QoE-Based Scheduling Algorithms for Adaptive HTTP Video Delivery in LTE

Frederico Freire Rodrigues
Department of Electrical and Computer Engineering (DEEC)
Instituto Superior Técnico (IST)
Lisbon, Portugal
frederico.rodrigues@tecnico.ulisboa.pt

Abstract— In the last years, the multimedia content consumed by mobile users has exponentially increased, particularly video content. Mobile network capacity cannot be increased as fast as the demand growth. A possible solution is the development of intelligent schedulers that allocate resources very efficiently and satisfy the maximum possible number of clients, providing them a good quality of experience (QoE). This work proposes a new and effective solution, Maximum Buffer Filling (MBF), that increases the number of users who are satisfied with their streaming session in a Long Term Evolution (LTE) network. To do so, it makes use of the user's current buffer level reported by the clients which implement the MPEG-DASH specification, the reported radio channel status to the base station (BS) and the bitrate of the requested video segment. MBF allocates resources according to the current buffer level of the users and their achievable buffer filling. It can operate in two modes: to maximize the number of satisfied users or to minimize the number of non-satisfied ones. To ensure that 90% of the users have a good or excellent QoE, MBF regardless the mode in which it operates supports clearly a higher number of users streaming video than Round Robin (RR), Blind Equal Throughput (BET), Proportional Fair (PF) and Proportional Fair with Barriers for Frames (PFBF). With respect to RR, it supports more 15% and 12% of satisfied users, when operating in mode 1 and mode 2, respectively. Furthermore, MBF leads also to a smaller number of users with a bad or poor QoE than RR, PF and PFBF, regardless the mode in which it operates.

Keywords — mobile video streaming, DASH, scheduling algorithms, quality of experience, throughput, buffer.

I. INTRODUCTION

Nowadays, users are constantly downloading video streams, using video conferencing applications, or even broadcasting their own video streams to the Internet through social applications, congesting the network. Furthermore, users' expectations about the services provided are getting higher. However, in a wireless network environment, there are problems such as background noise, narrow frequency spectrum and varying degrees of network coverage and signal strength. These result in undesired effects such as long initial buffering times, video pauses (stalls), long re-buffering periods, frequent changes in video quality, frame rate drops and/or in audio/video desynchronization, degrading user's QoE. Thus, it is necessary to develop intelligent and efficient QoE-aware scheduling algorithms. A lot of research has been done in this field. However, there are very few studies that analyze the capacity of the schedulers regarding the number of users that can be streaming video and have a certain level of QoE. In this context, in this work it is analyzed the capacities of some of the most well-known schedulers. Furthermore, it is developed a scheduling algorithm whose metric uses the current buffer levels reported by the mobile clients who implement the MPEG-DASH specification. The simulation results obtained in this work provide an assessment of the capacities of the algorithms regarding the number of users who are satisfied and non-satisfied with their streaming session, as a function of the total number of connected users streaming video.

Section II presents a QoE overview: the factors that influence QoE and the existing QoE measurement methods. Section III reviews the most relevant concepts of MPEG-DASH specification: its architecture and the QoE metrics that the clients report to the BS. Section IV presents an overview of the LTE network. Section V presents the most well-known QoE-unaware and QoE-aware schedulers. Section VI presents the proposed scheduling algorithm. Section VII describes the simulator developed to assess the performance of the schedulers in study. Section VIII presents the performance assessment methodology, the simulation parameters and the assessment results. Finally, Section IX concludes this work with a summary and suggestions for future work.

II. QUALITY OF EXPERIENCE OVERVIEW

QoE has been defined as "the degree of delight or annoyance of a person experiencing an application, service, or system. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application, service or system in the light of the person's personality and current state" [1]. Clearly users' expectations about the services provided are getting higher. Thus, it is important to know what influences QoE, how QoE can be measured and improved.

A. Factors influencing QoE streaming video

QoE is usually expressed using Mean Opinion Score (MOS) where people evaluate a video after compression and transmission, scoring it from 1 ("Bad") to 5 ("Excellent"). The fact is that two different persons will probably perceive a video streaming experience in different ways. This may be due to:

- Human factors: user characteristics that can affect QoE (e.g. gender, age, mood or visual and auditory acuity);
- System factors: factors related to the transmission system (codec, packet loss rate, delay or screen resolution);
- Context factors: external factors that affect QoE (e.g. the time of the day and the day of the week or service type.

B. Quality of experience measurement methods

Clearly, QoE can be very subjective so it is relevant to know how QoE can be measured. The most used metric for accessing QoE is MOS which can be assessed based on two approaches:

1) *Subjective assessment*: It is the only reliable and most accurate method for assessing perceived QoE. The outcome of any subjective experiment are quality ratings from viewers, which are then averaged into MOS. However, MOS is very expensive and time-consuming. Furthermore, it is impractical in certain types of services such as live streaming.

2) *Objective assessment*: through QoE models it is possible to predict the QoE based on objective metrics such as:

- Initial buffering time: it measures the period between the starting time of loading a video and playing it.

- Mean re-buffering duration: when there are no more data in the buffer, the playback pauses, and the player enters into a re-buffering state. This metric measures the average duration of a re-buffering event.
- Re-buffering frequency: this metric measures the frequency of the re-buffering events.
- Spatial resolution: generally higher frame resolutions are perceived as better (e.g. 1080p vs. 640 x 480 SD).
- Video quality changes: perceptual quality is also impacted by the number of image quality changes during playback, especially those from a higher to a lower.
- Frame rate: higher frame rate is more essential for high-motion scenes, such as a sporting event.
- Audio/video synchronization: audio that is out of sync with the video may be very annoying.

III. DASH Overview

Especially in a wireless network environment, there are problems such as background noise, varying degrees of network coverage and signal strength. These may lead to re-transmissions causing delays, degraded video quality and re-buffering events, affecting user's QoE. Thus, technologies like Adaptive Bitrate Streaming (ABS) have been developed. Its mechanism consists of changing the bitrate according to currently available resources (bandwidth, battery charge left, screen size, etc.). ABS is built into technologies and products like Adobe HTTP Dynamic Streaming (HDS), Apple HTTP Live Streaming (HLS) and Microsoft Smooth Streaming. However, these solutions are closed systems with its own manifest formats, content formats and protocols. To overcome this lack of interoperability, MPEG developed the MPEG-DASH specification.

A. Scope of MPEG-DASH

Figure 1 illustrates a simple streaming scenario between an HTTP server and a DASH client. Only the orange blocks are defined by the specifications and are responsible for creating content and handle metadata, such as resources location and format. The green elements are outside of DASH scope. So, the delivery of the metadata file and media encoding formats containing the segments is not specified. This also applies to the client's action for fetching, adapting and playing the content [2].

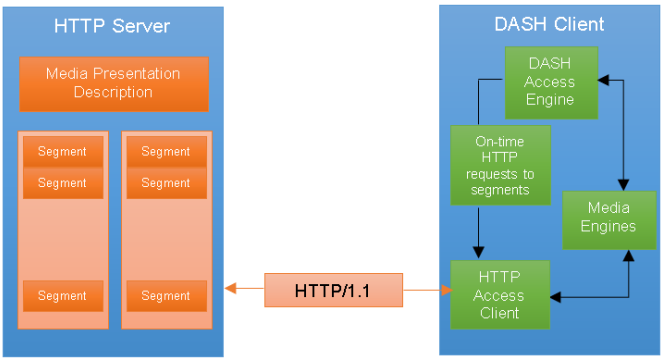

Figure 1 – MPEG-DASH system description.

The server holds variable encoded media data of the same content that is formed by a content creator generator. DASH specifies how content is produced and how metadata is handled. The content exists on the server in two parts:

- Segments: contain the actual multimedia bit streams, in single or multiple files. A segment belongs to a certain representation that have a certain bitrate.
- Media presentation description (MPD): a XML document that describes the temporal and structural relationships between segments. The MPD allows the user to learn about the program timing, media-content availability, resolutions, minimum and maximum bandwidths, and the existence of various encoded alternatives of multimedia components

DASH does not prescribe any client-specific playback functionality; rather, it just addresses the formatting of the content and associated MPD [3]. DASH client monitors the network bandwidth, selects the appropriate encoded alternative and starts streaming the content by fetching the segments using HTTP GET requests, according to the Figure 2.

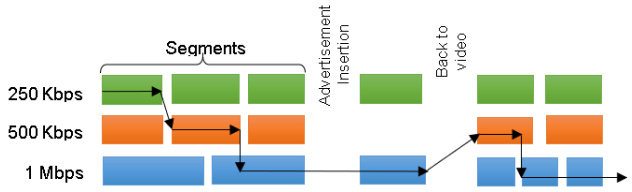

Figure 2 – Client streaming process.

B. Quality of Experience in DASH

The development of evaluation methodologies, performance metrics and reporting protocols play a key role for optimizing the delivery of HAS services. In addition, QoE monitoring and feedback are beneficial for detecting failures, managing streaming performance, enabling intelligent client adaptation (useful for device manufacturers), and allowing for QoE-aware network adaptation and service provisioning (useful for the network operator and content/service provider). Having recognized these benefits, MPEG body have adopted QoE metrics as part of their DASH specifications. The Metrics element in the MPD contains the list of DASH Metrics for which the measurements are desired, the time interval and the granularity for the measurements and the scheme according to which the metric reporting is desired. Some of the metrics are:

- List of representation switch events - it reports a list of representation switch events that took place during the measurement interval.
- Average throughput - it indicates the average throughput that is observed by the client during the measurement interval.
- Initial playout delay - it signals the initial playout delay at the start of the streaming of the presentation.
- Buffer level - it provides a list of measurements of the buffer level carried out during playout at normal speed.
- Device information - it informs about the displayed video resolution and physical screen characteristics.

IV. LTE Network Overview

Regarding the LTE network, radio resources are allocated into the time/frequency domain. In the time domain, the total time is split in 10 ms frames, each one composed of 10 consecutive sub-frames or Transmission Time Intervals (TTI), each one lasting 1 ms. Furthermore, each TTI is made of 2 time slots with length 0.5 ms. Each time slot corresponds to 7 OFDM symbols. In the frequency domain, the total bandwidth is divided into sub-channels of 180 kHz. A time/frequency radio resource spanning over 1 time slot in the time domain and over one sub-channel in the frequency domain is called Resource Block (RB) (see Figure 3). On the frequency domain, it occupies 12 sub carriers' space which corresponds to 180 kHz (12×15 kHz). In time domain, it occupies one slot which is half of a sub-frame (0.5 ms). Each RB carries 12×7 symbols.

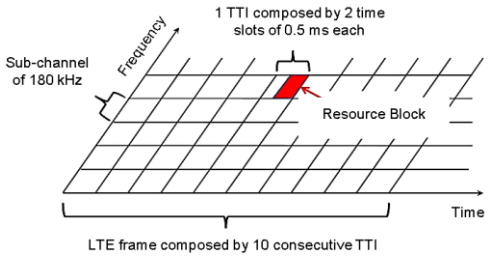

Figure 3 – Time-frequency radio resources grid [4].

Although RBs are defined over 1 time slot, the basic time-domain unit for scheduling in LTE is one sub-frame, consisting of two consecutive slots. This means that the scheduler makes allocation decisions every 1 ms (or TTI). The reason for defining the resource blocks over one slot is that distributed downlink transmission and uplink frequency hopping are defined on a slot basis [5]. The minimum scheduling unit, consisting of two time-consecutive RB within one sub-frame (one RB per slot), is referred as a resource-block pair [5]. The bandwidths defined by the standard are 1.4, 3, 5, 10, 15, and 20 MHz. Table 1 shows how many RBs there are in each bandwidth.

Table 1 - Number of resource block pairs for each bandwidth.

Bandwidth [MHz]	Resource-block pairs per sub-frame (per 1ms)
1.4	6
3	15
5	25
10	50
15	75
20	100

For example, the 20 MHz bandwidth has 200 RB available per sub-frame. However, there are only 100 addressable locations of 2 RB each. This means that for each user, a minimum of 2 RB (1 RB pair) can be allocated per TTI. A maximum of 100 users can be served per TTI. Therefore, from now on, a resource-block pair is going to be called simply a RB.

It is also important to mention that in a LTE network all the users who are connected to a BS report a parameter that keeps the base station informed about users' channel conditions, the Channel Quality Indicator (CQI), periodically. The minimum periodicity is 2 ms for FDD and 1ms for TDD. The efficiencies are presented in Table 2, from a CQI=0 to a CQI=15 [6].

Table 2 - Efficiencies for each CQI.

CQI index	Modulation	Efficiency (eff) [bits/symbol]
0	Out of range	
1	QPSK	0.1523
2		0.2344
3		0.3770
4		0.6016
5		0.8770
6		1.1758
7	16QAM	1.4766
8		1.9141
9		2.4063
10	64QAM	2.7305
11		3.3223
12		3.9023
13		4.5234
14		5.1152
15		5.5547

V. SCHEDULING ALGORITHMS

Nowadays, mobile operators have two challenges. On the one hand, they need to satisfy the high expectations that customers have on the delivered quality of the services. On the other hand, mobile network capacity cannot be increased as fast as the demand growth. Among possible strategies, scheduling algorithms have an important role. Scheduling is conducted by allocating resources to users who want to download or upload content from or to the internet. Resource allocation is usually based on the comparison of per-RB metrics: the k^{th} RB is allocated to the j^{th} user if his metric $m_{j,k}$ is the biggest one among all users, i.e., if it satisfies the equation:

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

The $m_{i,k}$ metric can depend on radio channel condition, buffer state, resource allocation history, etc. The choice of metric results in providing a higher or lower throughput to a given user and thus in a certain level of satisfaction depending on his system requirements and expectations.

A. *QoE-unaware Schedulers*

1) *Channel-unaware strategies:* these strategies are unrealistic because they do not consider the channel condition, ignoring that the channel quality is time-varying. This fact makes them unsuitable in mobile networks. However, some of them are the basis of other more complex algorithms. The most well-known channel-unaware scheduler is the following:

- Round robin (RR): it allocates radio resources cyclically to the users, according to the metric:

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

where t is the current time and T_i is the last time when i^{th} user was served. The users are served one after another. The principal advantages of RR are the guaranty of fairness for all users in terms of the amount of RB allocated to each user and an easy implementation. However, it is unfair in terms of user throughput and QoE.

2) *Channel-aware / QoS-unaware strategies*: Channel-awareness is a fundamental concept for achieving high performance in a wireless environment, and it can be used by exploiting the channel quality indicator reporting. If one can estimate the channel quality perceived by a user on a given RB, in fact, it is possible to allocate radio resources obtaining very high data rate. To this aim, the most significant parameter is the expected achievable throughput for the i^{th} user at the t^{th}TTI over the k^{th} RB, i.e. $d_k^i(t)$. For a bandwidth B, it depends on signal-interference plus noise ratio (SINR) and can be calculated using the Shannon expression for the channel capacity [7]:

$$d_k^i(t) = B.log(1 + SINR_k^i(t)) \quad (3)$$

Nevertheless, spectral efficiency is not the unique objective for a mobile operator because it should be able to guarantee a minimum quality level of service also to cell-edge users. The following scheduling algorithms are some of the most well-known channel-aware/QoS-unaware strategies:

- Blind Equal Throughput (BET): although this algorithm does not estimate the expected data-rate $d_k^i(t)$, it considers the channel quality by storing the past average throughput achieved by each user and using it as metric to calculate user priority for scheduling. BET provides throughput fairness among all users regardless of their current channel conditions. BET allocates RBs according to the metric:

$$m_{i,k}^{BET} = \frac{1}{\overline{R_i(t)}} \quad (4)$$

where $\overline{R_i(t)}$ is the past average throughput at time t. It is updated every TTI for each user.

- Proportional Fair (PF): this scheduler provides a good compromise between throughput and fairness among the users. While trying to maximize the total throughput, it tries at the same time to provide a minimum quality of service to all the users, ensuring that no user is starving. Its metric is:

$$m_{i,k}^{PF} = \frac{d_k^i(t)^\alpha}{\overline{R_i(t)}^\beta} \quad (5)$$

The parameters α and β tune the "fairness" of the scheduler. They adjust the balance between serving the users with the best channel conditions often and serving the ones who achieved lower throughputs often enough that they have an acceptable level of service quality.

3) *Channel-aware / QoS-aware strategies:* these strategies consider both the communication channel conditions and the different service requirements. Different applications may have different requirements. Some may demand a guaranteed minimum data rate for users whereas others may require a minimum delay when delivering the packets such as Voice over IP (VoIP) or video live streaming:

- Schedulers for guaranteed data-rate: these schedulers are solutions for flows requiring guaranteed data-rate.
- Schedulers for guaranteed delay-requirements: these scheduling algorithms aim at guaranteeing that the packets are delivered within a certain deadline.

B. QoE-aware Schedulers

QoE-aware schedulers somehow take into account the QoE of the users are have using a certain service, trying to optimize it. Although some of them can be very complex, they can be very interesting for both the operators and end-users. In the end, the operators want to provide a certain level for QoE (a good one) for the largest number of users. According to [8], QoE-aware scheduling strategies can be divided in three sub-groups according to the role of the user and his device in the scheduling process: passive end-user device strategies, active end-user device / passive user strategies and active end-user device / active user strategies (see Figure 4).

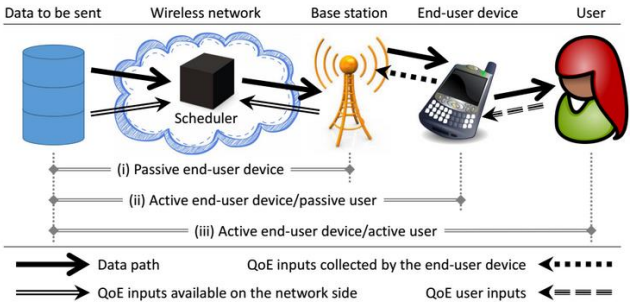

Figure 4 – QoE-oriented schedulers classification [9].

(i) *Passive end-user device strategies*: in this strategies neither the user nor his device needs to measure, report or monitor any parameter. The needed measurements can be carried out on the BS side. However, many metrics cannot be known at the BS side and although some can be estimated, these strategies do not lead to the maximum QoE performance.

In [10], a QoE-aware scheduler for HTTP Progressive Video in Orthogonal Frequency-Division Multiple Access (OFDMA) is presented. The authors propose the use of a metric that directly affects the end-user experience, namely an estimation of the users' buffer levels, for resource allocation. It is important to note that if the buffer level is reported from the client device the accuracy of the estimation could be improved but thus it would not be a passive end-user device strategy.

In [11], the authors focus on a QoE-driven cross layer optimization for wireless video delivery. Their goal is to provide a mechanism that allow the network operators to dynamically adapt between achieving a maximum system efficiency and achieving the minimum unfairness of perceived quality of the services between the users. They define the system efficiency as the total sum of the MOS values perceived by all users and unfairness as the perceived quality difference between the user experiencing the highest MOS and the user experiencing the lowest MOS. MOS is calculated using the metric Video Structural Similarity (VSSIM).

(ii) *Active end-user device / passive user strategies*: these schedulers consider information that is feedbacked from the users' device. The user does not have an active role. In this type of schedulers are inserted the DASH-based ones. For example, clients implementing DASH specification can send some metrics to the base station such as the buffer level, initial playout delay, average throughput and others.

In [12], the authors propose a QoE-aware radio resource scheduler, the Proportional Fair with Barrier for Frames (PFBF). It acts like a proportional fair until some user's reported buffer level is below a certain threshold, forcing the scheduler to serve this user first. The priorities among users for resource allocation are adjusted based on the dynamic feedback of re-buffering percentage. The first user to be served by the scheduler is the one who has the highest metric $m_{i,k}$:

$$m_{i,k}{}^{PFBF} = V_i\{\frac{\alpha \times d_k^i(t)}{S_{frame,i}} \times e^{\beta(f_{min}-f_i)} + \frac{d_k^i(t)}{\overline{R_i(t)}}\} \quad (6)$$

$$V_i = \begin{cases} 1 + \dfrac{n \times p_{rebuf,i}}{\sum_{j=1}^{n} p_{rebuf,j}} & if \sum_{j=1}^{n} p_{rebuf,j} > 0 \\ 1 & otherwise \end{cases} \quad (7)$$

where $d_k^i(t)$ is the expected achievable throughput, $S_{frame,i}$ is the size of the frame in transmission, f_i is the number of frames in the buffer, $\overline{R_i(t)}$ is the average throughput of the i^{th} user, and f_{min} is tunable, representing the minimum number of frames in the buffer. A higher value of f_{min} implies higher fairness to users in terms of their playback buffer. As long as the video frames in the playback buffer f_i exceed the minimum value of f_{min} PFBF allocates approximately like the PF. When f_i drops below the f_{min}, then the user metric increases, increasing the probability of being served. V_i includes fairness in terms of re-buffering percentage. $p_{rebuf,i}$ is the percentage of the total streaming time spent re-buffering and n is the total number of connected users. α and β are tunable parameters.

(iii) *Active end-user device / active user strategies*: these strategies receive both inputs from the user's device and from the user himself. The users have an active role and can influence the way the scheduler allocates the RBs through their actions. For example, for video streaming application the users may feedback their QoE at the end of the video. The user interaction with the video (for example the number of times the user pauses the video or changes the resolution) may be also traced.

VI. PROPOSED QOE-AWARE SCHEDULING ALGORITHM

In [13], it is shown using subjective evaluations that the users of the experiment preferred a fluent playback of the video above a higher resolution, frame rate and bitrate. The idea was

to develop a scheduler based on the users' buffer levels, a QoE metric that clients implementing the MPEG-DASH specification may report. The scheduler should firstly try to avoid a re-buffering event. If a stall happens, the re-buffering period must be as short as possible. So, the first idea of the proposed scheduler, Maximum Buffer Filling (MBF), was to simply allocate RBs to the users with lower buffer levels:

$$m_{i,k}^{MBF-1^{st}Attempt} = \frac{1}{Buffer_i(t)} \quad (8)$$

However, this metric does not consider directly the current channel condition of the user, which is a waste of useful information. For example, it would be inefficient to allocate resources to a user who has momentarily a poor channel condition just because he has the lowest buffer level among the connected users. Although he has the lowest buffer level, he may have several seconds of video stored in the buffer, allowing the user to play the video smoothly during some seconds even if he is not served while he has a poor channel condition. Why not take advantage of the reported CQIs and serve the user with the second lowest buffer but that has a great CQI? Furthermore, sending many bits (by having the highest throughput) may not mean to greatly increase the buffer level. If the user is requesting for a very good video quality segment, the buffer level increases very slowly. For instance, consider several users who have the same buffer level, a low one. To avoid stalls, it is more efficient to allocate the resource blocks to the users whose channel conditions and requested bitrates allow them to store a higher number of milliseconds in his buffer, avoiding it at least for these users. Thus, to consider both the first idea (take advantage of the user channel conditions) and the second one (take advantage of knowing the users' requested video bitrates), it came intuitively that maybe it would be a good idea to not only consider the current buffer level (denominator) but also to measure how much the buffer is increasing if the resources blocks are allocated to the user. Hence, the numerator would be the quotient between achievable throughput and requested video bitrate. The metric would be:

$$m_{i,k}^{MBF-2^{nd}Attempt} = \frac{\frac{U_{throughput}^{i}(t)}{U_{RequestedVideo}^{ji}(t)}}{Buffer_i(t)} \quad (9)$$

where $U_{RequestedVideo}^{i}(t)$ is the bitrate of the video that is being downloaded at time t and $U_{throughput}^{i}(t)$ is the achievable throughput by the ith user. After a small number of experiments, the final MBF metric is the following:

$$m_{i,k}^{MBF} = \begin{cases} \frac{\Delta U_{buffer}^{i}(t)}{Buffer_i(t)}, & if\ \forall i\ Buffer_i(t) > \alpha \\ \frac{1}{Buffer_i(t)}, & otherwise \end{cases} \quad (10)$$

$$\Delta U_{buffer}^{i}(t) = \frac{U_{throughput}^{i}(t)}{U_{RequestedVideo}^{i}(t)} - 1 \quad (11)$$

where $Buffer_j(t)$ is the number of milliseconds of video stored in the i^{th} user's buffer and $U_{RequestedVideo}^{i}(t)$ is the bitrate of the video that is being downloaded and $U_{throughput}^{i}(t)$ is the achievable throughput at time t given the i^{th} user's CQI. $\frac{U_{throughput}^{i}(t)}{U_{RequestedVideo}^{i}(t)}$ is the predicted number of milliseconds that will be possible to transmit to the i^{th} user if the resource blocks are allocated to him. $\Delta U_{buffer}^{i}(t)$ is the buffer variation in an interval of 1 ms. It is equal to the number of milliseconds of video sent to the user less 1 ms that the user plays every 1 ms spent. The term $\Delta U_{buffer}^{i}(t)$ increases the network efficiency by increasing the probability of serving the user with better channel capacity. The user with better CQI has a higher achievable throughput, $U_{throughput}^{i}(t)$. If this term was used solely without the denominator $U_{RequestedVideo}^{i}(t)$, it would be maximizing the number of bits transmitted but not the number of milliseconds transmitted. By dividing the achievable throughput $U_{throughput}^{i}(t)$ by $U_{RequestedVideo}^{i}(t)$, the term $\Delta U_{buffer}^{i}(t)$ maximizes the sum of the number of milliseconds of video stored in the users' buffers, each TTI. The term $U_{RequestedVideo}^{i}(t)$ increases also the fairness among the users in terms of video quality served. The lower the quality that is being served to the i^{th} user, the higher the probability of being served. The term $Buffer_i(t)$ normalizes the number of milliseconds possible to transmit, using the number of milliseconds that are stored in the client's buffer, providing fairness in terms of buffer occupancy and trying also to avoid re-buffering events or buffer overflows.

If there are one or more users with a buffer level $Buffer_i(t) < \alpha$ (and if the user is not in the initial buffering state), the scheduler serves the user who has the lowest buffer level, using a metric equal to $\frac{1}{Buffer_i(t)}$. This is an emergency allocation to avoid a stall. When the user buffer level drops bellow α, he has the lowest buffer and must be served.

If all buffer levels are above the threshold α, the served user is the one who has the chance to fill his buffer with a bigger quantity of milliseconds, normalized to his current buffer level. The MBF pseudo-code is the following:

VII. Simulator Implementation

In order to test the new scheduling algorithm proposed in this work, a Java program was developed using an integrated development environment, Eclipse. This program simulates a LTE network scenario where several mobile users which implement the MPEG-DASH specification are streaming video. The program was designed in a modular form, so as to facilitate the process of implementing additional scheduling algorithms, QoE models and rate adaptation algorithms for requesting video segments. The main class has a "for loop" where the number of iterations is equal to the number of milliseconds that one wants to simulate. In each iteration the allocation algorithm calculates the metric for all users and allocates the resource blocks to one single user, the throughput and QoE until the moment of each user are calculated, the buffer is updated and it is verified if any user requested for a new segment. If the user with the highest metric does not need all the available resource blocks to finish receiving the entire requested video segment, then the user with the second highest metric receives the remaining RB as long as he needs them and so on.

A. User CQI, throughput and buffer

It is considered that there are no losses during the transmission of data to the users. However, the channel quality must vary according to its conditions. The variability of the channel conditions affects the number of bits that can be transmitted per resource block. This variability makes it essential to have a good scheduler capable of maintaining a

good level of QoE to all users while maintaining a high bandwidth efficiency. The BS has access to this information through the periodic CQI reports sent by users. In the simulator, all the users report their CQIs to the base station every 5 milliseconds, as it is used in [14]. The CQIs of the users were obtained through a simulation that was made using OMNeT++ [15, 16]. In addition, SimuLTE [17, 18] tool was used which is an open source project that is built on top of OMNeT++ and INET Framework. It is a simulation tool enabling complex system level performance-evaluation of LTE and LTE Advanced networks (3GPP Release 8 and beyond). A scenario with a single base station and 200 mobile users moving around with a random trajectory was simulated, for three minutes. The CQIs reported by the users to the base station were saved. There are users that report always CQI=15 during all the entire simulation but there are also users whose CQI varies between 2 and 15, so there is heterogeneity. The users created in the developed simulator simulate randomly the channel of one of the 200 users simulated in OMNET++.

In each TTI the scheduler assigns an amount of resource blocks to the i^{th} user, and according to the number of resource blocks allocated ($NbrRB_i$), user's CQI_i and the number of antennas used ($NbrA$), the user has a certain throughput:

$$U^i_{throughput}[bits/s] = NbrRB_i \times 1000 \times 12 \times 7 \times 2 \times eff[CQI_i] \times NbrA \times 0.75 \quad (12)$$

where $eff[CQI_j]$ is the RB efficiency which depends on the i^{th} user reported CQI, CQI_i. These efficiencies are presented in Table 2. It is also considered 25% of overhead [19].

User's buffer is measured in milliseconds. Every 1 millisecond spent the buffer leaks out 1 millisecond of video and is filled with a certain number of milliseconds of video, according to the user throughput and requested video bitrate. Except when the user is requesting the initial segments (initial buffering) or recovering from a stall (re-buffering event) and thus the buffer is not leaking out 1 ms of video every millisecond spent, the buffer variation is given by:

$$\Delta U^i_{buffer}[ms] = \frac{U^i_{throughput}}{U^i_{RequestedVideo}} - 1 \quad (13)$$

$U^i_{RequestedVideo}$ is the bitrate of the segment that the user requested and that is being downloaded.

B. QoE adaptation algorithm

The implemented QoE adaptation algorithm in the DASH client is QoE-enhanced Adaptation Algorithm over DASH (QAAD), presented in [20], and consists of two parts:

1) *Bandwidth estimation part:* the available bandwidth at time t, denoted by $bw_{est}(t)$, is estimated every θ seconds. It uses the previously calculated bandwidth estimated, $bw_{est}(t-\theta)$. Since the channel quality is time-varying, the estimated bandwidth is smoothed by means of a weight moving average:

$$bw_{est}(t) = w \times bw_{est}(t-\theta) + (1-w) \times bw_{sample} \quad (14)$$

where w is the weight factor for sampled bandwidth, bw_{sample} ($0 < w < 1$). The bw_{sample} is the available bandwidth, sampled in every θ seconds and computed using equation:

$$bw_{sample} = \frac{K}{\theta} \quad (15)$$

where K is the amount of data downloaded during θ seconds.

2) *Bitrate selection part:* the video quality of the next segment is selected, denoted by l_{next}, based on the available estimated bandwidth and the buffer length, $B(t)$. Furthermore, l_{best} represents the highest video quality that does not exceed the estimated bandwidth. The algorithm pseudocode is:

1: if $l_{best} == l_{prev}$ **then**
2: $l_{next} = l_{prev}$
3: else
4: if $l_{best} > l_{prev}$ **then**
5: if $B(t) > \mu$ **then**
6: $l_{next} = l_{prev} + 1$
7: else
8: $l_{next} = l_{prev}$
9: end if
10: else
11: if $l_{best} < l_{prev}$ **then**
12: $k = 0$
13: do
14: $t_{l_{prev}-k,\sigma} = \frac{B(t)-\sigma}{1-bw_{estimated} / b(l_{prev}-k)}$
15: $n_{l_{prev}-k} = \frac{t_{l,\sigma} \times bw_{estimated}}{\tau \times b(l_{prev}-k)}$
16: $k = k + 1$
17: while $n_{l_{prev}-k} < 1$ && $k < l_{prev} - 1$
18: end while
19: $l_{next} = l_{prev} - k$
20: end if

Initially the algorithm tests if the current best quality reachable is equal with the previous one ($l_{best} == l_{prev}$, line 1). If it is, no change needs to be made ($l_{next} = l_{prev}$, line 2). Then, it is tested the possibility of increasing the bitrate for the next segments. If $l_{best} > l_{prev}$ (line 4), the current buffer length is considered due to possible unpredictable throughput fluctuation. The video quality requested is incremented ($l_{next} = l_{prev} + 1$, line 6) if the current buffer available is larger than a certain marginal buffer length μ ($B > \mu$, line 5). Otherwise, the quality of the requested segments does not change ($l_{next} = l_{prev}$, line 8). Finally, if $l_{best} < l_{prev}$, the previous video quality cannot be kept. To avoid lowering more than two levels and this way avoiding a QoE reduction, the algorithm tries to keep the next video quality comparable to the previous one. To this end, the elapsed time to deplete the segments in the playback buffer, $t_{l,\sigma}$ and the expected number of segments downloaded during the time, n_l, should be first determined. The deduction of $t_{l,\sigma}$ and n_l can be found in [20]. In lines 13-18, the algorithm searches for the maximum feasible quality level lower than l_{prev}. The loop terminates when it founds a quality level $l_{prev} - k$ that is feasible, i.e., $n_{l_{prev}-k} \geq 1$.

Analyzing the algorithm QAAD it was found that if $l_{best} < l_{prev}$ (line 11), once inside the loop (lines $13-17$) the resulting $k \geq 1$ because k is always incremented at least once (line 16). Thus, the video quality will always decrease at least 1 level (line 19, $l_{next} = l_{prev} - k$). Then, to make possible the behavior of the fluctuation test presented in [20], the line 19 was modified to $l_{next} = l_{prev} - (k-1)$.

C. QoE model

In order to know if the connected users streaming video are satisfied with the service, it is necessary to use a model capable

of estimating the QoE of each user. In video streaming there are some parameters that affect the QoE. The average served quality, the standard deviation of the served quality, the frequency of changes of video served quality, the frequency and mean duration of re-buffering events and the initial delay affect QoE and are the most used parameters to calculate the QoE, in the literature. In [21], a QoE model is proposed:

$$QoE_i = 5.67 \times \frac{\bar{q}_j}{q_{max}} - 6.72 \times \frac{\hat{q}_j}{q_{max}} + 0.17 - 4.95 \times F_i \quad (16)$$

$$F_i = \frac{7}{8} \times \max\left(\frac{\ln(\phi_i)}{6} + 1\,, 0\right) + \frac{1}{8} \times \frac{\min(\varphi_i\,, 15)}{15} \quad (17)$$

$\bar{q}_i$ is the average video served quality to i^{th} user, q_{max} is the highest video quality level available in the server (or the number of quality levels in the server), $\hat{q}_i$ is the standard deviation of the video served qualities to user, ϕ_i is the frequency of re-buffering events and φ_j is the mean duration of these re-buffering events [21]. However, this QoE model presented does not consider the initial buffering delay. To consider this parameter, it is considered that the initial delay is not a re-buffering event and thus does not affect the re-buffering frequency. Nevertheless, the period of initial buffering is added to the duration of re-buffering events, affecting this way the mean duration of re-buffering events. Table 3 classifies the quality of experience the i^{th} user had streaming the video.

Table 3 - Perceived quality by the i^{th} user.

QoE_i	Perceived quality
$4 \leq QoE_i \leq 5$	Excellent
$3 \leq QoE_i < 4$	Good
$2 \leq QoE_i < 3$	Fair
$1 \leq QoE_i < 2$	Poor
$0 \leq QoE_i < 1$	Bad

So, it is considered that i^{th} user had an excellent experience streaming the video if his QoE is between 4 and 5. He had a bad experience if his QoE is between 0 and 1, and so on.

D. Implemented scheduling algorithms

Among all the studied schedulers studied, RR, BET, PF and PFBF were chosen to make the comparison with MBF. Each of these algorithms has a reason to have been chosen:

- RR serves as a reference in the comparison because it is the simplest scheduler. It allocates the same number of RB to all users. It is a good scheduler as long as the users channel conditions are similar to each other and do not vary much along the time. Otherwise, the RBs keep being allocated equitably to the users and some may suffer from starvation.
- BET allocates the RB to the user which has the lowest average throughput so far. It is the QoE fairest algorithm providing the same QoE to all the users, if they implement the same rate adaptation algorithm. This is achieved by serving always the user with the lowest average throughput. However, BET is very sensitive. A few users with very bad channel conditions need a lot of RB and this can greatly affect the users' QoE. Once MBF is a QoE-aware scheduler and it seeks to increase the number of satisfied users and minimize the number of non-satisfied users, BET is relevant.
- PF tries to take advantage of the users' channel conditions. The PF metric multiplies the BET's metric by the user achievable data-rate. This avoids the BET's problem stated before because not only the average throughput is considered but also the current achievable throughput, avoiding the starvation of the users with bad channel conditions. Furthermore, PF is largely used in the literature.
- PFBF modifies PF including fairness in terms of re-buffering percentage [12]. PFBF leads to smaller re-buffering periods. Note that PFBF requires feedback of each user's buffer level and re-buffering percentage. Such as PFBF, MBF's metric uses the buffer level trying to avoid stalls and decrease the re-buffering periods, so PFBF is used in this work

VIII. SIMULATIONS AND RESULTS

In this work, a maximum confidence interval size of 1% of the mean (w=0.01), with a probability of 95% are assumed. The number of simulations made guarantee with 95% of certainty that the real solutions, relatively to the values that are presented in the Table 4, Table 5, Figure 8 and Figure 10do not differ by more than 1% from the presented values, according to the method of Monte Carlo. The values presented were calculated doing liner interpolations between two points.

A. Simulation Parameters

The simulation consists of three minutes of resources scheduling. During this period, users are streaming video, being continuously downloading and playing the video sent by the BS. The number of connected users streaming video is constant. The video exists in the server in the following bitrates: {0.2, 0.25, 0.3, 0.4, 0.5, 0.7, 0.9, 1.2, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, and 6.0} Mbps [22]. Users request always 1 second video segments. It is considered that there are not delays between the server and the BS and between the BS and the user.

There are 100 RBs (bandwidth of 20 MHz) available. The RBs efficiencies used to calculate the throughputs depend on the CQI reported by the user every 5 ms, according to Table 2.

The DASH clients request segments using the QAAD, with the parameters $\theta = 0.3s, w = 0.875, \mu = 10s\,, \sigma = 3s$, as suggested in [20]. The initial delay and re-buffering periods are equal to the time that a user needs to download 5 seconds of the lowest video quality. During these periods, the buffers do not leak out 1 ms of video every millisecond spent in the simulation. When the users start the streaming or re-buffering after a stall, the users request 5 segments at the lowest quality level. Then, the requests are made according to QAAD.

The PF algorithm uses $\alpha = \beta = 1$. For PFBF scheduler, it is used $\alpha = \beta = 1$ as suggested in [12]. It is also considered that all videos have a constant bitrate and a frame rate equal to 30 fps and thus the frame size, $S_{frame,i}$, is given by the requested video bitrate divided by 30. It is used $f_{min} = 10$ frames, as suggested in [12], which is approximately 333 ms.

B. MBF's α parameter choice

The term "global QoE" refers to the overall QoE that a user has with all the streaming session. Let's admit that a user is considered satisfied if his global QoE is above a certain limit U and non-satisfied if his global QoE is below a limit Z (with $Z < U$). For a given number of connected users, the percentage of satisfied and non-satisfied users is Y and W %, respectively.

Before studying the schedulers' performances, the parameter α used by MBF must be chosen because it impacts the results presented in section C. To do so, it is plotted the percentage of users with global QoE≥U, for U=3, as a function

of the number of connected users in the network streaming video for α[s] = {0, 2.5, 5, 7.5, 10, 11, 12.5, 15 and Infinity} (see Figure 5). Only these values of α were tested for time simulation reasons. Then, it is also studied the percentage of users with global QoE<2.

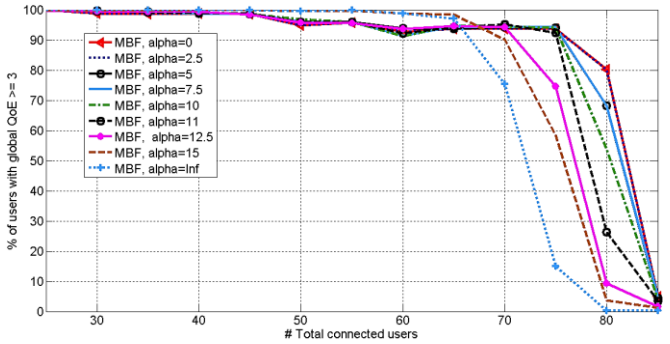

Figure 5 - Percentage of users with QoE≥3 for different α [s].

According to Figure 5, for Y<90 %, the higher the value of α the higher the total number of users that can be connected streaming video. However, it is also important to know how good or bad the remaining users are served. In this context, in Figure 6, it is plotted the percentage of users with global QoE<2 as a function of the total number of connected users.

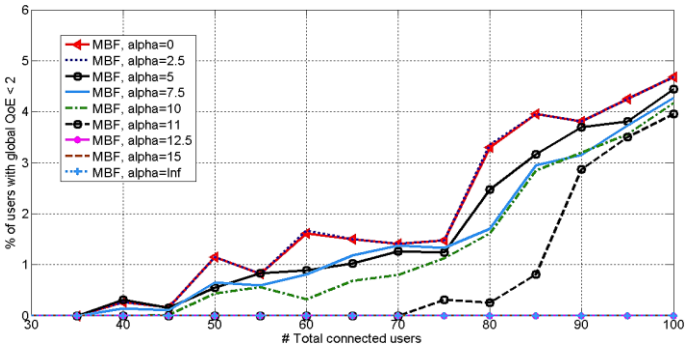

Figure 6 - Percentage of users with QoE<2 for different α [s].

As expected, the α values that showed to lead to a higher total number of connected users have also a higher number of non-satisfied users, with QoE<2. So α is a parameter whose value implies a trade-off between the number of satisfied users with QoE≥U and the non-satisfied users, with QoE<Z. This allows the algorithm to operate in two modes, according to the operator's intention:

- Mode 1: for Y% of users with global QoE≥U, this mode aims at maximizing the number of users with global QoE≥U (first priority), minimizing at the same time the number of users with global QoE<Z. For this mode, an appropriate α value is **5 s** because it leads to a slightly less number of satisfied users for U=3 and U=4 α=0 s and α=2.5 s but it is better at maintaining a low number of non-satisfied users.
- Mode 2**:** this mode seeks to minimize the percentage of users with global QoE<Z (first priority), maximizing at the same time the number of users with global QoE≥U, for a given percentage Y% of users with global QoE≥U. For this mode, it is chosen α=**11 s** because it leads to a low number of non-satisfied users (see Figure 6) and still maintains a high number of satisfied users (see Figure 5).

C. Analysis of the number of satisfied users

In this section, different schedulers are compared regarding the number of satisfied users. Particularly, it is plotted the percentage of satisfied users as the number of connected users increases, for U=3 and U=4. In this analysis, MBF operating in mode 1 is expected to have better results than mode 2 because the first priority of the mode 1 is the number of satisfied users (global QoE≥U) whereas mode 2 is fairer, seeking to have a smaller number of non-satisfied users (with global QoE<Z).

1) For U = 3:

Figure 7 plots the percentage of users with QoE≥3 as the total number of users streaming video increases for the schedulers: RR, BET, PF, PFBF and both modes of MBF. By multiplying the established percentage by the total number of connected users, it is obtained the number of satisfied users.

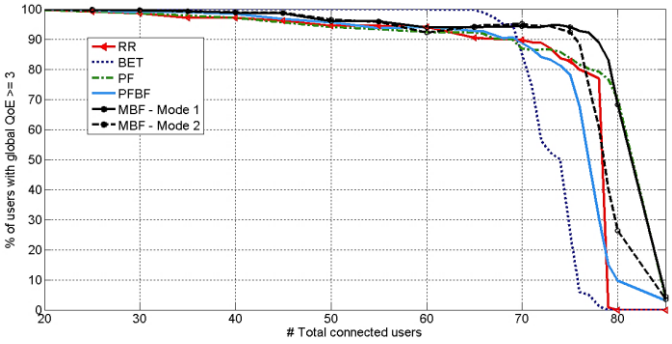

Figure 7 – Percentage of satisfied users for U=3.

Figure 8 presents the number of satisfied users (U=3) for all the algorithms for some percentages of satisfied users that may be interesting (Y=90, 80 and 70%).

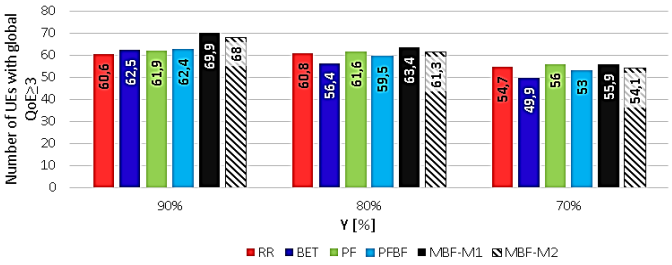

Figure 8 - Number of satisfied users for U=3, Y=90,80 and 70%.

According to Figure 8, in order to satisfy 90% of the connected users (when the criterion of satisfaction is to have a global QoE≥3) the scheduler that supports the largest number of connected users between those that are being tested, is the MBF scheduler operating in mode 1, with the possibility of having on average 69.9 satisfied users, followed by MBF operating in mode 2, supporting on average 68.0 satisfied users. For 80%, the allocation algorithm that leads to the highest capacity is MBF operating in mode 1, supporting on average 63.4 satisfied users. The second one is MBF operating in mode 2 and this scheduler supports on 61.3 satisfied users.

Table 4 - Algorithms capacities and gains with respect to RR.

#Satisfied UEs #Total UES Gain [%]		Algorithm					
		RR	BET	PF	PFBF	MBF Mode 1	MBF Mode 2
Y [%]	90	**60.6** 67.3 0%	**62.5** 69.5 +3.2%	**61.9** 68.7 +2.1%	**62.4** 69.4 +3.0%	**69.9** 77.6 +15.3%	**68.0** 75.6 +12.3%
	80	**60.8** 76.0 0%	**56.4** 70.5 -7.9%	**61.6** 77.0 1.4%	**59.52** 74.40 -2.2%	**63.4** 79.2 +4.23%	**61.3** 76.6 +0.8%
	70	**54.7** 78.1 0%	**49.9** 71.2 -9.6%	**56.0** 80.0 +2.5%	**53.0** 75.8 -3.0%	**55.92** 79.88 +2.31%	**54.1** 77.3 -1.0%

According to Table 4, MBF operating in mode 1 has the highest gains for the Y percentages under test, supporting 15.3% more satisfied users than RR does for Y=90%. It is also the only algorithm with positive gains for Y=90, 80 and 70%. Although MBF operating in mode 2 is not optimized for achieving the highest possible number of satisfied users, it is still better than RR, BET and PF for Y=90%. BET scheduler has the lowest values for approximately Y>95 % but leads to the highest capacities for approximately Y<90 % because it serves the users with the same global QoE. PFBF, whose metric is based on PF's metric, is slightly better than PF for Y>90 %.

2) For U = 4:

Figure 9 presents the percentage of connected users with global QoE≥4 as the total number of connected users increases.

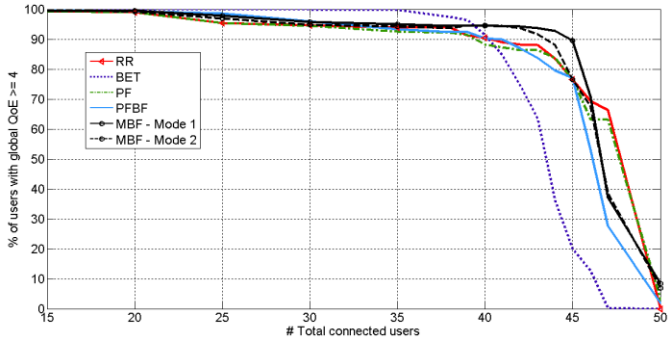

Figure 9 - Percentage of satisfied users for U=4.

Figure 10 presents the number of satisfied users for all the algorithms for Y=90, 80 and 70%.

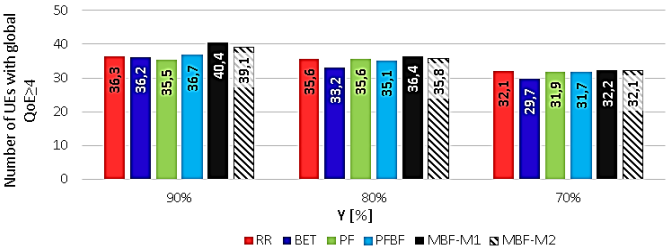

Figure 10 - Number of satisfied users for U=4, Y=90,80 and 70%.

According to Figure 10, in order to satisfy 90 % of the connected users, the scheduler that supports the largest number of connected users between those that are being tested, is the MBF scheduler operating in mode 1, with the possibility of having on average 40.4 satisfied users, followed by MBF operating in mode 2, supporting on average 39.1 satisfied users. For 80%, MBF leads also to the largest capacities. Table 5 presents for each algorithm the number of satisfied users, the total number of connected users and the capacity gains with respect to the RR, for U=4 and Y=90, 80 and 70%.

Table 5 - Algorithms capacities and gains with respect to RR.

#Satisfied UEs / #Total UEs / Gain [%]		Algorithm					
		RR	BET	PF	PFBF	MBF Mode 1	MBF Mode 2
Y [%]	90	36.3 40.3 0%	36.2 40.2 -0.3%	35.5 39.5 -2.2%	36.7 40.8 +1.2%	40.4 44.8 +11.2%	39.1 43.4 +7.8%
	80	35.6 44.5 0%	33.16 41.46 -7.4%	35.6 44.5 +0.1%	35.1 43.9 -1.4%	36.4 45.5 +2.3%	35.8 44.7 +0.4%
	70	32.1 45.9 0%	29.7 42.4 -8.3%	31.85 45.50 -0.9%	31.7 45.3 -1.3%	32.2 46.0 +0.3%	32.1 45.9 0.0%

According to Table 5, MBF operating in mode 1 is the scheduler algorithm with the highest gains with respect to RR and the only one with positive gains for Y=90, 80 and 70%. Although MBF operating in mode 2 is not optimized for achieving the highest possible number of satisfied users, it is still better than RR, BET and PF for Y=90, 80 and 70%, allowing 7.8 % more satisfied users than RR does for Y=90%.

BET has the lowest values for approximately $Y < 85$ % but leads clearly to the highest capacities for approximately $Y > 95\%$ because it is the fairest scheduler by providing the same QoE to all users, regardless their radio channel conditions.

D. Analysis of the number of non-satisfied users

An operator may be also interested in knowing how well or bad the remaining $(100 - Y)$ % users (the non-satisfied ones, with QoE below the threshold Z) are served. In this context, this section analyzes the percentage of users with a global QoE<2 (poor and bad QoE) as a function of the total number of connected users streaming video.

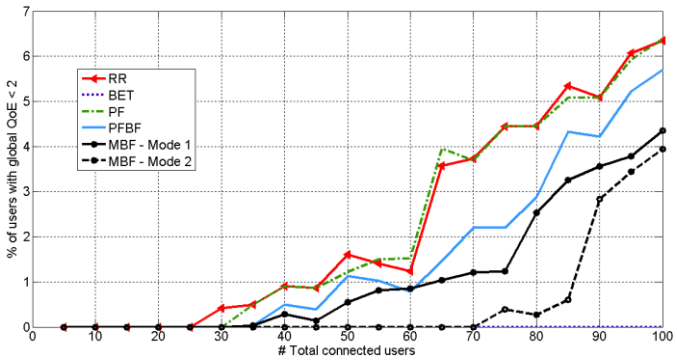

Figure 11 – Percentage of users with global QoE<2.

According to Figure 11, as expected BET is the algorithm that, for a given number of connected users between 5 and 100, leads to the lowest number of users with QoE<2. The second one is the MBF operating in mode 2, the mode that tries indeed to minimize the number of users with a bad or poor quality of experience streaming video MBF operating in mode 2 allows a percentage of non-satisfied users smaller than 1% until there are 85 users streaming video which is a higher number of users than the capacities calculated in section C (Table 4 and Table 5). This positive property of this mode is just possible by having a lower number of users with QoE≥U when compared to the same algorithm operating in mode 1, as it was seen in section C. Still, even with very low percentages of non-satisfied users (less than 3 %), regardless MBF's mode, it supports a higher number of satisfied users than BET for U=3 and U=4 and Y=90, 80 and 70% (see Figure 8 and Figure 10). MBF, regardless of the mode in which it operates, guarantees a lower number of users with QoE<2 than RR, PF and PFBF and supports always more satisfied users for Y=90%.

IX. Conclusions

In today's world, people are getting more and more demanding in terms of the quality of experience they have using a service they pay for. If the service does not provide them the quality of experience they are expecting for, they simply stop paying for it and search for a new service provider. Furthermore, the exponential growth of consumed mobile multimedia content may congest the network. This way, it is of great interest of the operators to provide a service that satisfies the highest possible number of users. To do so, the development

of intelligent and efficient scheduling algorithms may play an important role. A lot of research has been done in this field, however very few schedulers are analyzed regarding the provided network capacity in terms of the number of satisfied and non-satisfied users.

The developed algorithm, Maximum Buffer Filling, uses the buffer level, a QoE metric reported by the users who implement the MPEG-DASH specification. It allocates the resources based on the users' buffer levels and their achievable buffer fillings. The MBF scheduler can operate in two modes, according to an adjustable parameter that tunes the QoE level fairness between the users. One of them has as first priority to maximize the number of satisfied users with a good or excellent QoE. The second one minimizes the number of non-satisfied users with a poor or bad QoE.

MBF regardless the mode in which it operates has shown positive results by providing higher capacities than RR, PF, PFBF and BET, for a 90% of satisfied users. It has a gain between 7.8% and 15.3% with respect to RR, regarding the number of users with a good or excellent QoE. Furthermore, it leads to a lower percentage of users with a bad and poor QoE than RR, PF and PFBF. For a 90 % of users with a good or excellent QoE, it leads to a percentage less than 1% of non-satisfied users. As expected, BET has shown to be better in guaranteeing a minimum number of non-satisfied users.

For future work it would be interesting to study the sensibility of the capacities of the algorithms to the QoE model and to the number of video qualities stored in the server. Furthermore, it would be interesting to analyze the QoE along the streaming session, analyzing the possibility of having users giving up from their streaming sessions. A further development of the metric used by MBF can also be done in order to improve its operating modes capabilities.

X. BIBLIOGRAFIA

[1] "Qualinet, "Qualinet White Paper on Definitions of Quality of Experience", 2013," 2012.

[2] I. Sodogar , "The MPEG-DASH Standard for Multimedia Streaming Over the Internet", IEEE Multimedia, vol.18, nº4 , pp 62-67 , 2011.

[3] ISO/IEC, "Information technology — Dynamic adaptive streaming over HTTP (DASH) —Part 1:Media presentation description and segment formats", 2014.

[4] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 2, pp. 678–700, Second Quarter 2013.

[5] Erik Dahlman, Stefan Parkvall, and Johan Sköld, 4G LTE / LTE Advanced for Mobile Broadband, Academic Press is an imprint of Elsevier, 2011.

[6] 3rd Generation Partnership Project, 3GPP TS 36.213 V14.3.0 (2017-06-23) : Physical Layer Procedures (Release 14).

[7] S. Mumtaz and J. Rodriguez (eds.), Smart Device to Smart Device Communication, DOI:10.1007/978-3-319-04963-2_1,Springer International Publishing Switzerland 2014.

[8] I. Sousa, M. Queluz and A. Rodrigues, "A Survey on QoE-oriented Wireless Resources Scheduling", IEEE Journal.

[9] I. Sousa, M. Queluz and A. Rodrigues, "A Survey on QoE-oriented Wireless Resources Scheduling", IEEE Communications Surveys and Tutorials (submitted).

[10] J. Navarro-Ortiz, P. Ameigeiras, J. M. Lopez-Soler, J. Lorca-Hernando,Q. Perez-Tarrero, and R. Garcia-Perez, "A QoE-Aware Scheduler for HTTP Progressive Video in OFDMA Systems," IEEE Communications Letters, vol. 17, no. 4, pp. 677–680, April 2013.

[11] S. Thakolsri, S. Cokbulan, D. Jurca, Z. Despotovic, and W. Kellerer, "QoE-driven cross-layer optimization in wireless networks addressing system efficiency and utility fairness," in 2011 IEEE GLOBECOM Workshops, December 2011, pp. 12–17.

[12] S. Singh, O. Oyman, A. Papathanassiou, D. Chatterjee, and J. G. Andrews, "Video capacity and QoE enhancements over LTE," in 2012 IEEE International Conference on Communications (ICC 2012), June 2012, pp. 7071–7076.

[13] Quantifying the Influence of Rebuffering Interruptions on the User's Quality of Experience During Mobile Video Watching [J]. IEEE Transactions on Broadcasting, 2013, 59(1):47–61.

[14] "O. Oyman , S. Singh , "Quality of experience for HTTP adaptive streaming services", IEEE Communications Magazine, vol. 50, nº 4, 2012.".

[15] "OMNET. (2012) INET Framework.," [Online]. Available: https://omnetpp.org/. [Acedido em 13 7 2017].

[16] Varga, A. and Hornig, R. (2008), "An overview of the OMNeT++ simulation environment", in Proc. SIMUTools '08, Marseille, France, March 2008.

[17] A. Virdis, G. Stea, and G. Nardini, "SimuLTE - A Modular System-level Simulator for LTE/LTE-A Networks based on OMNeT++," in 4thInternational Conference on Simulation and Modeling Methodologies,Technologies and Applications (SIMULTECH 2014).

[18] A. Virdis e G. Nardini, "SimuLTE LTE User Plane Simulation Model for INET & OMNeT++," SimuLTE, 2015. [Online]. Available: http://simulte.com/. [Acedido em 13 7 2017].

[19] P. K. Rekhi, M. Luthra, S. Malik and R. Atri, White Paper, "Trhoughput Calculation for LTE TDD and FDD Systems", December 2012.

[20] Dongeun Suh, Insun Jang, and Sangheon Pack,"QoE-enhanced Adaptation Algorithm over DASH for Multimedia Streaming", Proceeding of the International Conference on InformationNetworking 2014 (ICOIN2014), pp. 497 - 501, 2014.

[21] Stefano Petrangeli, Jeroen Famaey, Maxim Claeys, Steven Latré, Filip De Turck: QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming. TOMCCAP 12(2): 28:1-28:24 (2016).

[22] WordPress, "ITEC – Dynamic Adaptive Streaming over HTTP," [Online]. Available: http://www-itec.uni-klu.ac.at/dash/?page_id=6. [Acedido em 24 7 2017].