shared by competing video streams. This is one of the earliest work that tries to resolve this difficulty. However, these models are more suitable to study non-adaptive video streaming traffic. In [11], the authors formulate the QoE maximization as an optimization problem and provide optimal rate allocation for video streaming in a wireless network considering user dynamics. The integration of elastic and adaptive streaming services was considered in [12] where performance bounds for, three key performance indicators, mean video bit rate, deficit rate and buffer surplus were presented. In [13], the authors analytically quantify a few important QoE metrics of adaptive video streaming in the presence of elastic flows, and arrival and departure of flows.

B. Contribution and organisation

We develop a flow-level model for the performance of adaptive streaming with flow arrival and departures. We extend this model to scenarios where users may leave the system before finishing their service. This can happen when the user's video stream experiences starvation, large start-up delay or poor quality. More specifically, we present an analytical framework to evaluate the impact of user leaving on account of poor QoE (balking). We derive approximations for the QoE metrics such as probability of starvation, average prefetching delay, average video bitrate and rate of bitrate switching accounting for user balking. Then, we go on to investigate the impact of schedulers, in particular, *proportional fairness scheduler* [14] and *D-VIEWS* [15] under user dynamics. Our theoretical study reveals that *D-VIEWS* scheduler is able to achieve a significant reduction of bitrate video switching while efficiently utilizing the network resources under dynamic flows and mobility.

The remainder of the paper is organized as follows: in section II we present the system model. Generic stability conditions and the Markov model of our system are presented in section III and IV, respectively. Analytical expressions for the various QoE metrics are presented in section V. Result of extensive simulations that were conducted to validate our analysis for *Proportional Fairness (PF) scheduler* as well as *D-VIEWS schedulers* are presented in section VI. Finally, in section VII, we conclude the paper.

II. System model

A. System description

We consider co-existing adaptive video streams and non-adaptive traffic over a single-cell network with a population of mobile users. With adaptive video streaming, each video is divided into multiple segments, and each segment is encoded into multiple bitrates/resolutions at the server. Let $\mathcal{L} = \{\ell_1 < \ell_2 < \cdots < \ell_m\}$ denote the set of video bitrates supported by adaptive video streams.

We assume that there are K different user classes, and each class represents a category of statistically identical users in term of flow sizes, arrival rates and channel statistics. Flows of class k arrive at a rate λ_k and have sizes that follow a given distribution function $D_k(.)$, with mean $1/\theta_k$. *For adaptive video streams and non-adaptive traffic, the flow size are presented in seconds and bits, respectively*. The arrival processes are assumed to be *independent Poisson processes*. Thus, the arrival rate of a flow is given as $\lambda = \lambda_1 + \cdots + \lambda_K$. Let $\mathcal{K}_D$ and $\mathcal{K}_E$ be the set of user classes for adaptive video flows and non adaptive flows, respectively.

We also consider the scenario where streaming users, due to poor video quality, could leave the system before their video stream has been completed [5], [7]. Thus, a departure from the system could be due to a user finishing service or a user leaving before its service finishes (user abandonment). Denote by $X = (n_1, \ldots, n_K)$ the state space of our system, where n_k denotes the number of class-k flows in the system.

B. LTE eNB scheduler

Current cellular base stations incorporate sophisticated radio resource management techniques for flow scheduling. In this section, we briefly present the well-known and popular *utility-based scheduling policies*. In such policies, a utility is attributed to the users' average throughput. In this paper, we restrict ourselves to the well-known class of fairness measure called α-fairness [16]. Specifically, we have

$$u_i(r_i) = \begin{cases} \log r_i & \text{if } \alpha_i = 1 \\ \frac{{r_i}^{1-\alpha_i}}{1-\alpha_i} & \text{otherwise} \end{cases}$$

where r_i is user i's long-term average throughput. We recall that α-fairness is a general fairness measure that satisfies the four axioms from [14]. If $\alpha_i = 1$, α-fairness becomes the well-known *proportional fairness*; when all $\alpha_i \to \infty$, it becomes *max-min fairness*.

In [15], the authors designed D-VIEWS scheduler which enforces bitrate stability for each adaptive video stream. D-VIEWS allocates radio resources to each video streaming flow based on its current channel state and the set of available video bitrates. This set is predefined by the streaming service and is usually fixed. The main goal of D-VIEWS was to ensure that the average throughput attained by the adaptive video streams takes values only in the set of the bitrates available at the server. This assures video bitrate stability, while efficiently utilizing the network resources. D-VIEWS only needs to be aware of the peak video bitrates and requires no changes to the streaming client and other network functions.

III. Generic stability conditions

In this section, we study stability conditions under a given scheduler. We start our analysis by considering the case where only adaptive video flows are present in the system and all classes have identical channel statistics. Let R_D be the average throughput when there is a single user in the system. We define $\rho_k = \frac{\lambda_k}{\theta_k}$; it represents the video duration coming into the system from class k, also referred to as traffic intensity of adaptive video flows from class k. $\rho_D = \sum_{k \in \mathcal{K}_D} \rho_i$ is then the total traffic intensity of adaptive video flows.

Under a given scheduler, let $G(n)$ be the gain in throughput in comparison to a channel oblivious round-robin scheduling. Since the relative scheduling gains are identical for all classes then the throughput received by an adaptive flow is $r(\vec{n}) = \frac{R_D G(n)}{n}$. This gain function will be increasing, reflecting the fact that the total throughput gains increase with the degree of multi-user diversity. It has its limit $G^* = \lim_{n \to \infty} G(n)$.

If the system is in state $\vec{n}$ then between t and $t + \delta t$ each user obtains an amount of service equal to $\frac{R_D G(n)}{n l(\vec{n})} \delta t$. Thus the remaining service (in seconds) of an adaptive video flow is reduced at rate $\frac{R_D G(n)}{n \ell(\vec{n})}$, which means that the remaining service requirements evolve in a similar probabilistic fashion as the remaining service requirements in a *multi-class Processor-Sharing (PS)* system with arrival rates λ_k, generic service

requirements θ_k , and service rate $\frac{R_D G(n)}{n\ell(\vec{n})}$. Based on Cohen's paper [17], the stability condition for such a system is given by

$$\rho_D < \frac{R_D G^*}{\ell_1} \text{ or } \frac{\rho_D \ell_1}{R_D G^*} < 1 \tag{1}$$

Furthermore, under this stability condition, the probability of having n users in the system is

$$P(X=n) = c\Big(\frac{\rho_D}{R_D}\Big)^n \Big\{\prod_{j=1}^{n} \frac{G(j)}{\ell(j)}\Big\}^{-1} \tag{2}$$

where c is a normalization constant. Using *Little's law*, we get the expected time in the system for a user of class k as

$$\mathbb{E}[S_k] = \frac{\mathbb{E}(n)}{\theta_k \rho_D} \tag{3}$$

Let us now consider a scenario where both adaptive video and other flows are sharing the capacity. In such a setting, the process sharing equivalence is not valid directly and the calculation of stationary distribution needs additional assumptions on the service time. So, we study the stability condition of the system under general conditions. Here each class represents a category of statistically identical users in term of flow sizes, arrival rates and channel statistics. Let R_k be the average when there is a single user of class-k in the system.

For adaptive streaming users, the video bitrate decreases as the number of active flows increases. In fact, when the system approaches the stability limit, the adaptive video flows uses minimum quality l_1. Hence, stability is guaranteed when the sum of offered loads for all classes is less than 1, i.e.,

$$\sum\nolimits_{k\in\mathcal{K}_D} \frac{\rho_k \ell_1}{R_k G^*} + \sum\nolimits_{k\in\mathcal{K}_E} \frac{\rho_k}{R_k G^*} < 1 \tag{4}$$

IV. Markov model

For ease of presentation, we just consider just two classes of traffic: adaptive video flows (class-1), and non-adaptive video traffic (class-2)[1].

A. Markov model for the system dynamics

Under a given scheduler[2], (e.g., proportional fairness or D-VIEWS), let $r_j(\vec{n})$ be the average rate of a user of class-$j \in \{1,2\}$. Here, $\vec{n} = (n_1, n_2)$, and n_k is the number of users in class-k. Since adaptive video streaming protocols, such as DASH, tend to match the video bitrate to the available channel rate, the user's video bitrate of class-1 depends only on the state of the system $\vec{n}(t)$. Let $\ell(\vec{n})$ denote the video bit-rate of a user of class-1 when the system in state $\vec{n}$.

Given the assumption of exponential distributed video size, the service time of a mobile user is also exponentially distributed. This implies that the departure of a mobile users given the current state at time t is independent of the past. We assume that the times till departure of the different users, due to '*impatience*', are independent exponentially distributed random variables. Under the above mentioned assumptions, the dynamics of coexisting mobile users in the cell can be depicted as a continuous time Markov chain (CTMC) with state space $\mathcal{X}$.

The slots for the scheduler at the base station are in *milliseconds*, whereas the video segment playout happens in *seconds*. The timescale of the user arrival and departure process is in *hundreds of seconds*. Thus, the scheduler dynamics happen at a much faster time scale than the video segment playout and the user arrival and departure dynamics. Due to this ***timescale separation***, the slot-wise variations in the channel rate and users' average channel rate between the state transitions are negligible. Thus, we can then assume that, when the system state is $\vec{n}$,

- the average channel rate of a user of class-j is $r_j(\vec{n})$ and
- the average video bit-rate for video streaming (class-1) is given by $\ell(\vec{n})$.

Let us denote the rate transition matrix for the CTMC, $\{\vec{n}(t) \in \mathcal{X}\}$ by $\mathbf{Q}$. Let us denote by $\{e_j, j = 1,2\}$, the standard basis for $\mathbb{N}^2$, with e_j a unit vector with 1 in the j^{th} position. Now, $\mathbf{Q}(\vec{n}, \vec{n}+e_j) = \lambda_j$ (λ_j is the arrival rate of class-j flows) and $\mathbf{Q}(\vec{n}, \vec{n}-e_j) = \nu_j(\vec{n})$ where $\nu_j(\vec{n})$ is

$$\nu_1(\vec{n}) = \frac{n_1 r_1 \theta_1}{\ell(\vec{n})} + n_1\gamma(\vec{n}) \tag{5}$$

$$\nu_2(\vec{n}) = n_2 r_2 \theta_2 \tag{6}$$

where $\gamma(\vec{n})$ is the departure rate of adaptive video flows (class-1 users) due of poor QoE in state $\vec{n}$, and $1/\theta_j$ mean size of the flows in class-$j \in \{1,2\}$. The first term in the above equations corresponds to departures due to users finishing service, whereas the second term in eq. 5 for video streaming, i.e., class-1, corresponds to users leaving system before their video is downloaded. The CTMC, $\vec{n}(t)$ is a finite irreducible Markov chain and hence has a unique stationary distribution. Let us denote this distribution by $\boldsymbol{\pi} \triangleq \{\pi(\vec{n}) : \vec{n} \in \mathcal{X}\}$.

B. The tagged class-1 user Markov chain and probability of finishing

The QoE computation of the class-1 user, i.e., adaptive video flow, requires us to study the system dynamics as seen by a class-1 user in its sojourn in the system. This can be modeled as a CTMC with absorbing states. Let us denote by $Y_1(t)$ the state of the system as seen by a class-1 user. We note the state space of $\mathcal{Y}_1 = \{Y_1(t), t \geq 0\}$ same as that of $\mathcal{X}$ with the exception that $\mathcal{Y}_1$ has two absorbing states F and B corresponding to the user finishing service and the user leaving the system before its video is completely downloaded (balking). Let us denote the rate transition matrix for $Y_1(t)$ by $\mathbf{Q}_1$. As before, the transition rate of the for the arrival of a new user is $\mathbf{Q}_1(\vec{n}, \vec{n}+e_j) = \lambda_j$ for $j \in \{1,2\}$. The transition rate from state $\vec{n}$ to the absorbing states is given by

$$\mathbf{Q}_1(\vec{n}, F) = \frac{r_1(\vec{n}+e_1)\theta_1}{l(\vec{n}+e_1)} \quad \text{and} \quad \mathbf{Q}_1(\vec{n}, B) = \gamma(\vec{n}+e_1)$$

The transition rate from state $\vec{n}$ to $\vec{n}-e_j$ is given by

$$\mathbf{Q}_1(\vec{n}, \vec{n}-e_1) = n_1\mathbf{Q}_1(\vec{n}, F) + n_1\mathbf{Q}_1(\vec{n}, B) \tag{7}$$

$$\mathbf{Q}_1(\vec{n}, \vec{n}-e_2) = n_2 r_2(\vec{n}+e_1)\theta_2 \tag{8}$$

The additional term e_1 of the above equations accounts for the tagged user of class-1. Let $f(\vec{n}, F)$ be the probability of an adaptive video flow finishing its service starting out in state $\vec{n}$. Now, we use the standard method to compute the probability of

[1] We recall that each class of users represents a category of statistically identical users in term of flow sizes, arrival rates and channel statistics. We would like to remark that our analysis can be easily extended to scenarios with multiple classes.

[2] This study may use any scheduler at eNB but for simulation we use proportional fairness or D-VIEWS

landing in an absorbing state when the user enters the system in any transient state. We first rewrite the transition probability matrix in its canonical form.

$$\begin{array}{c} \\ \mathcal{A} \\ \mathcal{T} \end{array} \begin{array}{c} \begin{array}{cc} \mathcal{A} & \mathcal{T} \end{array} \\ \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{U} & \mathbf{T} \end{bmatrix} \end{array}$$

where $\mathcal{T}$ and $\mathcal{A}$ denote the set of transient and absorbing states, respectively. For our system we have two absorbing and multiple transient states. Thus, the probability of being absorbed in state $\vec{j}$ when the user joins the network at state $\vec{i}$ is given by $f(\vec{i},\vec{j})$ which is the i,j th entry of the matrix $\mathbf{F}=(\mathbf{I}-\mathbf{T})^{-1}\mathbf{U}$. In our case the $f(\vec{n},F)=\mathbf{F}_{\vec{n},1}$.

V. QoE METRICS

In this section we focus on the QoE of adaptive video flows, i.e, class-1 users. QoE metrics offer a means to describe, qualitatively and quantitatively, users' perception of the quality of a video stream. The following list includes few popular key performance indicators in adaptive video streaming services: pre-fetching delay, probability of starvation, duration of starvation, average video bitrate and rate of video bitrate switches. We note that the overall QoE for a user could possibly be quantified as some combination of the above metrics. We describe these metrics and their computation in subsequent sections. Besides this QoE metrics, a network operator would also be interested in the probability that the user leaves the system before in the middle of the stream.

We can compute the above metrics for users who finish service as well as for users who leave before finishing with our model. However, it seems reasonable to focus only on the QoE of users who have finished their service. Hence, for computation of the QoE metrics, we restrict our attention to only those users who finish service.

A. Mean start-up delay

A streaming client usually waits for the playout buffer to have some video segments before beginning playout. Also, if the playout buffer becomes empty during playout, the client needs to re-buffer, i.e., wait for some video segments, before it can resume playing the video. While this can lead to a delay in the video playout, it avoids potential recurrent starvation events which happen when the playout buffer is empty. The amount of video segments that the playout buffer needs to have before the client begins (or restart playout after starvation) is called the the *startup threshold.*

Let $q(t)$ denote the number of seconds in the playout buffer at time t. Assuming that there is no starvation in the interval $[t,t+h]$, r to be the average channel rate and ℓ the average video bitrate in $[t,t+h]$, we get

$$q(t+h)=\begin{cases} q(t)+rh/\ell, & \text{during prefetching,} \\ \max\{0,q(t)-h+rh/\ell\}, & \text{otherwise.} \end{cases} \quad (9)$$

Let $D^{q_a}(q,\vec{n})$ be the expected startup delay for a class-1 user, given the initial entry state $\vec{n}$, the current buffer duration q and the startup delay threshold q_a. We note that $D^{q_a}(q,\vec{n})=0$ for all $q \geq q_a$. Let $b(\vec{n})=\frac{r_1(\vec{n})}{l(\vec{n})}>0$. After an infinitesimal amount of time h, the queue length increases from q to $q+b(\vec{n})h$. This gives rise to the following dynamics of $D^{q_a}(q,\vec{n})$ for user class 1

$$\begin{aligned} D^{q_a}(q,\vec{n}) &= (1-\lambda h-\nu(\vec{n})h)(h+D^{q_a}(q+b(\vec{n})h,\vec{n})) \\ &+\sum\nolimits_{j=1}^{2}\lambda_j h\cdot D^{q_a}(q+b(\vec{n})h,\vec{n}+e_j) \\ &+\sum\nolimits_{j=1}^{2}\nu_j(\vec{n})h\cdot D^{q_a}(q+b(\vec{n})h,\vec{n}-e_j) \quad (10) \end{aligned}$$

where $\lambda=\lambda_1+\lambda_2$ and $\nu(\vec{n})=\nu_1(\vec{n})+\nu_2(\vec{n})$. Rearranging Eq. (10), and taking the limit $h\to 0$, we obtain the following set of ordinary differential equation

$$\begin{aligned} b(\vec{n})\frac{d}{dq}D^{q_a}(q,\vec{n}) &= -1+(\lambda+\nu(\vec{n}))d^{q_a}(q,\vec{n})) \\ &-\sum\nolimits_{j=1}^{2}(\lambda_j d^{q_a}(q,\vec{n}+e_j)+\nu_j(\vec{n})d^{q_a}(q,\vec{n}-e_j)) \quad (11) \end{aligned}$$

The boundary condition of $D^{q_a}(q,\vec{n})$ is given by $D^{q_a}(q_a,\vec{n})=0$. Let $\mathbf{D}^{q_a}(q)$ be a vector of the expected startup delay at different states of Markov chain for given q_a and q. Then, equation (11) can be rewritten as follows

$$\frac{d}{dq}\mathbf{D}^{q_a}(q)=\mathbf{B}(\mathbf{Q}\mathbf{D}^{q_a}(q)-\mathbf{1}) \quad (12)$$

where $\mathbf{B}$ is a diagonal matrix with $\frac{1}{b(\vec{n})}$ as its entries, and $\mathbf{1}$ is a column vector of all ones. The solution of the above ODE is as follows

$$\mathbf{D}^{q_a}(q)=e^{\mathbf{B}\mathbf{Q}q}C+\mathbf{Q}^{-1}\mathbf{1} \quad (13)$$

where C is the constant of integration, and can be computed using the boundary condition $\mathbf{D}^{q_a}(q_a)=\mathbf{0}$. Thus, we get

$$\mathbf{D}^{q_a}(q)=\left(\mathbf{I}-e^{\mathbf{B}\mathbf{Q}(q-q_a)}\right)\mathbf{Q}^{-1}\mathbf{1} \quad (14)$$

Now, the expected startup delay for a class-1 user entering the system (i.e., $q=0$) for different state of the system (i.e., $\{D^{q_a}(0,\vec{n}),\forall\vec{n}\}$) is given by

$$\mathbf{D}^{q_a}(0)=\left(\mathbf{I}-e^{-\mathbf{B}\mathbf{Q}q_a}\right)\mathbf{Q}^{-1}\mathbf{1} \quad (15)$$

Since the arrival process is *Poisson*, the probability that a class-1 user, upon arriving, sees the systems in state $\vec{n}$ is $\pi(\vec{n})$ by PASTA. Thus, we get

$$D^{q_a}(0)=\frac{(\boldsymbol{\pi}^T\mathbf{F})\mathbf{D}^{q_a}(0)}{(\boldsymbol{\pi}^T\mathbf{F})\mathbf{1}} \quad (16)$$

where $\mathbf{F}$ is a diagonal matrix with entries $f(\vec{n},F)$, and $\boldsymbol{\pi}$ is a row vector denote the stationary distribution of the system.

B. Probability of starvation

We assume that a class-1 user experiences starvation is system states $\vec{n}$ where the channel rate $r_1(\vec{n})$ is less than the lowest available video bit-rate (ℓ_{min}). Let $\mathcal{B}=\{\vec{n}\in\mathcal{X}: r_1(\vec{n}+e_1)<\ell_{min}\}$. We note that in practice, the user's playout buffer may have sufficient video segments so that the user visits and exits a state in set $\mathcal{B}$ before its playout buffer depletes. In such cases, the user may not experience starvation even though it visits the set $\mathcal{B}$ in its sojourn. Thus, our assumption gives us an upper bound on the probability of starvation. The bounds are very close to the actual values in the regime where video sizes and inter-arrival time of video requests are large.

For the embedded DTMC, $\{\hat{Y}(t),t\in\mathbb{N}\}$, we have two absorbing states, F and A. Let us consider a class-1 user

which enters the system in state $\vec{n}_0$ and finishes service at a random time T without aborting its service in the middle, i.e., its corresponding Markov chain hits F eventually. Then

$$P(\vec{n}) \triangleq \mathbb{P}\left\{\hat{Y}(t) \in \mathcal{B} \text{ for some } n | \hat{Y}(T) = F, \hat{Y}(0) = \vec{n}_0\right\}$$

gives us the probability that this user experiences starvation in its sojourn in the system. Let $P(\vec{n}, F) = \mathbb{P}\{\hat{Y}(t) \in \mathcal{B} \text{ for some } n, \hat{Y}(T) = F | \hat{Y}(0) = \vec{n}\}$. Define the taboo probability as

$$M^t(\vec{n}, \vec{m}) \triangleq \mathbb{P}\big\{\hat{Y}(t) = \vec{m}, \hat{Y}(t') \notin \mathcal{B}, \forall 0 < t' < t \\ | \hat{Y}(0) = \vec{n}\big\}$$

i.e., $M(\vec{n}_0, \vec{n})$ is the probability the chain $\{\hat{Y}(t), t \in \mathbb{N}\}$ moves from state $\vec{n}_0$ to state $\vec{n}$ in t steps without entering the taboo set B. Then, for $\vec{n} \in \mathcal{X} - \mathcal{B}$,

$$P(\vec{n}, F) = \sum_{t \in \mathbb{N}} \sum_{\vec{n}' \in \mathcal{X} - \mathcal{B}} M^t(\vec{n}, \vec{n}') \sum_{\vec{m} \in \mathcal{B}} M^1(\vec{n}', \vec{m}) f(\vec{m}, F)$$

This says that the probability the chain $\{\hat{Y}(t), t \in \mathbb{N}\}$ visits a state in $\mathcal{B}$, starting from state $\vec{n}$, in $t+1$ steps is equal to the probability that it moves from $\vec{n}$ to any state $\vec{n}' \in \mathcal{X} - \mathcal{B}$ in t steps, and then it moves from $\vec{n}'$ to $\vec{m}$ in 1 step. For $\vec{n} \in \mathcal{B}$, $P(\vec{n}) = 1$. Otherwise, $P(\vec{n}) = P(\vec{n}, F)/f(\vec{n}, F)$.

Thus, the unconditional probability of starvation for users of class-1 who finish service is given by

$$P = \frac{\sum_{\vec{n} \in \mathcal{X}} \pi(\vec{n}) f(\vec{n}, F) P(\vec{n})}{\sum_{\vec{n} \in \mathcal{X}} \pi(\vec{n}) f(\vec{n}, F)} \tag{17}$$

In the preceding computation, we assumed that if the tagged class-1 user visits a state in $\mathcal{B}$, it stays long enough in that state so that its playback buffer depletes to 0 before it exits the state. This ignores the possibility that there could be a transition from a state in $\mathcal{B}$ to $\mathcal{X} \setminus \mathcal{B}$ before the buffer depletes to 0, thus avoiding starvation. Thus P, as computed above, gives us an upper bound on the probability of starvation. Next we describe an approximation which takes into account this possibility.

Accounting for the effect of transitions: We consider two possibilities : (a) avoiding starvation due to prefetching and (b) tagged user avoiding starvation by exiting $\mathcal{B}$ before the buffer depletes to 0. With this, the probability of starvation for a class-1 user entering the system in state $\vec{n}$ and finishing service is given by

$$P(\vec{n}) = p(\vec{n}) f(\vec{n}, F) \\ + (1 - p(\vec{n})) \sum_{\vec{n}' \in \mathcal{X}} M(\vec{n}, \vec{n}') \hat{P}(\vec{n}, \vec{n}') f(\vec{n}', F) \tag{18}$$

where $p(\vec{n})$ is the probability that the users buffer depletes to 0 before transition out of state $\vec{n}$. The quantity $p(\vec{n})$ accounts for the impact of pre-fetching. The term $\hat{P}(\vec{n}, \vec{n}')$ is the probability that the users video is starved after it transits from state $\vec{n}$ to $\vec{n}'$. If $\vec{n} \in \mathcal{X} \setminus \mathcal{B}$, $p(\vec{n}) = 0$, if $\vec{n}' \in \mathcal{X} \setminus \mathcal{B}$, $\hat{P}(\vec{n}, \vec{n}') = P(\vec{n}')$. If $\vec{n}, \vec{n}' \in \mathcal{B}$, $\hat{P}(\vec{n}, \vec{n}') = 1$. If $\vec{n} \in \mathcal{X} \setminus \mathcal{B}$ and $\vec{n}' \in \mathcal{B}$, we have

$$\hat{P}(\vec{n}, \vec{n}') = p(\vec{n}, \vec{n}') f(\vec{n}', F) \\ + (1 - p(\vec{n}, \vec{n}')) \sum_{\vec{m} \in \mathcal{X}} M(\vec{n}', \vec{m}) \hat{P}(\vec{n}', \vec{m}) f(\vec{m}, F) \tag{19}$$

where $p(\vec{n}, \vec{n}')$ is the probability that the user's buffer depletes to 0 before transition out of state $\vec{n}'$ given that just prior to $\vec{n}'$ the user visited state $\vec{n}$. The term $p(\vec{n}, \vec{n}')$ takes into account the possibility of avoiding starvation by user transiting out of $\vec{n}' \in \mathcal{B}$ before its buffer depletes to 0. It is a function of $\vec{n}$ and $\vec{n}'$ since the initial buffer value just prior to entering $\vec{n}'$ is a function of $\vec{n}$ and the rate of buffer depletion then on is a function of $\vec{n}'$.

Computation of $p(\vec{n}, \vec{n}')$ and $p(\vec{n})$: For computing $p(\vec{n})$ for $\vec{n} \in \hat{\mathcal{B}}$, we need to find the time for the buffer to deplete to 0 for a user entering system in state $\vec{n}$. Let us denote this time by $T(\vec{n})$. The term $T(\vec{n})$ includes (a) the time to prefetch q_a seconds of video and (b) the time to deplete the prefetched video. Assuming the prefetching video bitrate is ℓ_1 and using Equation (9), we get

$$T(\vec{n}) = q_a/(r(\vec{n} + e_1)/\ell_1) + q_a/(1 - (r(\vec{n} + e_1)/\ell_1) \tag{20}$$

The sojourn time in state $\vec{n}$ is exponentially distributed with parameter $|\mathbf{Q}(\vec{n}, \vec{n})|$. Hence the quantity $p(\vec{n})$ is given by $\exp(-|\mathbf{Q}(\vec{n}, \vec{n})| T(\vec{n}))$.

We now compute $p(\vec{n}, \vec{n}')$ with $\vec{n} \in \mathcal{X} \setminus \mathcal{B}$ and $\vec{n}' \in \mathcal{B}$. Let $T(\vec{n}, \vec{n}')$ be the time required for the buffer to deplete to 0 when the Markov chain transitions from $\vec{n}$ to $\vec{n}'$. Let us assume that just before transition, the buffer size is $b(\vec{n})$ sec. Then, as $\ell(\vec{n}' + e_1) = \ell_{min}$, using Equation (9), we get

$$T(\vec{n}, \vec{n}') = b(\vec{n})/(1 - r(\vec{n}' + e_1)/\ell_1) \tag{21}$$

and $p(\vec{n}, \vec{n}')$ is then given by $\exp(-|Q(\vec{n}', \vec{n}')| T(\vec{n}, \vec{n}'))$.

Note that the startup delay and the starvation probabilities can be used to compute the expected buffering time. In fact, the expected buffering time is equal to the product of the start-up delay in each re-buffering and the mean number of starvation events (including the initial pre-fetching).

C. The average video bitrate

For computing the average video bitrate of a tagged class-1 user, we need to find the proportion of time that the corresponding Markov chain spends in different states. Let us denote by $N(\vec{n}, \vec{n}')$ the average number of times that a class-1 user entering the system in state $\vec{n}$ spends in state $\vec{n}'$ given that the user finishes service. It is given by

$$N(\vec{n}, \vec{n}') = \frac{\sum_{t \in \mathbb{N}} M^t(\vec{n}, \vec{n}') f(\vec{n}', F)}{f(\vec{n}, F)} \tag{22}$$

The proportion of time, $\tau(\vec{n}, \vec{n}')$ that this tagged user spends in $\vec{n}'$ can be approximated as

$$\tau(\vec{n}, \vec{n}') = \frac{N(\vec{n}, \vec{n}') \frac{1}{|\mathbf{Q}(\vec{n}', \vec{n}')|}}{\sum_{\vec{k} \in \mathcal{X}} N(\vec{n}, \vec{k}) \frac{1}{|\mathbf{Q}(\vec{k}, \vec{k})|}} \tag{23}$$

Thus, the average video bitrate $V(\vec{n})$ for a class-1 user entering the system in state $\vec{n}$ given that the user finishes service is given by

$$V(\vec{n}) = \sum_{\vec{k} \in \mathcal{X}} \tau(\vec{n}, \vec{k}) \ell(\vec{k}) \tag{24}$$

and the unconditional average video bitrate V is given by

$$V = \frac{\sum_{\vec{n} \in \mathcal{X}} \pi(\vec{n}) f(\vec{n}, F) V(\vec{n})}{\sum_{\vec{n} \in \mathcal{X}} \pi(\vec{n}) f(\vec{n}, F)} \tag{25}$$

D. The average rate of bitrate switches

The DASH client can experience video bitrate switches, during a state transition, due to a change in the average channel rate available to the client. For buffer based DASH clients, video bitrate switches can happen even between state transitions, if the average channel rate in a state does not equal any of the available video bitrates. Let $s(\vec{n})$ denote the number of switches that a class-1 user experiences when it visits state $\vec{n}$, during a single visit. Let $z(\vec{n},\vec{n}')$ denote the number of switches that a class-1 user experiences when it transitions from state $\vec{n}$ to $\vec{n}'$, during a single $\vec{n}$ to $\vec{n}'$ transition. Let $N(\vec{n},\vec{k},\vec{k}')$ denote the average number of $\vec{k}$ to $\vec{k}'$ transitions seen by a class-1 user that enters the system in state $\vec{n}$ and finishes its service. This is given by

$$N(\vec{n},\vec{k},\vec{k}') = \frac{\sum_{t\in\mathbb{N}} M^t(\vec{n},\vec{k})M(\vec{k},\vec{k}')f(\vec{k}',F)}{f(\vec{n},F)} \tag{26}$$

Let $S(\vec{n})$ denote the average number of bitrate switches that a class-1 user experiences during its sojourn given that it enters the system in state $\vec{n}$ and that it finishes service. This is given by

$$S(\vec{n}) = N(\vec{n},\vec{k})s(\vec{k}) + N(\vec{n},\vec{k},\vec{k}')z(\vec{k},\vec{k}') \tag{27}$$

and the unconditional average number of video bitrate switches S is given by

$$S = \frac{\sum_{\vec{n}\in\mathcal{X}} \pi(\vec{n})f(\vec{n},F)S(\vec{n})}{\sum_{\vec{n}\in\mathcal{X}} \pi(\vec{n})f(\vec{n},F)} \tag{28}$$

We note that the average rate of video bitrate switching for a video of average duration $\frac{1}{\theta_1}$ is given by S/θ_1.

VI. Simulation and validation

In this section, we compare the numerical results with the simulation framework developed using MATLAB. Extensive simulations have been conducted to validate our analysis using Proportional Fairness (PF) as well as D-VIEWS schedulers. The basic scenario is a LTE downlink with a single eNB and multiple users, including DASH flows and non-adaptive flows. We implemented DASH video streaming on top of the HTTP protocol as well as the adaptation algorithm in the clients. Our model exhibits excellent accuracy.

We simulate a total of 10^5 users entering the system, whose arrival into the network is a *Poisson process* with rate λ_j. Each arriving user requests a video whose duration is exponential distribution, and leaves the system once the file is downloaded. The set of available DASH bitrates $\mathcal{L}$ was chosen as $\{0.50, 1.14, 1.78, 3.07, 5.00\}$ $Mbps$ The duration of the flows was assumed to be exponentially distributed with mean 200 seconds. Furthermore, DASH users adapt their video bitrate according to a buffer-based strategy while non-adaptive users download a fixed video bitrate.

We first assume the presence of a bandwidth slicer that distributes the cell RBs among different traffics and a set of RBs that are dedicated for adaptive DASH flows. For our analysis, the channel rates for a single RB are $\{2,3,5\}Mbps$ with probability distribution over these states as $\{0.2, 0.5, 0.3\}$. We investigate the impact of the traffic load on the QoE metrics. Here we consider three values of the load : 0.4, 0.6 and 0.8, which represent low, medium and high load, respectively. In Figs. 1, 2, 3, 4 and 6, we compare the simulation results $(white)$ and analytic results $(green)$ for QoE metrics. The simulation results have a variance about the mean which is represented by an error bar.

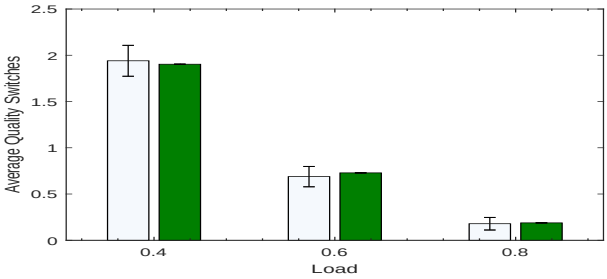

Fig. 1. **Average rate of switching**: comparison of simulation results $(white)$ and analytic results $(green)$ for different traffic loads.

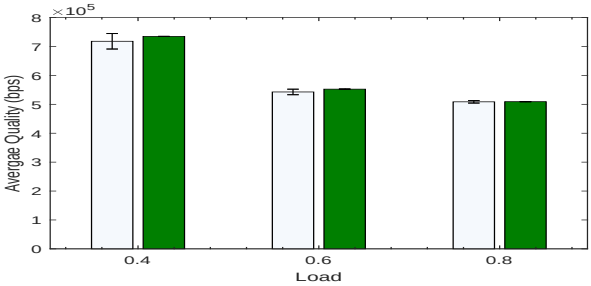

Fig. 2. **Average video bitrate**: comparison of simulation results $(white)$ and analytic results $(green)$ for different traffic loads.

Note: The scheduling policy used in this section is a modification of the D-VIEWS scheduler proposed in [15]. The modification version uses the average throughput of the PF scheduler itself as the target when the average throughput is less than ℓ_1 (the lowest video bitrate).

A. Average video bitrate and average quality switches

In Fig. 1, we compare the rate of switching as predicted by the model against simulation results. From Fig. 1, we can see that our mathematical models are in good agreement with the simulation results. We note that a higher load incurs lower switching rate in comparison with lower traffic load. Indeed, as the traffic load increases, the number of active users in the system increases. Hence, the throughput received by each DASH user decreases to a point where the the video quality chosen is always ℓ_1 and does not exhibit much fluctuations.

In Fig.2, we compute the average bitrate of a DASH user using our explicit model in Eq. 25. One can observe that each pair of average quality bars from the model and from the experiments are very close for different values of load traffic. We observe also that the average bitrate of DASH users decreases when the traffic load increases.

B. Probability of starvation

Next, we investigate the impact of the traffic load on the probability of starvation. We plot both the upper bound (see Fig. 3), which corresponds to the probability to visit a bad state, and exact model of the probability of starvation (see Fig. 4) as function of the traffic load. The upper bound considers only visiting the taboo states whereas the exact solution accounts for transitioning out of taboo states before the playout buffer reaches zero. When the system has a high traffic load, the probability of a DASH user visiting a bad state is higher. Thus, probability of starvation is high when

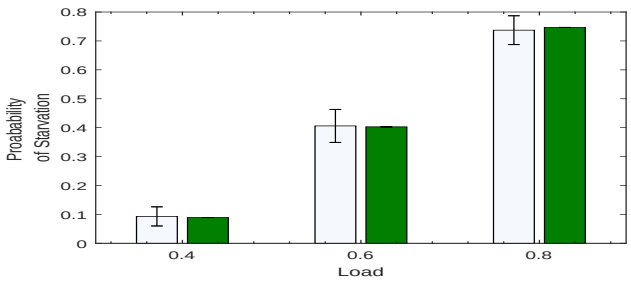

Fig. 3. **Probability to visit a bad state in $\mathcal{B}$**: comparison of simulation results $(white)$ and analytic results $(green)$ for different traffic loads.

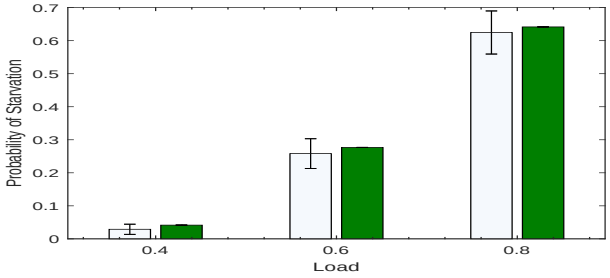

Fig. 4. **Probability of starvation**: : comparison of simulation results $(white)$ and analytic results $(green)$ for different traffic loads.

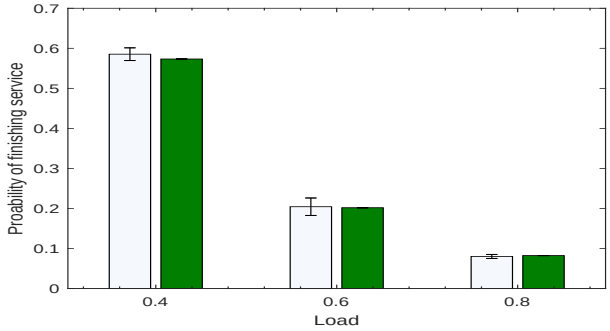

Fig. 5. **Probability of finishing service**: : comparison of simulation results $(white)$ and analytic results $(green)$ for different traffic loads.

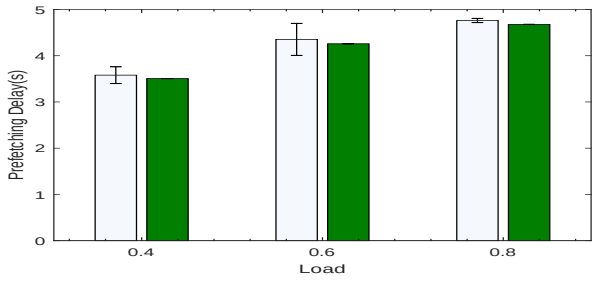

Fig. 6. **Pre-fetching delay:** : comparison of simulation results $(white)$ and analytic results $(green)$ for different traffic loads.

the traffic load is higher. From Figs. 3 and 4, we can see that our analytical approach predicts the starvation probabilities accurately. We observe that the upper bound and the exact solution are very close at low traffic load, and the difference between them increases with traffic load. Fig. 5 shows that the analytical model predict with accuracy the probability of finishing service. We observe that the probability of finishing is decreasing with traffic load since at high load, the probability

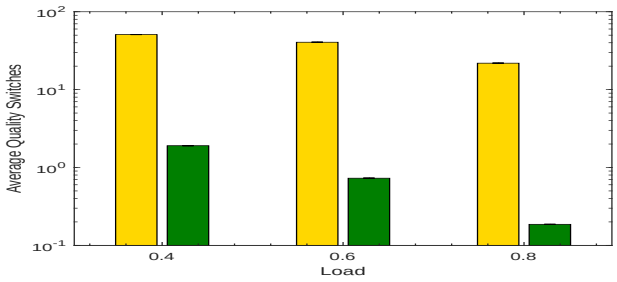

Fig. 7. **Average quality switches**: under PF (green) and D-VIEWS (yellow) schedulers for different traffic loads.

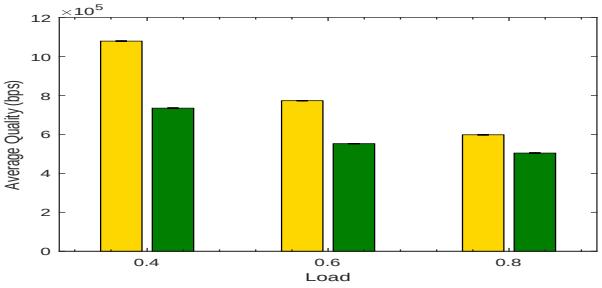

Fig. 8. **Average video quality**: under PF (green) and D-VIEWS (yellow) schedulers for different traffic loads.

of a DASH user visiting a bad state is higher. This may result in DASH users leaving the system on account of poor QoE.

C. Pre-fetching delay

In Fig. 6, we compare the analytically computed pre-fetching delay against simulation results. The pre-fetching threshold is 2 video segments which corresponds to 4 seconds of actual video duration. The video bitrate chosen during pre-fetching is ℓ_1. Our analytical model predict very well the prefetching delay for different traffic load. We also observe that the prefetching delay is slightly impacted by the traffic load.

D. Proportional fairness vs D-VIEWS

In this section, we compare the performance of D-VIEWS with Proportional Fairness scheduler (PF) [14] in terms of QoE metrics using our mathematical framework. We use the same parameters used for validation in the previous section. We compare D-VIEWS with PF for three traffic load : 0.4, 0.6 and 0.8.

In Fig. 7, we plot the average number of switches as function of load traffic. As seen in Fig. 7, D-VIEWS $(yellow)$ ensures a much lower switching rate among the competing DASH users than PF $(green)$ scheduler. With D-VIEWS, we reduce the switching rate by $50-89\%$ while the average bitrate only decreases by $5-18\%$(see Fig. 8). Indeed, D-VIEWS allocates radio resources to each DASH user on its current channel conditions and the set of available peak supported video bitrates. It ensures that the average throughput of DASH users takes values only in the set of DASH available bitrates. The average video quality drop is expected. However, this is not a disadvantage since the resources that are not being used can be allocated to non-DASH users.

In Fig. 9, we plot the probability of starvation for different values of traffic loads under D-VIEWS and PF schedulers. We

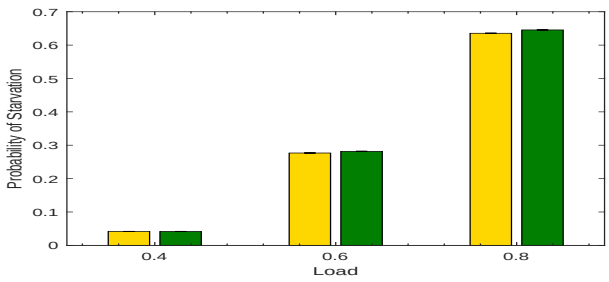

Fig. 9. **Probability of starvation**: under PF (green) and D-VIEWS (yellow) schedulers for different traffic loads.

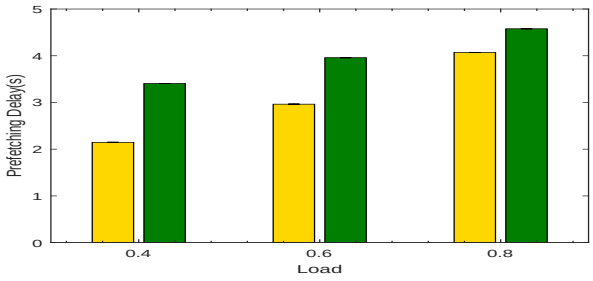

Fig. 10. **Pre-fetching delay**: under PF (green) and D-VIEWS (yellow) schedulers for different traffic loads.

observe that the probability of starvation results change by less than 1%. This is a further validation for the operation of the D-VIEWS scheduler which is designed to reduce switching without negatively affecting the probability of starvation.

Finally, we compare D-VIEWS and PF in terms of pre-fetching delay. As seen in Fig. 10, D-VIEWS achieves a higher prefetching delay as compared to PF since DASH users are allocated lesser resources under D-VIEWS as compared to PF. A straightforward modification (not reported in the original algorithm) is to use the average throughput of PF scheduler during start-up/prefetching without generating further switching.

VII. Conclusion

In this paper, we developed an analytical framework to compute the QoE metrics of adaptive video streaming in wireless data networks. Our frameworks takes into account the dynamics of the system, DASH protocol and the possibility of users leaving the system on account of poor QoE. Specifically, we derived approximations for probability of starvation, average startup delay, average video quality and switching frequency for buffer-based DASH clients. We showed that the proposed models can accurately predict the QoE metrics under different values of traffic load. Our analysis also helped us evaluate the performance of D-VIEWS – our earlier DASH-aware scheduling mechanism İn particular, we evaluated the performance of D-VIEWS as compared with PF in terms of QoE metrics. Our theoretical study reveals that D-VIEWS scheduler is able to achieve a significant reduction of bi-trate video switching while efficiently utilizing the network resources under dynamic flows.

References

[1] Cisco Systems, "Global mobile data traffic forecast update, 2016-2021," *White Paper*, 2017.

[2] C. Ge, N. Wang, G. Foster, and M. Wilson, "Toward QoE-assured 4K video-on-demand delivery through mobile edge virtualization with adaptive prefetching," *IEEE Trans. Multimedia*, vol. 19, pp. 2222–2237, Oct 2017.

[3] K. T. Bagci, K. E. Sahin, and A. M. Tekalp, "Compete or collaborate: Architectures for collaborative DASH video over future networks," *IEEE Trans. Multimedia*, vol. 19, pp. 2152–2165, Oct 2017.

[4] J. Samain, G. Carofiglio, L. Muscariello, M. Papalini, M. Sardara, M. Tortelli, and D. Rossi, "Dynamic adaptive video streaming: Towards a systematic comparison of ICN and TCP/IP," *IEEE Trans. Multimedia*, vol. 19, pp. 2166–2181, Oct 2017.

[5] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 362–373, August 2011.

[6] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," in *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 339–350, ACM, 2013.

[7] M. Z. Shafiq, J. Erman, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Understanding the impact of network dynamics on mobile video user engagement," in *The 2014 ACM international conference on Measurement and modeling of computer systems*, ACM, 2014.

[8] Y. Xu, E. Altman, R. El-Azouzi, S. E. Elayoubi, and M. Haddad, "QoE analysis of media streaming in wireless data networks," in *NETWORKING 2012*, pp. 343–354, Springer, 2012.

[9] A. Goldsmith, M. Effros, R. Koetter, M. Medard, A. Ozdaglar, and L. Zheng, "Beyond Shannon: the quest for fundamental performance limits of wireless ad hoc networks," *Communications Magazine, IEEE*, vol. 49, no. 5, pp. 195–205, 2011.

[10] Y. Xu, Y. Zhou, and D.-M. Chiu, "Analytical QoE models for bit-rate switching in dynamic adaptive streaming systems," *Mobile Computing, IEEE Transactions on*, vol. 13, no. 12, pp. 2734–2748, 2016.

[11] V. Joseph and G. de Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks," in *INFOCOM, Proceedings IEEE*, April 2014.

[12] S. E. E. Thomas Bonald and Y.-T. Lin, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Globcom*, 2015.

[13] S. Poojary, R. El-Azouzi, E. Altman, A. Sunny, I. Triki, M. Haddad, S. Valentin, and D. Tsilimantos, "Analysis of QoE for adaptive video streaming over wireless networks," in *Wiopt, Shanghai, China*, 2018.

[14] T. Lan, D. Kao, M. Chiang, and A. Sabharwal, "An axiomatic theory of fairness in network resource allocation," in *Proc. IEEE INFOCOM*, pp. 1–9, March 2010.

[15] A. Sunny, R. El-Azouzi, E. Altman, S. Valentin, and D. Tsilimantos, "D-VIEWS: Enforcing bitrate-stability for adaptive streaming traffic in cellular networks," in *Technical report, University of Avignon*.

[16] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw*, vol. 8, pp. 556–567, Oct 2000.

[17] J. W. Cohen, "The multiple phase service network with generalized processor sharing," *Acta Informatica*, vol. 12, pp. 245–284, Oct 1979.