Scheduling Algorithms for 360º Video Streaming Over 5G Networks

Pedro Martin
Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal
pedrogcmartin@tecnico.ulisboa.pt

***Abstract*—With the evolution of wireless networks over the last years, the consumption of multimedia content has constantly been increasing, leading to the development of new applications. In particular, the arrival of a new mobile communications standard – the 5th Generation (5G) – is expected to provide the required network capacity for the enhancement of 360º video streaming. Nevertheless, the efficient management of the available network resources will continue to be one of the main challenges that mobile operators will have to face, to support many users and provide them a high Quality of Experience (QoE). In this sense, this work presents a study of scheduling algorithms for 360º video streaming over 5G networks. The performance of conventional schedulers – namely the Round-Robin (RR), the Blind Equal Throughput (BET), the Maximum Throughput (MT), and the Proportional Fair (PF) – are evaluated from a users' QoE perspective. Two buffer-level-based scheduling algorithms – the Buffer Equalization Approach (BEA) and the Proportional Buffer Filling (PBF) – are also suggested and compared with the conventional schedulers. Based on the obtained results, a QoE-driven Joint Admission Control and Scheduling (QJACS) is proposed, to further enhance the 360º video streaming experience. A simplified version of QJACS when the BET scheduler is adopted, is also proposed.**

***Keywords*—360º Video Streaming, 5th Generation (5G), Scheduling Algorithms, Quality of Experience (QoE)**

I. INTRODUCTION

In the coming years, a revolution in wireless networks is expected to take place. Despite the impact of COVID-19 in all parts of society, the year-on-year increment in the number of mobile services subscriptions did not change; in 2020, 7.9 billion subscriptions worldwide, an all-time record, were reached [1]. The grew in the number of network-connected devices, mobile data traffic, number of new network-based applications and services, and expected high Quality of Experience (QoE) has led to the need for a complete wireless network's renewal. Thus, the next generation of wireless systems, also known as the 5th Generation (5G), aims to provide significant performance enhancements regarding the current generation, referred to as Long Term Evolution (LTE).

Additionally, the primary cause of mobile traffic growth has been the increasing consumption of video content in recent years. Nowadays, 66% of mobile data traffic accounts for video traffic, and in 2026 it is forecast an increase to 77% [1]. The mobile service client seeks more online video streaming, video conference, and video call platforms. 360º video streaming is one of the video services which, due to the 5G advent, is expected to achieve more popularity among mobile businesses and clients. This type of visual content allows the user to navigate spatially through the video by changing its viewing direction. To provide a more immersive experience, 360º video can be displayed using Head Mounted Display (HMD), allowing the control of the viewing direction with head movements. The availability of HMDs and 360º cameras, the creation of video-sharing websites, and of social media platforms enabling users to publish and view 360º video content are the three main factors for 360º video increasing popularity [2].

A good economy of the available network resources will play a significant role in future applications of 360º video streaming in 5G wireless networks. Scheduling algorithms apply different strategies, seeking better management of the available network resources. In particular, 360º video streaming may have a set of issues where users with a bad connection may experience not only low video quality but also too long initial video display waiting time, video stalls, frequent changes of video quality, audio/video desynchronization, and viewports with no video content, among others. Thus, it is necessary to have an intelligent resource allocation strategy to ensure a high degree of user satisfaction while efficiently using network resources.

Since there is a lack on scheduling algorithms for 360º video streaming over wireless networks, the main objective of this paper is to design, implement and evaluate the performances of scheduling algorithms solutions for this type of video transmission over 5G networks. For that purpose, both conventional and new scheduling solutions are evaluated and compared. The performance evaluation of each scheduler follows a QoE-centered methodology, as in a real-world scenario, where the number of users having a good service experience is a key information for a mobile operator.

The paper is organized as follows: Section II reviews the related work; Section III presents the conventional and suggested schedulers: Buffer Equalization Approach (BEA) and Proportional Buffer Filling (PBF); Section IV describes the development of the simulator used for the algorithms' performance evaluation; Section V presents and analyzes the simulation results and also details the proposed QoE-driven Joint Admission Control and Scheduling (QJACS); and Section VI concludes the paper.

II. RELATED WORK

This section addresses the related work by providing an overview of the (i) 5G networks; (ii) 360º video streaming; and (iii) scheduling algorithms.

A. 5G Networks

Roughly, every ten years, there is a revolution in the mobile radio networks field. Each new technology generation

provides an improvement in the performance of these systems in order to satisfy user demands. Even though in the last ten years the massive data content growth was caused chiefly by video streaming applications, 5G will respond to many use cases. The three main use cases established by ITU-R in IMT-2020 [3] are the following:

(i) Enhanced Mobile Broadband (eMBB): The focus of this use case is to improve the communication performance between Base Stations (BSs) and Mobile Terminals (MTs) by increasing the data rate of the data services. Using higher frequencies – mmWaves – and wider bandwidths, the Uplink (UL) and Downlink (DL) throughput reach up to 10 Gbps and 20 Gbps, respectively. However, when the working frequency is very high – in 5G reaches up to 52.6 GHz – the attenuation due to transmission loss increases compared to lower frequencies. Therefore, network densification and mMIMO technology are proposed, which means using smaller cells in higher quantities – especially at urban locations – and using antenna diversity techniques to lower the transmission losses throughout the radio channel and increase the received signal power. However, the sole implementation of these mechanisms is not sufficient to ensure a good QoE. Cellular devices use omnidirectional antennas, which, in a BS densification scenario, can generate severe interference. In order to reduce the interference levels, complex beamforming algorithms are implemented at the BS, allowing the transmission of a specific data stream to a specific MT and allowing the support of simultaneous data streams by the BS;

(ii) Ultra-Reliable Low Latency Communication (URLLC): Some applications like autonomous driving, remote surgery, and online games require stringent reliability, availability, and latency. In particular, this use case focus on latency-sensitive and ultra-reliable technologies. For example, in a remote surgery application, the network must fill both these requirements to avoid drastic scenarios. Accordingly, it is established that the 5G systems should provide an end-to-end latency with duration no longer than 1 ms and reliability of 99.9999%. Features like integrated frame structure, grant-free based uplink transmission, and advanced channel coding schemes are essential in implementing this use case;

(iii) Massive Machine-Type Communication (mMTC): Several new applications such as smart cities, smart agriculture, and wireless control of industrial manufacturing will arise in the near future. These applications belong to a new technological field generically mentioned as the Internet of Things, and its implementation will be possible due to mMTC. This use case enables a massive application of Machine-to-Machine systems which interconnects different devices, exchanging a considerable volume of data within the network. Tens of billions of Internet Protocol (IP) devices will transmit and receive information to orient every device into a specific task. This type of application with high connection density requires a high Energy Efficiency (EE), which means that devices must have low power circuits and very long battery life. The data rate will be in the order of kbps.

Each one of these use cases establishes extremely demanding requirements regarding current mobile radio networks configurations. The 5G key technologies that will enable the network operation enhancement are the following:

(i) Increased bandwidth: The 5G deployment takes advantage of the free spectrum frequency region located above the conventional ones (3 GHz). Accordingly, the centimeter and millimeter wave bands (28 – 300 GHz) are regions with a plenitude of the available frequency spectrum that enables wide carrier bandwidths – in the order of 1 GHz. Two regions were then defined to use 5G technologies, which are mentioned as Frequency Region 1 (FR1) and Frequency Region 2 (FR2). FR1 corresponds to the frequency range between 410 MHz and 7.125 GHz, and FR2 corresponds to the frequency range between 24.25 GHz and 52.6 GHz.

(ii) mMIMO and beamforming: One of the positive consequences of using high frequencies for mobile communications is that it makes it possible to deploy smaller antennas and, therefore, large-scale antenna arrays at the BSs. This has the benefit of decreasing the transmission loss and increasing the communication reliability by introducing an array gain and providing spatial multiplexing gain. Each antenna array may be composed of up to 256 to 1024 (for the mm-wave bands) cross-polarized elements arranged in a 2D array, where each element consists of a dipole or patch antenna. Due to the large number of antenna elements, the Massive Multiple-input Multiple-output (mMIMO) system enables beamforming techniques and the operation of sub-arrays of the main antenna array depending on the network conditions and user demands. Hence, it provides greater adaptability and steerability caused by the high number of degrees of freedom on the antenna system. Beamforming techniques allow the adaptation of the antenna radiation patterns both in azimuth and elevation angles, which enables a better adaptation to the rapidly time-varying traffic and multipath radio propagation conditions.

(iii) Network densification: One of the main contributions to reach a high spectral efficiency is using small cells for network densification. It has been established that besides macrocells (with a cell radius between 8 and 30 km), the following 5G cell categories will be used: microcells (0.2 – 2 km), picocells (0.1 – 0.2 km), and femtocells (0.01 – 0.1 km). Small cells are specifically oriented to cover indoor hotspots, urban areas, and dense urban areas. The deployment of small cells does not exclude the need to use macrocells to carry control plane traffic.

(iv) New waveforms: It was set by 3GPP that the signal used by 5G would be Cyclic-Prefix based Orthogonal Frequency Division Multiple Access (CP-OFDMA) for both DL and UL. Basically, a given amount of data is transmitted throughout the radio channel simultaneously inside different data blocks, using orthogonal subcarrier frequencies for each block. Each data block carries a set of symbols and, as the first part of each symbol is cut and appended either at the end or beginning of the corresponding symbol, the waveform is denominated as CP-OFDMA. The cyclic prefix or postfix technique allows the reduction of intra-symbol interference since, given the length of the chosen prefix or postfix, it can avoid selective fading negative impacts on the correct transmission of each symbol. Additionally, flexible numerology regarding the radioframe structure has been defined to cover service for the entire set of applications described before. This will allow eMBB and mMTC proper

operation and play an essential role in LTE and 5G coexistence and beamforming.

B. 360º Video Streaming

As the user connection is not stable in wireless communications, Adaptive Bitrate Streaming (ABS) techniques are often used to allow the adaptation of the video quality to the network conditions (*e.g.*, end-user device buffer occupancy level, radio channel conditions, available bandwidth). In particular, the MPEG has defined the DASH standard for that purpose. Figure 1 illustrates a DASH operation example.

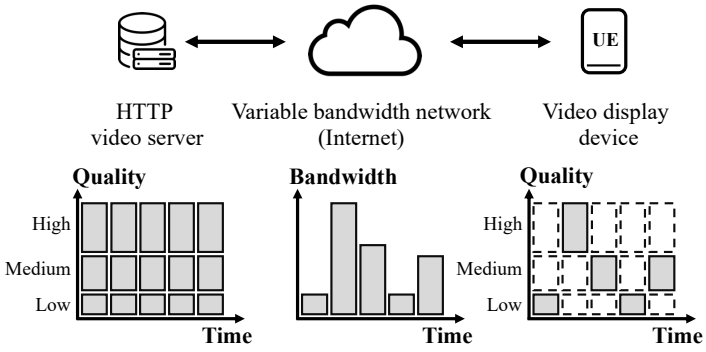

FIGURE 1: Workflow example for DASH streaming.

In practice, the implementation of DASH requires a content creator which encodes and segments the respective media content and creates a Media Presentation Description (MPD) file containing information about program timing, media-content availability, media file types, resolutions, available bitrates, media-component location, among others. The resulting files are then stored in a regular Hypertext Transfer Protocol (HTTP) web server. The closer it is to the client location, the lower the bandwidth requirements and server load and response times. Concisely, the media content available at the server for user consumption consists of segments containing the actual multimedia bitstreams and a MPD document that describes the temporal and structural relationships between segments.

Moreover, the SRD feature extends the MPD of DASH by describing spatial relationships between related parts of the video content. Accordingly, with SRD is possible to offer bitrate adaptation for not only each video segment but also different spatial parts. In particular, SRD plays a major role in 360º video streaming. As the user can only select for visualization a fraction from the 360º sphere (viewport), some streaming techniques exploit the bitrate adaptation of different spatial parts (tiles) of the same segment. Figure 2 illustrates a Viewport Adaptive Streaming (VAS) example using SRD.

As shown in Figure 2, with VAS the server can adapt the resolution of each tile based on parameters such as the user's viewport direction, connection quality, and buffer length. According to SRD, the MPD must have for each tile information associated with its position on the overall video frame. A 2D Cartesian coordinate system is used, having its origin on the top-left corner of the video. Each tile is attributed with an ID – where the order of the IDs attributed describes the sequential representation of the video frame. Each ID is used as a reference to a pre-established spatial position. Furthermore, the width and height of each component and the total width and height of the frame are also defined in coordinate system units. This information is relevant when overlapping tiles exist in complex representations.

In order to present high-quality omnidirectional video content, the acquisition, encoding, and transmission of the full omnidirectional frame are required, leading to spatial resolutions up to 12K (11520x6480 pixels). A temporal resolution of up to 90 frames/s should also be delivered to mitigate motion sickness and offer the best user experience, implying that bitrates around 1 Gbps are required. Accordingly, if many users are located in the same geographic region and desire to watch the same omnidirectional video with a satisfying QoE, the most likely scenario is that current LTE networks will not support it. As previously explained, given that during the video playback, only the viewport is displayed at each instant, to reduce the amount of required bandwidth, the streaming of the video content according to the viewport location is considered in VAS. Thus, the server adapts the resolution of each tile based on parameters such as the user's viewing direction, connection quality, and buffer length. This sets some requirements for the video streaming: (i) there must be a viewport prediction technique to allow the quality enhancement of some tiles over others according to the viewport visualization probability; (ii) as long viewport predictions have low accuracies, the buffer size must be reduced concerning the traditional 2D video streaming; and (iii) the network latency must be lower than 20 ms, as the user's head movement can change the viewport position as quick as that time [4].

Moreover, many factors can affect the user experience of omnidirectional video streaming. Depending, for example, on the number of video stalls, long re-buffering periods, frequent changes in video quality, low frame rates, audio/video desynchronization, presence of artifacts (noticeable distortion of media content), and long initial buffering times, the user might decide not to pay for an omnidirectional video service. In particular, drastic changes in the omnidirectional video streaming quality can be introduced by erroneous mapping projection, lossy coding, unpredicted or quick head movements. In particular, it is important for mobile operators to have some objective quality assessment metrics, which can be used to estimate the user QoE. Based on the estimated user QoE, operators can schedule resources more efficiently. The MPEG-DASH contemplates the reporting of some of these metrics by the user's device (*e.g.*, buffer level, average throughput, and initial playout delay).

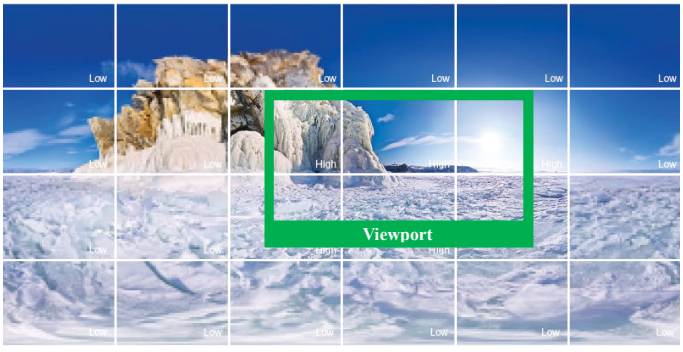

FIGURE 2: SRD use case, [19].

C. Scheduling Algorithms

The deployment of wireless technology as the 5G, even with every specification presented and discussed in Section II.A, can be not enough to upgrade the wireless network services if the management of the available resources is poorly. Thereby, the distribution of the available network resources among each user has always been one of the essential tasks for a mobile operator, namely to enhance the delivered Quality of Service (QoS) and QoE. Accordingly, and to efficiently allocate the network resources, a scheduler – which is typically located at each BS – decides how users share the wireless channel. More specifically, the scheduler allocates network resources to users by aiming at the maximization of an utility function.

The resource allocation decision is usually based on a comparison between transmission priorities of specific a Resource Block (RB) for each user. If the metric $m_{j,k}$ is the highest metric among which the k^{th} available RB can be assigned, then the j^{th} user will be assigned with it. In other words, the allocation of the k^{th} RB to the j^{th} user, among every possible i^{th} user, is defined as:

$$m_{j,k} = max_i\{m_{i,k}\}, \quad (1)$$

where the $m_{j,k}$ metric can be dependent on the radio channel condition, buffer state, resource allocation history, among others. For example, the higher the radio channel quality for a given user, the higher could be the metric, giving preference to this user over the users with lower radio channel quality.

The scheduling algorithms considered in this paper can be described as follows:

(i) Round-Robin (RR): Being considered a fair algorithm, RR provides a cyclic allocation of the available RBs to the user's requests. Thus, RR allocates some number of RBs to every user over the streaming session. The RR metric of the i^{th} user for the k^{th} RB depends on the current time instant, t, and the instant when the user was last served, T_i:

$$m_{i,k}^{RR} = t - T_i. \quad (2)$$

(ii) Blind Equal Throughput (BET): Being the fairest scheduler, BET provides the same average throughput to every user, independently of their channel conditions. Hence, on average, users with lower reported CQIs are allocated with more RBs than users with higher reported CQIs. Denominating as $\overline{R_i(t)}$ the average throughput of the i^{th} user at the current instant, t, the BET metric is described as:

$$m_{i,k}^{BET} = \frac{1}{\overline{R_i(t)}}. \quad (3)$$

(iii) Maximum Throughput (MT): In order to exploit the expected achievable throughput information immediately, MT prioritizes the allocation of resources to the user with the current best channel conditions. Consequently, it maximizes the overall throughput (or spectral efficiency) since, at a given TTI, the maximum possible throughput in the network is achieved, according to the users' available resources and channel conditions. Nevertheless, this is considered an unfair algorithm regarding the users with poor channel conditions, which are condemned to starvation in most scenarios. The MT metric can be expressed as:

$$m_{i,k}^{MT} = d_k^i(t), \quad (4)$$

where the expected achievable throughput, $d_k^i(t)$, for the i^{th} user at instant t with the k^{th} RB, can be given by:

$$d_i(t) = N_{RB_i} \cdot eff(CQI_i) \cdot \frac{12}{T_s^\mu} \cdot G_{MIMO} \cdot (1 - OH^T), \quad (6)$$

where N_{RB_i} is the number of allocated RBs to the i^{th} user, $eff(CQI_i)$ is the CQI efficiency which corresponds to the current CQI index of the i^{th} user, G_{MIMO} is the transmission gain due to the use of the mMIMO antenna configuration, OH^T is the percentage of overhead bits, having in consideration the required bits for tiling, and T_s^μ is the average OFDMA symbol duration in a subframe for numerology μ and it is given by:

$$T_s^\mu = \frac{10^{-3}}{14 \cdot 2^\mu}. \quad (7)$$

III. Proposed Scheduling Algorithms

From the mobile operator viewpoint, the best algorithm is the one that maximizes the number of satisfied users with a QoE above a specific threshold and minimizes the number of unsatisfied users with a QoE below a specific threshold. With the scope of reaching this target, two buffer-level-based algorithms are proposed in this paper: Buffer Equalizing Approach (BEA) and Proportional Buffer Filling (PBF). These algorithms' metrics can be described as follows:

(i) Buffer Equalization Approach (BEA): This algorithm is proposed to evaluate the scheduling decision from a buffer management viewpoint. In 360º video streaming the buffer size has a relevant impact on the service. The BEA's metric gives priority to the user with lowest buffer level, $B_i(t)$:

$$m_{i,k}^{BEA} = \frac{1}{B_i(t)}. \quad (8)$$

(ii) Proportional Buffer Filling (PBF): In order to add a level of opportunism to BEA, the PBF scheduler is also proposed having in consideration the average buffer level of the user, $\overline{B_i(t)}$:

$$m_{i,k}^{PBF} = \frac{\overline{B_i(t)}}{B_i(t)}. \quad (9)$$

The study of the algorithms presented in (8) and (9) is motivated by the fact that the buffer size has a critical impact on video streaming. Since the 360º video streaming has such reduced buffer sizes concerning 2D video, and consequently, is more vulnerability to the staling phenomenon, the idea of controlling its level directly at the scheduling decision gains interest. The first metric, in (8), always chooses to serve the user more vulnerable to staling, *i.e.*, with a currently lower buffer length. The second metric, in (9), adds a layer of opportunism to the previous one by also considering the users who have kept on average a higher buffer length, as users with better channel conditions should have more facility in keeping their buffer levels high regarding the remaining users.

Accordingly, to evaluate the performance of the proposed algorithms, the RR, BET, MT, and PF schedulers are also studied, allowing for a scheduler´s comparison. RR is typically used as a benchmark in literature; BET is the fairest algorithm concerning the QoE of each user; MT maximizes the network spectrum efficiency; and PF takes advantage of the user's channel conditions while allowing for some fairness among different users.

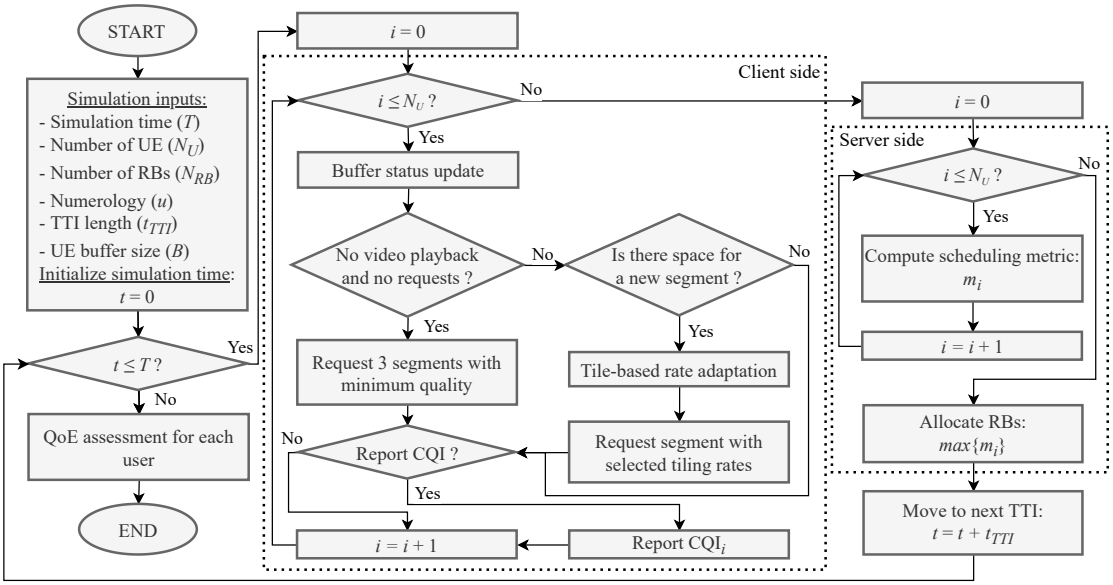

FIGURE 3: Simulator architecture flowchart.

IV. SIMULATION SETUP

Regarding the simulation setup, it has a time duration of 1 minute. The number of connected users is constant over the simulation, and the users have independent CQI reporting and HMD trajectories. The CQIs are reported at every 5 milliseconds. The requested segments have a playback duration of 1 second. There are no network or transmission time delays between the server and the client. There are no bit losses in the transmission. The number of available RBs for allocation at each TTI equals 11 RBs. The VAS algorithm used is the BQA with the parameters $B_{min} = B_{max} = 1.8$ seconds. When there is the video initialization or a stall, the client's device requests for 3 segments with the middle-quality (4 out of 7 levels, and homogeneously regarding the tiling configuration). The tiling configuration is 6 columns and 4 rows.

A. Simulator's Architecture

A simulator was developed from scratch based on previously exposed knowledge to evaluate the scheduling algorithms performances in a 360º video streaming. The simulator architecture is described in Figure 3 and has the following working principle: a client desires to watch a given online 360º video on a device; the video content is located at a server; the user's device is going to access that server in order to receive the video content; as the connection's quality between the user's device and the server is time-variant, in order to adapt the video quality to those conditions, the video exists on the server in a temporal-segmented version and, for each segment, different bitrates are available for streaming. For a 360º video streaming context, the segments can be divided into complementary tiles, allowing for the bitrate adaptation at the video region level. So, different qualities can be observed depending on the user's head behavior for each visualized segment and, indirectly, on the channel conditions. Moreover, in the first requested segments, the user's device does not know the throughput conditions. Thus, the user's device first requests low bitrate segments. The server's decision to answer those requests depends on conditions such as if there are more requests from other clients and any available RBs for allocation. When the server allocates the available RBs to a given user, the client starts receiving bits from the requested segments. The better the channel conditions, the more bits are possible to transmit for a given period, and vice-versa. The user's device stores the received bits at a buffer component. For simplification, this simulator's buffer occupancy level measurement is the stored duration, in milliseconds, of the currently playing sequence, and not the number of bits. This conversion depends on the playback duration of the requested segment, its bitrate, and the number of received bits so far. When the user's buffer is filled with the initial segments, the video playback takes place. Henceforth, the buffer level update also depends on the user's visualization rate - negatively contributing to the buffer occupancy level. After the playback initialization, depending on the buffer's length, *i.e.*, if there is enough space for a new segment, the user's device will request new segments. This time, as the client had already received bits and can estimate its throughput, the adaptation of the following requested bitrates and tiling configurations is made. During the 360º video streaming, the buffer occupancy level will vary according to the number of received bits and the number of visualized milliseconds per TTI. Depending on the number of served clients, network configuration, requested segments, and the implemented scheduler, the buffer level of a given user may become empty, originating a stall event. Whenever there is a stall, the user's device will repeat the initialization procedure of requesting low-quality segments until the buffer is filled and apt to play more video content.

B. VAS Algorithm

In order to perform streaming adaptation to the viewport position, the Buffer-quality-based Algorithm (BQA) [5] is used at the client-side. This algorithm selects the rate of each tile according to the estimated throughput, viewport prediction, and buffer level. BQA defines two buffer thresholds. If the buffer length, $B(t)$, is below B_{min}, the requested bitrate should be lower than the estimated throughput in order to restore a comfortable buffer occupancy level. If the buffer is above B_{max}, the requested bitrate should

be higher than the estimated throughput in order to smooth the video quality. Given the estimated throughput the segment bitrate defined by BQA is defined as:

$$R_{l,request} = \varepsilon \cdot U_{est}, \quad (10)$$

where,

$$\varepsilon = \begin{cases} \frac{B(t)}{B_{min}}, & B(t) < B_{min} \\ 1, & B_{min} \leq B(t) \leq B_{max} \\ \frac{B(t)}{B_{max}}, & B(t) > B_{max} \end{cases}. \quad (11)$$

The estimated throughput averages the perceived throughput for the last three replied segments. Depending on the number of segments considered, *S*, the bitrate of the i^{th} requested segment, r_i, the playback duration of the segment, t_0, and the download time of the i^{th} video segment, the estimated throughput, U_{est}, is the computed as:

$$U_{est} = \frac{1}{S}\sum_{i=i_{last}-S+1}^{i_{last}} \frac{r_i \cdot t_0}{t_{DL,i}}. \quad (12)$$

Regarding the viewport prediction, a persistence technique is implemented given it is typically used as benchmark, and this is not the scope of the study. The persistence technique states that the shorter it is the prediction interval, the more the future viewport position is similar to the current one, making it the predicted viewport.

Based on these considerations, the Algorithm I is implemented, being a joint optimization of a commonly used benchmark in literature [6, 7] and BQA operation. This algorithm can be described as follows: suppose that, at a given time, the client makes a new segment request, and at that time, the estimated throughput is U_{est}. Let l_m denote the selected quality level for the *m* tile. Initially, the algorithm will set the lowest quality for every tile (line 1). Then, the current bandwidth budget is estimated (according to (12)) to know if it is possible to enhance the quality of the tiles (line 2). Depending on the current viewport coordinates, the tiles are sorted from viewport to adjacent to outside zone (line 3). Starting from the second-lowest available tile quality (line 4), the algorithm will try to update each tile quality, iterating each tile by the sorted order (line 5). So, first, it computes the bandwidth cost, Δb, needed to update the tile quality to the current one by calculating the additional bitrate to pass from the previous bitrate, b_{prev}, to the current one, b_{cur}, (line 6). If this cost is within the total budget (line 7), the budget is updated for that cost (line 8), and regarding the current tile, the selected quality level increments one level (line 9).

Algorithm I: Viewport Adaptive Streaming Algorithm

Input:

R (Segment bitrate)
V (Current viewport coordinates)
N_T (Total number of video tiles)
b(-) (Available tile rates vector)

Output:

$\{l_m\}_{1 \leq m \leq N_T}$ (Assigned tiles quality vector)

1 $l_m \leftarrow 0$ for $1 \leq m \leq N_T$
2 $Budget = R - N_T \times b(0)$
3 $sortedTiles \leftarrow sort(V)$
4 **for each** $b \in b(-), \nexists b(0)$ **do**
5 **for each** $m \in sortedTiles$ **do**
6 $\Delta b = b_{cur} - b_{prev}$
7 **if** $Budget > \Delta b$ **then**
8 $Budget \leftarrow Budget - \Delta b$
9 $l_m \leftarrow l_m + 1$
10 **end if**
11 **end for**
12 **end for**

C. QoE Model

In order to estimate the user's level of satisfaction concerning the streaming service a QoE model is implemented [8]. Accordingly, the model inputs are from two categories: quality-oriented (quality average and standard deviation) and stalling-oriented (frequency and average duration of video stalls). The QoE of the i^{th} client is defined within the same range of MOS by the following expression:

$$QoE_i = min\left(max\left(5.67 \times \frac{\bar{q}_i}{q_{max}} - 6.72 \times \frac{\hat{q}_i}{q_{max}} + 0.17 - 4.95 \times F_i, 0\right), 5\right), \quad (13)$$

where $\bar{q}_i$ is the average quality level requested, $\hat{q}_i$, is its standard deviation – both normalized concerning the highest available quality level q_{max} – and F_i, models the influence of video freezes. The min-max guarantees the function limitation to the MOS scale. F_i is computed as follows:

$$F_i = \frac{7}{8} \times max\left(\frac{ln(\phi_i)}{6} + 1, 0\right) + \frac{1}{8} \times \frac{min(\varphi_i, 15)}{15}, \quad (14)$$

where ϕ_i is the stalling frequency and φ_i is the average stall duration. The QoE model implementation neglects the video initialization period, except the initial playout delay. This means that first the rate control gradually increases the video quality given the estimated throughput. When it reaches the first realistic video segment, the QoE model parameters are turned on. The first realistic segment is considered to be the first segment where the requested bitrate is equal to or lower than the previous one. At this point, the rate control is already sensitive to the estimated throughput. As this initialization is transversal to every user, this is a reasonable assumption.

Moreover, unlike the traditional 2D video streaming, where during the visualization of a given segment, the video quality remains the same – depending on the segment bitrate – in 360º video streaming, the viewport quality can vary during a given segment visualization. Therefore, the quality parameter is computed as a weighted average between the viewport tiles' qualities at each TTI, depending on the area of visualization for each tile.

Finally, when using (11), the mapping of the perceived quality is as follows: (i) very bad streaming when QoE is between 0 and 1; (ii) bad when between 1 and 2; (iii) fair when between 2 and 3; (iv) good when between 3 and 4; and very good when between 4 and 5.

D. 360º Video Dataset

A 360º video dataset has been selected to get the simulator's input coordinates that each user looks at each instant. The JVET dataset [9] is used for that purpose. The 360º video sequence visualized is the same for every user (UnderwaterPark), and the tiling bitrates present in the server are: {16.25, 20.83, 27.92, 37.50, 54.17, 74.17, 94.58} kBps (7 quality levels). As the video sequences present in [9] have 20 seconds playback duration, the dataset inputs are cyclic

acquired when the simulation time is longer than 20 seconds to overcome that problem.

V. Simulation Results

Based on the information provided in Section IV, the developed simulator was user in order to assess the performance of the scheduling algorithms described in Section II.C and III from a QoE viewpoint.

A. Algorithm Comparison Methodology

The 360º video streaming is simulated for a given range of connected users, between 1 and 100 users. Depending on the simulated scheduler, at each point, different users may or not have different outcomes, with higher or lower global QoEs, *i.e.*, the final QoE at the end of streaming. This heterogeneity in the users' outcomes can be measured in a way that makes it possible to compare what schedulers provide better service than the others for a given number of connected users. The service requirements' thresholds used in this document are: 95%, 90%, and 80% of users with global QoE ≥ 3; 95%, 90%, and 80% of users with global QoE ≥ 4; and 0%, 5%, and 10% of users with global QoE < 2. This way, according to the following simulation results of each scheduler, it is possible to study what scheduler provides better conditions for a 360º video streaming service where the scope is to maximize the number of connected users that have, at least, a good service, an excellent service, and not a bad service, respectively.

Accordingly, the scheduling comparison problem can be formulated as follow: for each scheduling algorithm what is the number of connected users, N_U, that minimizes the absolute value of the difference between the percentage of users with a global QoE contained within a particular QoE region, QoE_R, and the percentage of users' threshold, P_α:

$$arg \min_{N_U \in [1,100]} abs\left(\frac{|global(QoE_i) \in QoE_R|}{N_U} - P_\alpha\right), i \in Users, \quad (15)$$

Note that in (15), in order to distinguish the absolute value function from the counting of the users satisfying the $global(QoE_i) \in QoE_R$ condition and belonging to the set of users, *Users*, the following notation is used: $|\cdot|$ stands for the counter of elements of a given set, and $abs(\cdot)$ stands for the absolute value function.

Moreover, for each of the following simulation results, it is possible to say that with 95% certain the true prediction is, at least, within 10% of the mean of the sample, *i.e.*, when the QoE outcomes of the n simulated samples for each number of the connected users' point are averaged, the accurate mean of that stochastic process is 10% higher or bellow that value with 95% certainty – being n the number of Monte Carlo simulations.

B. Performance Analysis

In this section, the performance of each scheduling algorithm is evaluated according to the described methodology. In particular, the percentage of users with a global QoE contained in the QoE regions: $QoE_R = \{[3, 5], [4, 5]\}$ (*cf.* Figures 4 and 5), and $QoE_R = [0, 2]$ (*cf.* Figure 6); is plotted for each scheduler varying the total number of connected users from 1 to 100 users.

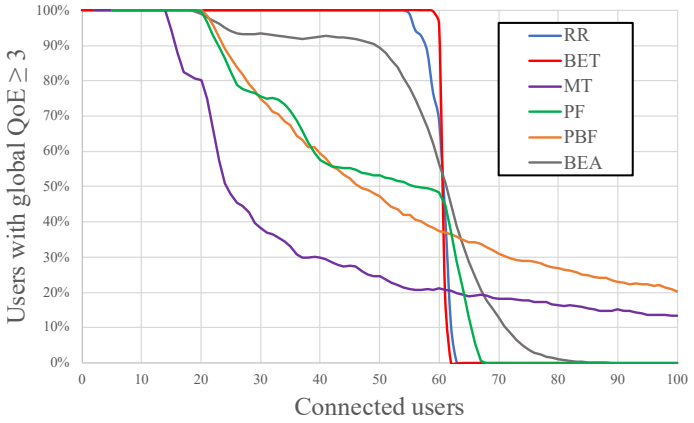

FIGURE 4: Percentage of users with QoE ≥ 3 over the number of total connected users.

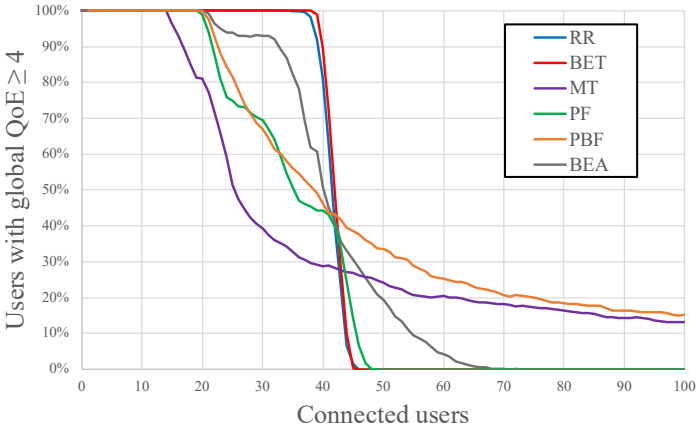

FIGURE 5: Percentage of users with QoE ≥ 4 over the number of total connected users.

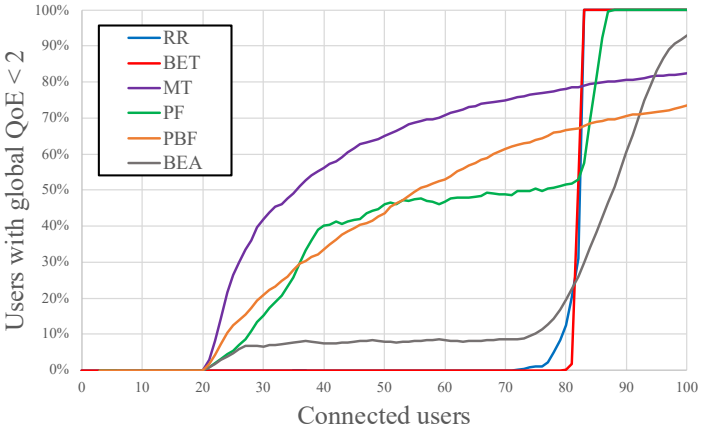

FIGURE 6: Percentage of users with QoE < 2 over the number of total connected users.

Figure 7 presents results in Figure 4 that solve (15) for each scheduler for $\alpha = \{80, 90, 95\}\%$. As it is possible to observe in Figure 7, the scheduler that has better results for every performance evaluation scenario is the BET: with 60.03 connected users for P_{95}, 60.48 for P_{90}, and 60.77 for P_{80}. The RR scheduler also has significantly better results than the remaining schedulers: with 3.66 connected users lesser than BET for P_{95}, 3.34 lesser for P_{90}, and 2.44 lesser for P_{80}. As the RR and BET approach the breaking point at the 60 connected users point, both schedulers decrease their capacity with similar slopes (*cf.* Figure 4). The buffer-level-based algorithms, despite not exceeding BET and RR, present good

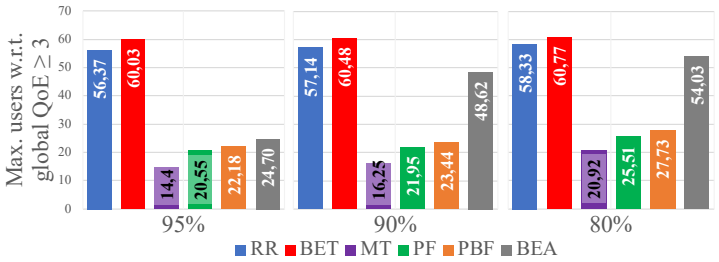

FIGURE 7: Maximum number of users w.r.t. global QoE ≥ 3 for $\alpha = \{80, 90, 95\}\%$.

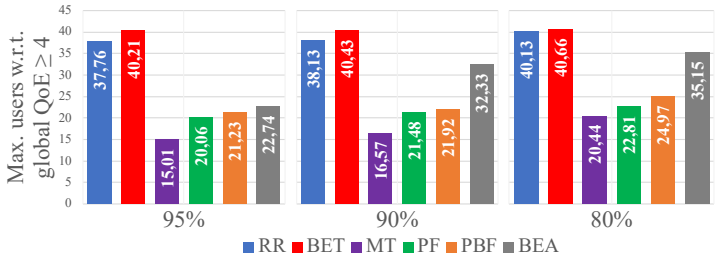

FIGURE 8: Maximum number of users w.r.t. global QoE ≥ 4 for $\alpha = \{80, 90, 95\}\%$.

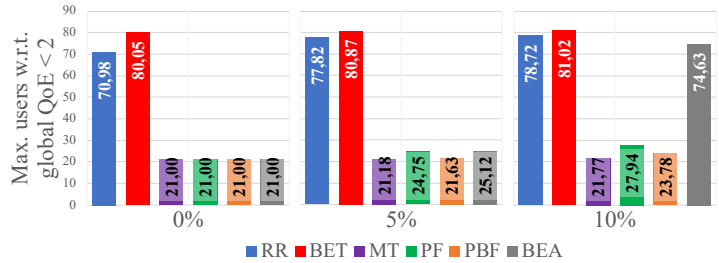

FIGURE 9: Maximum number of users w.r.t. global QoE < 2 for $\alpha = \{0, 5, 10\}\%$.

results compared to MT and PF. In particular, for a P_{90} and P_{80}, the BEA algorithm has a considerably better performance than the MT and PF schedulers, with a deviation between +28.52 (regarding PF for P_{80}) and +32.37 connected users (regarding MT for P_{90}).

Figure 8 presents the results in Figure 5 that solve (15) for each scheduler for $\alpha = \{80, 90, 95\}\%$. Regarding Figure 8, it is interesting to notice that in Figure 7, the order of better to worst performances remains the same for all scenarios. This is because Figures 4 and 5 have similar plots, except that when the global QoE condition is stricter, each scheduler begins decaying earlier in terms of connected users, as expected. It is also possible to verify that PBF scheduler exceeds the PF scheduler for more number of connected users points than in Figure 4. The BET algorithm remains the best scheduler: supporting 40.21 connected users for P_{95}, 40.43 for P_{90}, and 40.66 for P_{80}. Regarding the remaining schedulers, the deviation between BEA and the MT and PF algorithms for P_{90} and P_{80} is slightly reduced: between +10.85 (regarding PF for P_{80}) and +15.76 (regarding MT for P_{90}).

Figure 9 presents the results in Figure 6 that solve (15) for each scheduler for $\alpha = \{80, 90, 95\}\%$. In Figure 9, unlike the previous scenarios, when incrementing the number of connected users, lower percentage of users meeting the QoE condition means a better scheduler performance. Based on Figure 9, it is possible to consolidate the BET algorithm as the best scheduler for every scenario, given this evaluation methodology: with 80.05 connected users for P_0, 80.87 connected users for P_5, and 81.02 for P_{10}. RR keeps performing significantly better than most of the remaining schedulers (from +49.98 connected users than MT, PF, BEA and PBF for P_5 to +65.95 than MT for P_{10}) and slightly lower than BET (from –2.3 connected users for P_{10} to –9.07 for P_0). The MT algorithm also remains the worst scheduler: with 21 connected users for P_0, 21.18 connected users for P_5, and 21.77 for P_{10}. Unlike the previous scenarios, the PBF algorithm is now in no case better than PF. Despite that, the other buffer-level-based algorithm, BEA, supports a considerable 74.63 connected users for P_{10}, which means that when the number of connected users increases, the number of unsatisfied users does not increase during a significant period (*cf.* Figure 6). At last, it is also interesting to notice that for P_0, *i.e.*, the maximum number of connected users each scheduler supports without having unsatisfied users, the MT, PF, BEA, and PBF algorithms obtained the same result. When there are few connected users, depending on the available bitrates, network configuration, and the number of connected users, the server is able to serve every user without much effort. Thus, there will be simulations where the server will have TTIs without requests to answer. When this happens, independently of the implemented scheduler, all users will have the opportunity to have their requests satisfied. After passing the number of connected users where this situation is verified, each algorithm metric plays a fundamental role in the scheduling decision, solving request conflicts among different users. By making some assumptions, and some simple deductions, it is possible to verify that for this simulation setup, this collision-point – the minimum number of connected users where the server always has multiple requests to attend – takes place when the number of connected users is approximately equal to 21.

C. QoE-driven Joint Admission Control and Scheduling

In the presented simulation results, the BET algorithm shows to be the fittest one for the 360º video streaming application. The literature on scheduling algorithms for 2D video streaming has shown that more complex algorithms, such as PF, generally allow for a better streaming performance than the RR and BET [10]. Unlike the 2D video streaming, for 360º video, algorithms with a higher level of fairness, such as RR and BET, behave better in 360º video streaming than algorithms with a higher level of opportunism, such as MT and PF. Due to the buffer size impact on the video streaming – which causes it to be more vulnerable to the staling phenomenon – the algorithms which somehow favor some users over others have the following symptoms: first, the users receiving the worst service – which typically are users with the worst connection conditions – are not allocated with enough RBs to prevent stalls; second, at the cost of keeping the service to the worst behaved users, the users receiving better service may, in some cases, not receive the required number of allocations for a satisfying service. A secondary reason for the BET performance is that as BET tries to provide the same service to every user, the visualized video qualities tend to be the same for every user despite the possible oscillations on the channel conditions of each one.

Consequently, the BET algorithm generally provokes fewer quality switches than the remaining algorithms.

In order to enhance the server operation a QoE-driven Joint Admission Control and Scheduling (QJACS) is proposed. Due to its requirements (*e.g.*, high video resolution, low latency, high sensibility to stalls) the 360º video streaming is highly demanding. Thus, services with global QoEs below the satisfaction level (global QoE < 3), *i.e.*, a good perceived quality by the client, should not be provided by the server. For this reason, the admission control proposed in this paper denies access to new users when for the current number of connected users, the percentage of satisfied users is below a given threshold. The proposed QoE-driven Joint Admission Control and Scheduling (QJACS) is presented in Figure 10.

The proposed admission control can be described as follows: each time a new user is requesting service to a server, the server starts a timer and, until the timeout instant, analyzes the hypothesis of not denying the new user access; then, for each iteration, obtains the percentage of users with a QoE above or equal a specific perceived quality level, P_α; if that percentage is higher or equal to a given threshold, β, it means there is probably enough capacity to serve a new user with also a service above the QoE threshold, α, and so, the access is conceded; if that percentage is below β, it means there is probably not enough capacity to provide that service level to the new user, and the access is denied. Given that, before the timeout instant, there are possibly users who are not requiring service anymore, the admission control algorithm repeats itself until it considers that: there are no conditions to ensure a given level of service to the new user, any currently connected user is finishing its service, and even if so, that is not enough to provide a sufficiently good service to the new user.

For the admission control parameters, a possible and reasonable configuration would be: α of 3, β of 90%, Δt of 1 millisecond, and a t_{out} of 1 second. After receiving the new user access request, during one second, the admission control would update at every millisecond the percentage of users which had not last reported a good or excellent QoE. When that percentage is higher or equal to 90%, the algorithm allows the new user access, if that does not happen, at the end of the one-second, the admission control denies its access.

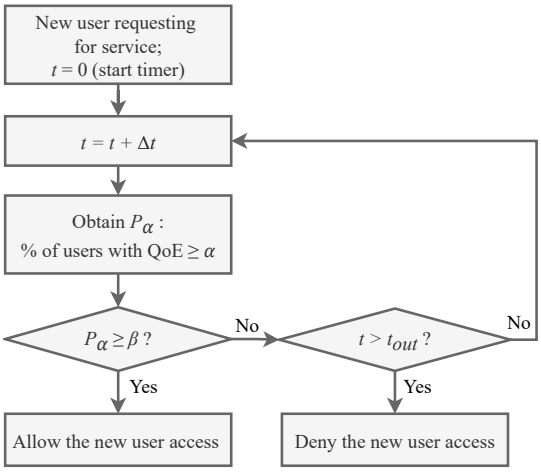

FIGURE 10: QJACS admission control flowchart.

Although this admission control algorithm can be jointly used with any scheduler, particularly for the 360º video streaming application, it should be used with the BET scheduler. In addition to being the scheduler with the best performance, it also presents characteristics that enhance and simplify the proposed admission control. The service provided by the server using the BET metric allows the convergence of the global QoE value of each user to approximately the same level. Since the admission control algorithm requires the QoE reporting of each user, the periodicity of each user reporting can be increased by using BET. This scheduler allows the assumption that, among the different connected users, the QoEs' reports are approximately equal for the same TTI. Hence, the server can take the admission control decision if only one user reports its QoE, making this a cooperative process. When this is the case, the admission control should have an alternative criterion instead of obtaining the percentage of the number of connected users satisfying the QoE condition. Figure 11 presents the simplified version of the admission control proposed for QJACS when the BET scheduling algorithm is adopted as the scheduler.

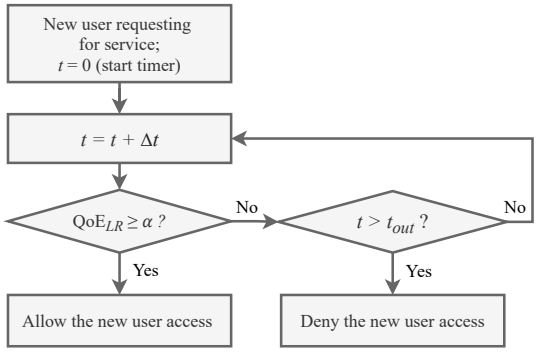

FIGURE 11: QJACS simplified admission control flowchart.

Depending on the comparison between the last reported QoE value, QoE_{LR}, and the QoE level threshold, α, the admission control decides the new user access. Note that, in the situation where the new user access is allowed, and the reported QoEs decay to a value below α, despite denying future admission requests if all users remain connected, this QoE decay is slight. As shown in Figure 12, for the simulation results obtained in Section V.B, when the BET scheduler is near its breaking point for the global QoE ≥ 3 (at 60 connected users), the QoE of each user does not drastically degrade from one point to another.

Accordingly, two conclusions can be drawn from Figure 12: first, it is reasonable to assume there is homogeneity among the users' QoEs. When the number of connected users is equal to 60 users, the measured samples are contained within the interval $[\bar{x} - 0.325, \bar{x} + 0.190]$, and for 61 users within $[\bar{x} - 0.340, \bar{x} + 0.213]$; second, when the admission control allows the access of a new user that provokes the general decay of the QoE below β, that decrease will probably not be perceived by each user. Comparing the server capacity between 60 and 61 connected users, despite the average QoE decreases from a good ($QoE \in [3, 4[$) to a fair ($QoE \in [2, 3[$) level, the actual means deviation equals 0.130 (*i.e.*, from $\bar{x} = 3.082$ when 60 users are connected to $\bar{x} = 2.952$ when 61 users).

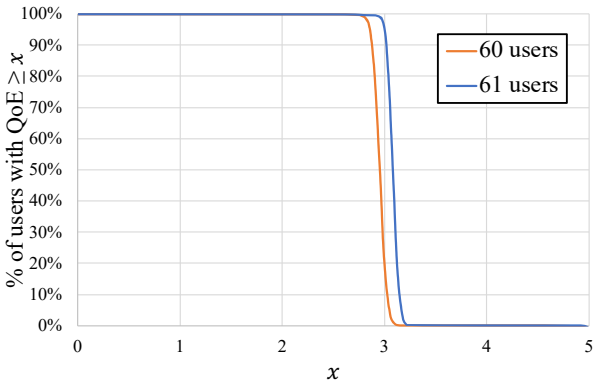

FIGURE 12: Distribution of the percentage of users with global QoE above x, given $x \in [0, 5]$.

V. CONCLUSION

With the arrival of 5G networks, the 360º video streaming application has gained the opportunity to improve its performance. One of the key challenges in any streaming service is the scheduling of the available RBs. Different scheduling algorithms may lead to very different results. In particular, some schedulers have higher levels of fairness, while others are more opportunistic, meaning the same service can be provided to all users, or, depending on each user's conditions, some users may be privileged over others.

From the study of the performances of conventional and of two new scheduling algorithms, this work concludes that the BET scheduler has an exceptional performance for the 360º video streaming context – supporting, for the considered test conditions, in average, a maximum of 60.03 connected users when 95% of the users have a good or excellent QoE, and, in average, 80.05 connected users without having users with bad service. The simulation results show that fairer algorithms, BET, RR, and BEA, perform better than opportunistic or relatively opportunistic ones, PBF, PF, and MT, respectively. In particular, despite not exceeding the RR and BET performances, the suggested BEA scheduler has the advantage of being less sensitive to the impact of increasing the number of connected users on the QoE level provided to them. At the same time, the BEA scheduler does not drastically reduce the capacities provided by RR and BET – supporting, for the considered test conditions, in average, a maximum of 48.62 connected users when 90% of users have a good or excellent QoE, and, in average, 74.63 connected users when 10% of users have a low or very low QoE.

As the BET scheduler performs exceptionally well for the 360º video streaming, the remaining work of this thesis focused on enhancing its behavior. In particular, the admission control procedure is revisited for this application's context and QJACS is proposed. From the perspective of the server and the MNO, it is not desirable to provide the 360º video streaming with low QoEs. Hence, the admission control is set to deny new users access when the current network status does not enable them to be served with a given level of satisfaction. This decision is remarkably easier to take when the BET algorithm is implemented as the server's scheduler, as its metric tends to provide service with the same level to every user – leading to the proposal of a simplified version of QJACS. Thus, based on the QoE reporting of one user, the admission control algorithm can decide if the current service is enough or not to provide a given level of satisfaction and, consequently, allow or deny a new user access. If an admission control similar to QJACS is not adopted by the server, the BET algorithm will unreasonably degrade the provided QoE as the number of connected users increases beyond its breaking point for a given satisfaction level.

Concerning future work, it would be interesting to study, for different network configurations, how the admission control could learn the QoE degradations when the number of connected users is variant. Given that QJACS is dependent on some model parameters, the network configuration of each real-world scenario requires their tuning. Instead of following a deterministic approach, where the MNO would have to estimate the model parameters previously, a learning approach could be implemented.

The QoE model should also be revisited, for its adaptation to the perceptual impact of having different tiles' resolutions within the same viewport; furthermore, the impact of stalling in the user's QoE is still scarce for the 360º video streaming.

Finally, it would also be interesting to introduce latency in the simulator development and study its impact on the streaming performance and, in particular, how the viewport orientation prediction would affect the QoE results. As the buffer size impacts so many video streaming components, it would also be interesting to analyze how its size could be optimized according to the trade-off between the viewport prediction and stalling prevention – as longer buffer sizes are less vulnerable to stalling but reduce the accuracy of the viewport prediction.

REFERENCES

[1] Ericsson, "EricssonbMobilebReport", 2020, [Online] Available: https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf [Accessed 14/02/2021].

[2] M. Zink, R. Sitaraman, and K. Nahrstedt, "Scalable 360° video stream delivery: Challenges, solutions, and opportunities", in Proceedings of the IEEE, vol. 107, no. 4, pp. 639-650, 2019.

[3] ITU-R, "Recommendation M.2083-0, IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond", 2015.

[4] ISO/IEC 23090-2:2019, "Information technology — Coded representation of immersive media — Part 2: Omnidirectional media format", January 2019.

[5] H. Yuan, S. Zhao, J. Hou, X. Wei, and S. K. Wong, "Spatial and Temporal Consistency-Aware Dynamic Adaptive Streaming for 360-Degree Videos", in IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 1, pp. 177-193, January 2020.

[6] H. L. Dieu Huong, D. CF. Nguyen, T. T. Huong, P. Ngoc Nam, and T. C. Thang, "Smooth Viewport Bitrate Adaptation for 360 Video Streaming", 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), pp. 512-517, 2019.

[7] Philip B. Kurland and Ralph Lerner, eds., *The Founders' Constitution.* Chicago, IL, USA: Univ. of Chicago Press, 1987, Accessed on: Feb. 28, 2010, [Online]. Available: http://press-pubs.uchicago.edu/founders/

[8] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. Turck, "QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming", ACM Trans. Multimedia Comput. Commun. Appl. 12, 2, Article 28 (March 2016), 2015.

[9] E. J. David, J. Gutiérrez, A. Coutrot, M. P. D. Silva, and P. L. Callet, "A Dataset of Head and Eye Movements for 360° Videos", in Proceedings of the 9th ACM on Multimedia Systems Conference (MMSys'18), Amsterdam, Netherlands, June 2018.

[10] F. Rodrigues, I. Sousa, M. P. Queluz, and A. Rodrigues, "A QoE-Aware Scheduling Algorithm for Adaptive HTTP Video Delivery in Wireless Networks", Wireless Communications and Mobile Computing, vol. 2018, 2018.