



HAL
open science

Adaptive video streaming solution based on multi-access edge computing advantages

Yassine Douga, Yassine Hadjadj-Aoul, Malika Bourenane, Abdelhamid Mellouk

► To cite this version:

Yassine Douga, Yassine Hadjadj-Aoul, Malika Bourenane, Abdelhamid Mellouk. Adaptive video streaming solution based on multi-access edge computing advantages. *Multimedia Tools and Applications*, 2023, 10.1007/s11042-023-17764-x . hal-04367845

HAL Id: hal-04367845

<https://inria.hal.science/hal-04367845>

Submitted on 1 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Adaptive video streaming solution based on multi-access edge computing advantages

Yassine Douga*

LRDSI Laboratory, University Saad Dahle Blida, Algeria

yassine.douga@hotmail.com

Yassine Hadjadj-Aoul

IRISA Laboratory, University Rennes1, France

yhadjadj@irisa.fr

Malika Bourenane

LRIIR Laboratory, University Ahmed Ben Bella Oran, Algeria

mb_regina@yahoo.fr

Abdelhamid Mellouk

LISSI Laboratory, University Paris Est

mellouk@u-pec.fr

Abstract—In the last few years, video streaming traffic surpassed all the other Internet traffic. This is mainly due to the substantial and ever-increasing size of video files and the widespread popularity of streaming services. A significant portion of the traffic above originates from mobile devices, thereby increasing demands on operators’ infrastructures and potentially degrading services. The content providers’ response has been to adopt adaptive video streaming techniques that avoid playback interruptions, as these interruptions are the leading cause of the deterioration in the Quality Of Experience (QoE). To go beyond end-to-end approaches, the authors proposed a new strategy that utilizes the Multi-access Edge Computing (MEC) standard to optimize streaming services. Being located at the MEC level allows, indeed, to be aware of the network congestion’s state, which facilitates optimizing network operations. The suggested seamless strategy considers the users’ devices’ parameters to prevent them from selecting video quality options that would not enhance their viewing experience. By avoiding choices that could diminish the quality or worsen the experience, our strategy optimizes network resources and reduces energy and bandwidth consumption on the mobile device side. The proposed approach was implemented and tested within an emulation environment. The findings demonstrate the suitability of this type of approach in comparison with existing strategies (YouTube service) in terms of users’ satisfaction and network performance.

Index Terms—Dynamic Adaptive Video Streaming, Multi-access Edge Computing (MEC), QoE, Optimization, Congestion, User Terminal.

1. INTRODUCTION

With the widespread availability of affordable high-speed Internet access, Internet traffic has grown at an unprecedented rate, mainly driven by video traffic [1]. This evolution has led to the emergence of new stakeholders in the field of video content distribution, including video-sharing platforms such as YouTube or video distribution platforms such as Netflix and Amazon Open Connect appliances. Given the nature of the Internet as a “best effort” network, streaming frequently carries the risk of playback interruptions for end users, resulting in a decline in the Quality of Experience (QoE). These factors contributed to the rapid adoption of adaptive Hypertext Transfer Protocol (HTTP) video streaming since it can accommodate variations in network conditions by lowering the received video quality to avoid interruption, which is the most critical element of the user experience [4]. Moreover, over the last few years, there has been a substantial rise in the utilization of mobile devices to access Internet services [8]. Among these services, video streaming has appeared as one of the most prevalent and captivating fields on the Internet. Many Internet service providers, including popular platforms like YouTube, Netflix, and Amazon Prime, actively pursue an expanded client base. To achieve this objective, each provider recognizes the need to offer their customers enhanced services and superior QoE. This prevailing trend has led to considerable research efforts and the emergence of various proposals in this area, such as Dynamic Adaptive Streaming over HTTP (DASH) [7], which is considered one of the most used techniques.

The QoE is defined here as the measure of the overall acceptability of an application or a service, which is perceived subjectively by the end user. Users’ profiles are important metrics influencing the QoE [1]. Recently, many improvements to DASH have been proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 17]. Most of these improvements are classified into two main classes. The first class represents the improvements concerning the users’ side. The improvement methods are directly injected into the users’ devices in this class. Users must use or install additional plugins and applications on their terminal devices to apply the improvement. Out of the fact that installing additional plugins/applications can disturb users by consuming the terminal’s resources, such as the battery, Central Processing Unit (CPU), Random Access Memory (RAM), and memory storage; one must be compatible with all users’ operating systems.

Furthermore, most of these improvements need to consider the Quality Of Service (QoS) parameters. The main advantage of such modification is that it does not require much effort or a big budget. After realizing an improvement, the service provider needs to release a new version of the plugin or an application update.

The second class covers server-based improvements. All the upgrades occur in this class from the user's terminal devices. Implementing such improvements by the service providers requires a dedicated effort and financial resources. The other advantage of this improvement is that the users do not need to install or add any additional plugins or applications since we have to notice the progress of their QoE. In addition, this optimizes their terminal's resources. Multi-access Edge Computing (MEC) presents a promising opportunity for leveraging networking infrastructures (see Figure 1) more effectively. By offering cloud computing capabilities and Information Technology (IT) services within the Radio Access Networks (RAN) [6], MEC enables Internet service providers to target a larger client base [3]. Providers must enhance their services and favor QoE for their customers to achieve this goal. This situation has sparked extensive research and numerous proposals, with one of the most notable techniques being DASH [7].

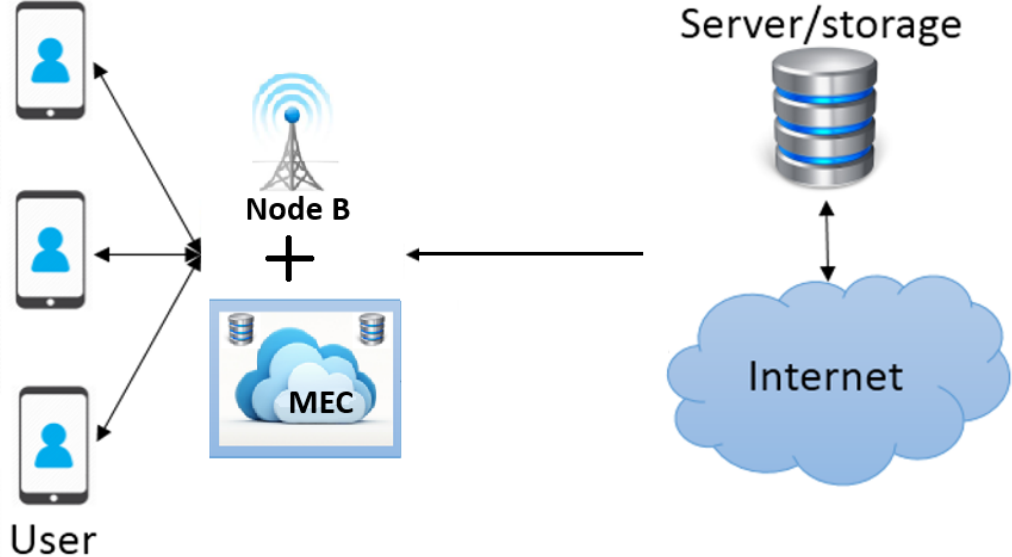


Fig.1 MEC environment

The MEC aims to create an agent near the mobile devices that improves the network QoS, such as good latency and ensures highly efficient network operations and service delivery while enhancing the users' experience [10]. The term “Edge” was added to refer to the radio base station and servers within the radio network (e.g., Node B, Radio Network Controller (RNC), etc.). In this new environment, the MEC agent can get the network, users, and radio parameters and information in real-time, which are not accessible to the users or the server. The MEC agent enables access to information that was previously unavailable in a traditional environment without MEC, thanks to the implementation of cross-layer solutions. In addition, the MEC agent can do some operations and treatments based on the information and parameters stored on the cloud to improve the users' experience and communication QoS since mobile users sometimes have limited computing capabilities and battery storage [14].

In this paper, the authors focus on a particular adaptive streaming approach that uses users' terminal parameters such as screen size, screen resolution, battery availability, and user feedback to improve their experience. This approach was based on DASH and adapted to a MEC environment. Similarly, this adaptation provides a better QoE and service by optimizing the resources of the user terminals by using the adaptation process in the MEC station instead of the user terminals. The paper proposes enhancing our previous idea, described in [1], by adding a congestion control function that allows improved QoS and QoE results.

The rest of the paper was organized as follows: Section 2 describes related works. Section 3 focuses on things that can be improved in [1] and the problem formulation. Section 4 describes the proposed adaptation and its advantages. The emulation environment scenarios and results are discussed in section 5. Finally, we conclude the paper with a conclusion.

2. RELATED WORKS

Video streaming services over HTTP have become widespread, with various approaches proposed to enhance the QoS and improve the overall QoE. In main objective of this section is to address the issue of QoE improvement in video streaming. A comparative analysis of these solutions is performed in a way to provide insight into their effectiveness and limitations. Examining these related works aims to contribute to the advancement of video streaming technology by suggesting a new solution to the identified problems.

Among the proposed solutions to improve video streaming quality, adaptive video streaming over HTTP stands out as a popular and widely adopted method [18]. The core idea behind adaptive video streaming was to store a video sequence with different quality levels. When a user requests a video, the system dynamically adjusts the streaming quality based on the user's network parameters. This approach mitigates video streaming issues and provides improved QoS and QoE outcomes.

In [1], the authors introduced a novel solution to enhance the users' QoE in adaptive video streaming over HTTP. They proposed integrating the user's terminals' parameters and their feedback as additional criteria in the adaptation process. A vital component of this approach was the creation of a virtual interface that manages all incoming video streaming flows with maximum bandwidth capacity calculated based on newly added metrics. The considered user parameters encompass the leading terminal's displaying resolution, screen size, and battery availability. These parameters are weighted using factors such as the Resolution Factor (FR), the Battery Factor (FB), and the Screen Size Factor (FS) and contribute to the calculation of a global User Factor (UF) that represents the user's terminal. Each UF value corresponds to a specific received video quality, thereby determining the allocated bandwidth for the virtual interface (see Tables 5 and 6 for more details).

To evaluate the user's satisfaction, the Mean Opinion Score (MOS) was employed in [1] as a rating system ranging from one to five, capturing the overall sentiment towards the service. In this approach, named TCP Based-User Control For Adaptive video streaming (TBUC), this value was used to update table values, which attribute for each UF value a video quality using a reinforcement learning solution with Markov Decision Process (MDP) [16], as shown in Figure 2. TBUC was tested in a LINUX environment with various scenarios, showing superior QoE results compared to a conventional adaptive video streaming service.

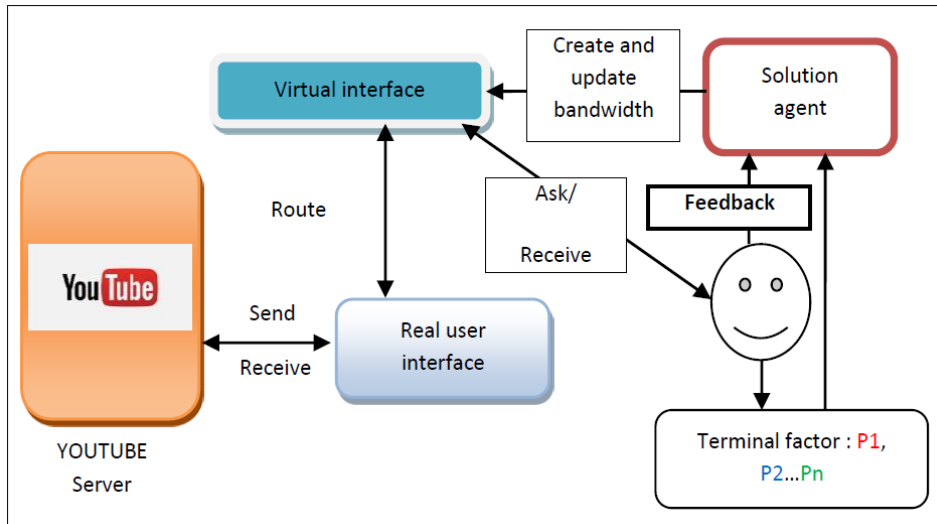


Fig.2 TBUC approach [1]

In [17], the authors proposed an adaptive streaming solution that leverages reinforcement learning in a MEC environment. This user-centric solution aims to enhance the users' QoE by employing machine learning techniques, specifically the MDP. The MEC agent utilizes information provided by edge computing assistance about the users to generate a multi-client adaptive streaming policy. Each user's measured bandwidth, buffer level, and requested bitrate in the proposed system serve as states. These states are input to a neural network, which determines the user's most suitable video bitrate. Additionally, the QoE metric was employed as a reward, which plays a crucial role in the neural network's decision-making process. We noticed the superiority of the proposed solution in terms of QoE against existing solutions from the obtained results.

In [2], the authors propose a novel architecture for adaptive HTTP video streaming services in a MEC context. The basis of this solution was to improve the users' QoE by reducing the probability of network congestion. To achieve this goal, the authors proposed an algorithm that dynamically controls the video representations available at the clients' side according to the current network status, thereby improving the clients' video adaptation mechanism. In a normal situation, when a particular user requests a video to a DASH server, the server sends back a Media Presentation Description (MPD) file, which includes information about the available bitrates, the timing, and the supported format/codec, as well as the Uniform Resources Locator (URL) of the video Scheme along with Edge Computing Assistance and audio segments. In case of congestion, the proposed algorithm automatically intercepts all MPD files at the MEC level and decides which video representations should be removed from the MPD file to relieve network congestion [2]. A video representation (i.e., a video quality) is suppressed by deleting the URI pointing to it in the MPD file. The proposed solution was tested through emulation. The study's findings indicate that the suggested solution gives a better MOS when overloading the network.

Table 1. Comparison between different solutions Studied

Solution	Based side	Improvement criteria	Complexity	Architecture
Solution [1]	User	QoE	$O(S)$	Not MEC
Solution [2]	Network	QoS	$O(S)$	MEC
Solution [17]	User	QoE	-	MEC
Proposed solution	User/Network	QoE	$O(S)$	MEC

Where S is a set of available video qualities on the MPD file.

Regarding the complexity of the solution [17], the proposed approach used a neural network model, which is generally a complex solution which consumes time (for training).

Table 1 compares the various states-of-the-art strategies, and situates the proposed approach more clearly in relation to the other approaches, as it enables actions to be taken not only at the user level, but also at the network level. Moreover, our approach was QoE-oriented and based on a MEC architecture.

3. TCP BASED-USER CONTROL FOR ADAPTIVE VIDEO STREAMING OVER MEC ENVIRONMENT

3.1 Problem context

One of the benefits of 5G and MEC architecture is its potential to enhance Internet services, especially Over The Top (OTT) video streaming services. We assign a default feedback value of 4. As video streaming traffic occupies the most important part of global traffic, improving the performances of such services highly impact the global Internet and its users [15]. However, most of the existing solutions for video streaming improvement focus on, either, the user side or the server side, but not both simultaneously.

The purpose of the proposed solution was to suggest an approach which considers the user side and the server side at the same time in order to achieve better results.

3.2 Proposed approach

The idea behind the proposed TBUC approach was to centralize all operations, treatment and data that were previously collected and gathered by the TBUC solution on one point - MEC agent. This relieves the users' devices from executing additional operations like calculating the UF of the device, adapting the maximum quality and creating a virtual interface. This improves devices' resource consumption and network latency. Moreover, in TBUC, the data gathered from all users is employed to enhance the network QoS, directly improving the users' QoE.

TBUC can be divided in three main steps. The first step corresponds to the adaptation of the received video quality based on the user's terminal parameters. The second step is responsible for improving the global QoS of the network using data gathered in the first step. The third step makes use of a Machine Learning (ML) algorithm based on the user's MOS to adapt and improve the obtained results.

In the first step of the proposal, each time a user makes a video request from an adaptive video streaming server, the MEC agent intercepts it (HTTP request), extracts the user device parameters and calculates the UF of the device, then estimates the video quality that corresponds to the UF. The calculated UF and

video quality of each device are stored into a database with devices' multimedia access control address (MAC) as the primary key. Once the adaptive video streaming server receives the user video request, it responds by sending the video MPD file to the user, which contains different links to video segments stored in multiple qualities. Before the MPD file reaches the user, the MEC agent intercepts it, removes all video's qualities superior to the estimated maximum, then sends it to the user's video player.

The value of UF was calculated according to the user's device parameters. The considered device parameters are screen size, screen resolution and battery availability.

To calculate the first value of UF authors use the formula 1

$$UF = FR + FB + FS \quad (1)$$

The values of FR, FB and FS are obtained from the tables 2, 3 and 4:

Table 2. Values of FS [1]

Screen Size (SS) in Inch	FS =
$55 \leq SS$	7
$42 \leq SS < 55$	6
$32 \leq SS < 42$	5
$15 \leq SS < 32$	4
$15 \leq SS < 7$	3
$SS \leq 7$	2

Table 3. Values of FR [1]

Screen Resolution (TR) in Pixel	FR =
$2160 \leq TR$	7
$1440 \leq TR < 2160$	6
$1080 \leq TR < 1440$	5
$720 \leq TR < 1080$	4
$480 \leq TR < 720$	3
$240 \leq TR < 480$	2

Table 4. Value of FB [1]

Available battery in %	FB =
$40 \leq BC$	6
$20 \leq BC < 40$	4
$0 \leq BC < 20$	2

To obtain the video quality that corresponds to UF, one needs to refer to table 5.

Table 5. Values of Video Quality –(VQ)

UF	VQ
8	240p
9	360p
10	360p
11	420p
12	420p
13	420p
14	720p
15	720p
16	1080p
17	1440p
18	1440p
19	2160p
20	2160p

To estimate DASH adaptive video streaming bandwidth consumption according to video quality, one may refer to Table 6

Table 6. The required bandwidth according to video quality

VQ	Required Bandwidth (RB) in Kb/s
240p	57
360p	128
420p	245
720p	430
1080p	720
1440p	1057
2160p	1970

The values of the tables 2, 3, 4, 5, 6 and 7 are obtained from TBUC solution [1].

To obtain the video quality that goes with the UF, one may also refer to table 5 of the TBUC solution.

The organizational chart on the figure 3, explains how the first step works (MEC agent side):

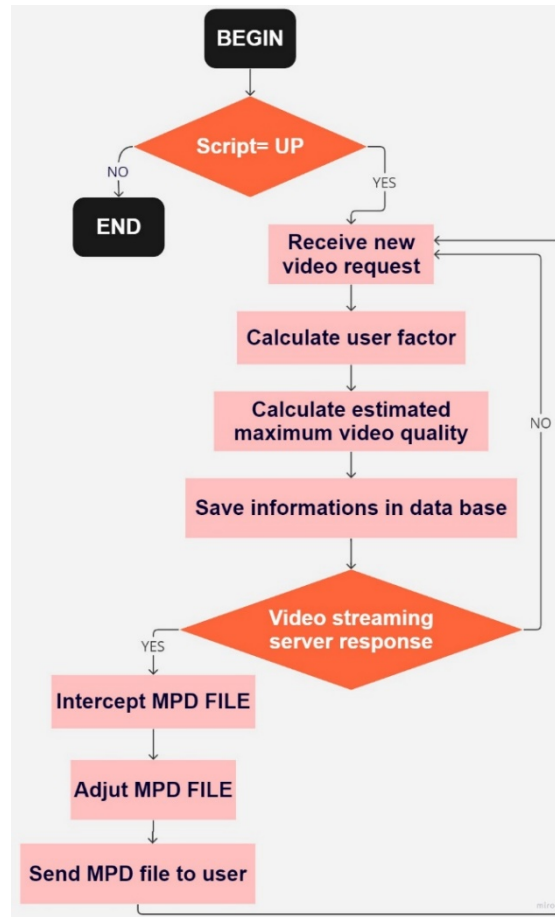


Fig.3 Flow diagram of the first step

In the second step, we adjust the QoS parameters to optimize bandwidth consumption and avoid network congestion. This was achieved by estimating the global required bandwidth value of all active and connected adaptive video streaming users to the same MEC agent.

To obtain the global required adaptive video streaming bandwidth value we summed up all maximum required bandwidth values corresponding to each UF (from the database) of connected and active users.

$$VSRB = \sum_{i=1}^n VSURB(i) \quad (2)$$

With VSRB: adaptive video streaming required bandwidth

n: number of connected and active users

VSURB: Adaptive video streaming user required bandwidth

The obtained adaptive video streaming global required bandwidth value was used as a reference to limit the bandwidth value allocated by the operator to adaptive video streaming services.

The closer the global available bandwidth value of the access point is to the global adaptive video streaming required bandwidth value, the higher the risk of network congestion. To solve this, when the difference between these two bandwidth values reaches a threshold, the MEC agent automatically removes temporarily the highest video quality links available in each MPD file destined for users.

In the last step, we applied a machine learning mechanism that adapts the video quality corresponding to the UFs values according to the average MOS values across all users. This operation was performed periodically.

3.3. Formalism of the proposed models:

a. First model: Adaptation according to the terminal parameters:

The adaptation problem can be formulated as an MDP, which is defined as a tuple (S, A, P, R) , where S and A denote state and action spaces, $P(S_{t+1}|S_t, a)$ represents the transition probability from state $s_t \in S$ to $s_{t+1} \in S$ when taking action $a \in A$, and R denotes the reward by taking action a at state s . [15]

When the MEC agent intercepts the MPD file (in its destination to the user) after a video request, it modifies it according to the values of tables 2, 3, and 4. The MEC agent updates the values of Ufs using a finite MDP. Let's suppose that K users are connected to the edge. Form K connected users, N users are using adaptive video streaming services. Each user was identified by his MAC address, $US = \{MAC_1, MAC_2 \dots MAC_N\}$. For each user from US , a value of UF was calculated and attributed. The Ufs are indexed by $I = 1, 2, 3 \dots L$. We called this set A where $A = \{A_i^l / i = MAC_1, MAC_2 \dots MAC_N \& l = 1, 2, 3 \dots L\}$.

In addition, we have a set of users' feedbacks F indexed by $j = 1, 2, \dots, M$ where $F = \{F_i^j / i = MAC_1, MAC_2 \dots MAC_N \& j = 1, 2, 3 \dots M\}$.

Authors proposed a framework based on a MDP to choose the adequate video qualities. We consider that each user Q has a maximum authorized video quality X , X was index by $S = 1, 2, 3, \dots, X$ and $Q = \{Q_i^s / i = MAC_1, MAC_2 \dots MAC_N \& S = 1, 2, 3 \dots X\}$.

- **MDP formulation:**

System states: The system state at time t was defined as $S_t(UF_t^l, VQ_t^l)$, where the vector represents the allowed VQ_t^l that go with a specific UF at time t . The allowed set of VQ_t^l changes according to users' feedbacks and network condition, so the system state set $S = \{S_1, S_2, S_3, \dots, S_t\}$ keeps changing in time.

System actions: A sequence of action was defined by $\{ac_t\}$ where $t = \{1, 2, 3, \dots, *\}$. The action ac_t is a decision made at time t . The action set of a specific state was given by $AC(s) = \{AC_{UFVQ}, AC_{up}\}$ where AC_{UFVQ} was the action that deletes all video qualities that are superior to the video quality that goes with the UF according to the table 5. The AC_{up} was the action that updates the values of Ufs according to the average value of the users' feedbacks.

State transition: The transition between S_t and S_{t+1} was determined by the available video quality of the video and the value of UF at time t .

The transition probability was obtained by this formula:

$$\begin{aligned}
 P_{Act}(S_t, S_{t+1}) &= Pr\{S_{t+1} | S_t, AC_t\} \\
 &= Pr\{(UF_t^l, VQ_{t+1}^l) | (R_t, VQ_t^l), AC_t\} \\
 &= Pr\{(UF_{t+1}^l | UF_t^l, UF_{t+1}^l = AC_t\} Pr\{VQ_{t+1}^l, VQ_t^l\} \quad (3)
 \end{aligned}$$

$Pr\{VQ_{t+1}^l, VQ_t^l\}$ was obtained as follows:

Knowing the set of allowed video quality that goes with a specific UF in time t , we can estimate the probability to transit to the set of allowed video quality of a specific UF at time $t+1$ using the transition matrix of the Markov Model: $Pr = \{R_{t+1} | R_t, R_{t+1} = AC_t\}$, the decision was made according to the action AC_t .

Reward function: The reward is the obtained payoff after executing an action AC_t to transit from a state S_t to S_{t+1} . In our case, we want to get a QoE reward function to consider the users' satisfaction. The reward value was obtained according to the average of obtained MOS of users. Each value of MOS was obtained while executing action AC_t to transit from state S_t to S_{t+1} .

$$R_{t+1} = f(AVG(R_{AC_t}(S_t \rightarrow S_{t+1}))) \quad (4)$$

Where $AVG(R_{AC_t}(S_t \rightarrow S_{t+1}))$ is the average value of obtained MOS values while executing action AC_t to transit from S_t to S_{t+1} .

The f function takes its values from the next table (Table 7):

Table 7. The relationship between average value of MOS and reward value

$AVG(R_{AC_t}(S_t \rightarrow S_{t+1}))$	$f(AVG(R_{AC_t}(S_t \rightarrow S_{t+1}))) = \text{Reward}_{t+1}$
$1 \leq \text{and} > 2$	2
$2 \leq \text{and} > 3$	1
$3 \leq \text{and} > 4$	0.5
$4 \leq \text{and} \geq 5$	0

The obtained reward value will be added to the value of UF of each connected user. The idea behind adding the reward value to the UF was to improve the maximum authorized video quality when most of the users are not satisfied or to keep the system state as it is when the users are satisfied.

The MOS, was obtained directly from the end users. The authors asked a group of about 15 participants to watch different videos with varying quality levels and rate their subjective quality perception on a scale from 1 (bad) to 5 (excellent), following the ITU-T P.800 recommendations. The MOS for each video was then computed as the arithmetic mean of the individual ratings.

It is important to mention that the MOS value for each user was initialized at 4, so if a user does not provide a MOS value, we consider they were satisfied.

Algorithm 1: Adding/updating user's information

Input: M, a, TR, BC

Output: M, UF, VQ, MBW

```

1  If ( video_request == 1 ) then // user request a video from the server
2       $M \leftarrow \text{Get\_source\_mac}(video\_request)$  // get user's mac address
3       $a \leftarrow \text{Select mac from database, where mac} = M$ 
4      If (  $a == \text{NIL}$  ) then //if new user
5           $TR \leftarrow \text{Get\_TR}(video\_request)$  // get user's terminal resolution
6           $BC \leftarrow \text{Get\_BC}(video\_request)$  // get user's available battery
7           $UF \leftarrow \text{user\_factor}(TR, BC, SS)$  //calculating the user factor
8           $VQ \leftarrow \text{choose\_quality}(UF)$  // Select the video quality that goes with the user factor
9           $MBW \leftarrow \text{choose\_bandwidth}(VQ)$  // Select the required bandwidth MBW that goes with the
           maximum video quality of the user
10          $\text{Save\_database}(M, UF, VQ, MBW)$ 
11     end if
12 end if

```

Algorithm 2: Adaptation according to user's information

Input: MPD, ms
Output: MPD, ms

- 1 **If** (video_answer == 1) **then** // Intercept MPD file
- 2 ms ← **Get_destination_mac**(video_answer) //get destination user's mac address
- 3 vq ← **Select** vq from database where mac=ms // get the maximum video quality of the user
- 4 mpd ← **Get_mpd**(video_answer)
- 5 **Update** (mpd,vq) // delete all MPD video segments links with video quality superior to vq
- 6 **Send** (mpd, ms)
- 7 **end if**

Algorithm 3: Updating the value of UF according to user's feedbacks

Input: fb, r, mpd
Output: UF, r

- 1 **While** (service == 1) **do** // while our system is on
- 2 fb ← **Select avg**(feedback) from database; // calculate average feedback
- 3 r ← **Get_reward**(fb) // get the reward value that goes with the average feedback value
- 4 mpd ← **Get_mpd**(video_answer)
- 5 **Update** (UF,r) // add the value of r to all values of UF on the database
- 6 **Sleep** (t)
- 7 **end while**

b. Second model: adaptation according to network conditions

A second finite MDP was used to formulate this adaptation process. It uses the same resources as the first process.

Periodically, the MEC agent checks the sum of the required bandwidths according to the maximum video quality of all connected and active users. The difference between the sum of the bandwidths and the bandwidth allocated to the adaptive video streaming service will be used to estimate the probability of network congestion.

To estimate the probability of congestion we use the following formula:

$$P_{C_t} = \frac{\sum_{i=1}^n BWQ(i)}{BWVS_t} \quad (5)$$

Where P_c is the probability of congestion, 'n' the number of connected and active users, $BWQ(i)$ the required bandwidth to display the maximum authorized quality of a user (i) and $BWVS$ is the adaptive video streaming service allocated bandwidth.

Since the value of P_c is under a threshold value, the adaptation system remains inactive. When the P_c reaches the threshold, the adaptation system reacts by temporarily removing the best video quality on each MPD file in its destination of users and modifying the values of UFs.

We also proposed a framework based on an MDP to prevent network congestion.

- MDP formulation:

System states: The system state at time t was defined as $SS_t(UF_t^1, Pr_t)$, where the vector represents the probability of network congestion Pr_t along with the value of UF of each user at time t. The network congestion probability changes according to the value of active UFs, so the system state set $SS = \{SS_1, SS_2, SS_3, \dots, SS_t\}$ keeps changing in time.

System actions: A sequence of action was defined by $\{a_t\}$ where $t = \{1, 2, 3, \dots, *\}$. The action a_t was a decision made at time t . The action set of a specific state was given by $AT(s) = \{AT_{cg}, AT_{uf}\}$ where AT_{cg} was the action that calculates the congestion probability, compares it to the threshold values and deletes all video qualities that are superior to the video quality that goes with the threshold value. The AT_{uf} was the action that updates the values of UFs according to the values of network congestion probability.

State transition: The transition between SS_t and SS_{t+1} was determined by the value of network congestion probability at time t .

Reward function (G): The reward was the obtained payoff of executing an action AT_t to transit from a state SS_t to SS_{t+1} . In our case, we want to get a reward function that modifies the values of UFs according to network congestion probability to prevent network congestion in the future.

$$r_{t+1} = G(Pr_t) \quad (6)$$

We suppose that Y is the value of the network probability congestion threshold.

Table 8. Relation between network congestion probability and the reward value

$(Pr_t - Y)$	Reward r_{t+1}
≤ 0.05	0
> 0.05 and ≤ 0.1	-0.5
> 0.1 and ≤ 0.2	-1
> 0.2	-2

After each transition between S_t and S_{t+1} a reward value was calculated and applied according to table 8. The reward value will be added to the values of UFs of connected and active users.

Algorithm 4: Updating the value of UFs according to congestion probability

```

Input: bw, bwvs, y, x, z
Output: UF
1  While (service==1) do // while the system is on
2    bw ← Select_Sum (MBW) from database; //calculate the global bandwidth consumption of all users
3    pr ← bw/bwvs; //calculate the network congestion probability
4    a ← pr-y; //y is the congestion probability threshold
5    While (a>=0) do
6      If (a > x and a=<y) then
7        Update database set UF ← UF-0.5 // Sum the value of 'a' to all database user factors
8      Else If (a > y and a=<z) then
9        Update database set UF ← UF-1 // Sum the value of 'a' to all database user factors
10     Else If a > z then
11       Update database set UF ← UF-2 // Sum the value of 'a' to all database user factors
12     end if
13   end if
14 end if
15 end while
16 end while

```

Where the values of x , y and z are obtained from table 8.

In adaptive video streaming over HTTP, the server sends and updates the MPD files periodically. In Netflix, for example, this corresponds to 30 seconds [19]. Thus, in case of poor network condition, the

proposed solution removes the highest video qualities from the newly received MPD file according to the user's parameters and network condition. For subsequent MPD file segments, if the network conditions remain the same, the solution removes the same video qualities from the second part of the MPD file, but if the network condition gets better, it only removes the video qualities according to terminal parameters.

4. EMULATION AND RESULTS

4.1 Emulation environment

The proposed solution has been tested and validated on a Linux environment. We developed four different scripts.

The script execution is independent of user and server operations. We tested the solution across three device types - smartphones, laptops, and smart TVs - whose parameters are defined in Table 9. YouTube has been chosen as the adaptive video streaming service because its popularity and the fact that it is based HTTP [3]. Another adaptive video streaming services-based HTTP could have been chosen.

Table 9. Emulation according to terminal parameters

Device	Screen size (inches)	Resolution	Battery life (hour)	Max number of users	Default UF
Smartphone	6.7	1440p	22	5	14
Laptop	15.5	1080p	2.5	5	16
Smart TV	55	2160p	-	5	20

At the beginning of each emulation scenario, all terminals (working under a battery) were fully charged.

4.2 Emulation topology

To emulate a MEC architecture, we used a computer equipped with a wireless network interface. This computer uses a wired link to access to the Internet. The computer Wireless interface gives access to the Internet and YouTube server while sharing at the same time the wired Internet connection. The computer represents the MEC mobile agent. Users with different kinds of terminals were connected to the wireless interface instead of 5G Node B in real life. The scripts were executed on the computer (MEC mobile agent). The maximum speed of the wireless interface was 500mb/s and the Internet speed that links the computer (MEC agent) to the Internet is 100mb/s.

The Figure 4 illustrates emulation our topology.

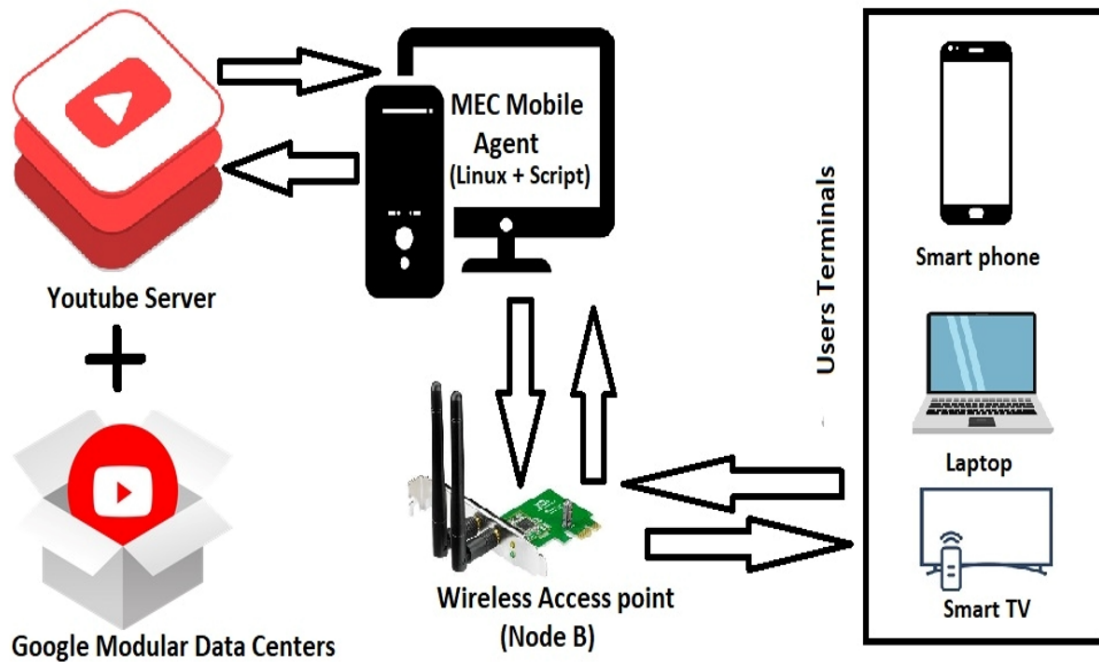


Fig.4 The Emulation topology

During all emulation scenarios, the value of congestion probability threshold ‘Y’ was set at 0.75 and the value of adaptive video streaming maximum reserved bandwidth ‘BWVS’ at 9000kb/s (using QoS parameters tuning).

4.3 Emulation scenarios

We began the emulation with 3 users, each using a randomly assigned device type. Every 30 minutes, additional users were added with randomly assigned devices. After each 30-minute period, all active users rated their video streaming experience. The system checks the congestion probability and react to it (avoid network congestion). It also takes in consideration the user’s feeling average mark (MOS) before applying it.

All users watch the same video from YouTube encoded in (240p, 360p, 480p, 720p, 1080p, 1440p, 2160p).

The same emulation was done without the proposed solution with YouTube service.

The table 10 shows the obtained results using our solution.

The table 11 shows the obtained results without using our solution.

The figure 5 shows the evolution of the sum of UFs according to the number of users.

The figure 6 shows the evolution of congestion probability according to the number of users.

The figure 7 shows the evolution of bandwidth consumption evolution according to time but without using our solution.

The figure 8 shows the evolution of the average value of MOS according to the number of users with and without using our solution.

4.4 Emulation results and discussion

Table 10. Emulation results when applying the proposed solution

Time	Number of users	UFs Sum	Congestion probability	Bandwidth consumption	Average value of MOS
0h	3	56	0,453	3171	4
30min	5	86	0,755	5285	5
1h	7	116	0,72	5040	4
1h30min	9	148	0,92	6480	5
2h	9	148-18	0,55	3870	3
2h30min	12	172+6	0,73	5160	2
3h	15	218+15	0,92	6450	3

The obtained results on table 10 (while using our solution) show how the bandwidth consumption and congestion probability evolve according to the number of users and how far this evolution impacts the average value of MOS. We can notice that the number of users was not the only factor that impacts the network parameters. In time (30min) we had five users that consumed 5285kb/s and in time (1h) we had seven users that consumed 5040 kb/s. The parameters of the user terminal are a factor that determine bandwidth consumption in addition to the number of users. We can also mention that with the same number and type of users we had 6480 kb/s as bandwidth consumption in time (1h30) and 3870kb/s in time (2h), this was due to the congestion control mechanism that reduces the maximum received video quality when congestion probability reaches the threshold value.

Table 11. Emulation results without applying the proposed solution

Time	Number of users	Congestion probability	Bandwidth consumption	Average value of MOS
0h	3	0,53	3747	4
30min	5	0,755	5285	4
1h	7	0,96	6774	3
1h30min	9	Congestion	8551	1
2h	9	Congestion	8551	1
2h30min	12	Congestion	12298	1
3h	15	Congestion	16045	1

The table 11 represents the obtained results of the emulation without using our solution. We can notice that the bandwidth consumption and the congestion probability have a direct relationship with the number of users and the type of device. Bandwidth consumption was more important with our solution because there was no mechanism that optimizes it according to the device parameters. When the congestion probability reaches a threshold value, no system exists that can help to avoid network congestion. Consequently, the congestion occurs faster compared to the one in our proposed solution.

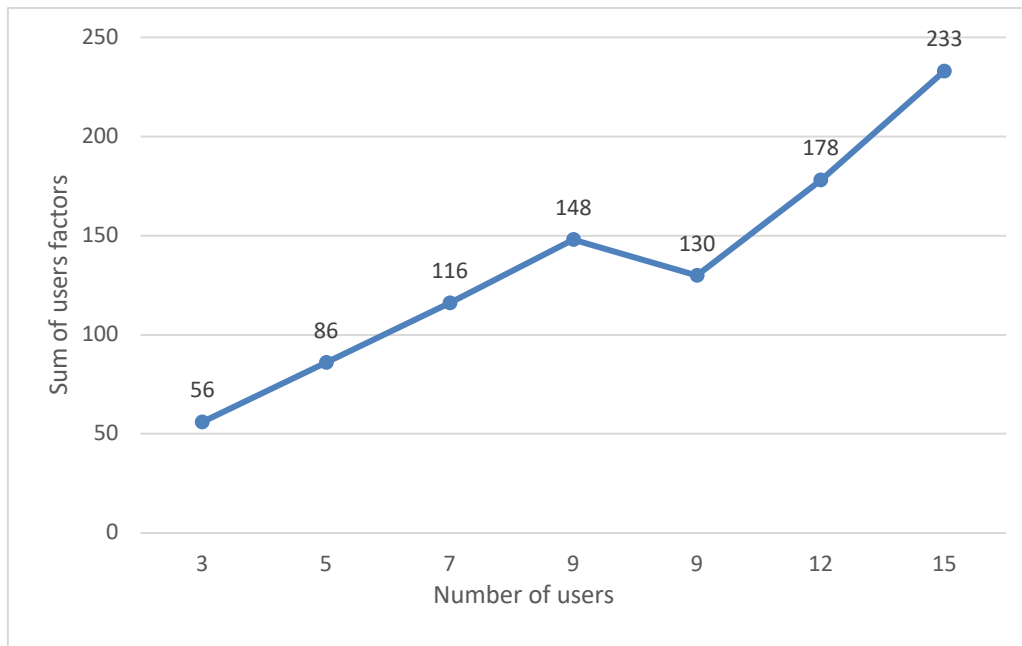


Fig.5 Sum of UFs evolution according to the number of users

The figure 5 shows how the sum of UFs changes according to the number of users. We notice that the evolution was not linear, due to the different type of terminals and their parameters. This shows that the value of users' feedback sum cannot be directly estimated according to the number of users without considering the type of devices used. It is important to remind that this value will be used **later** to estimate the network's congestion probability.

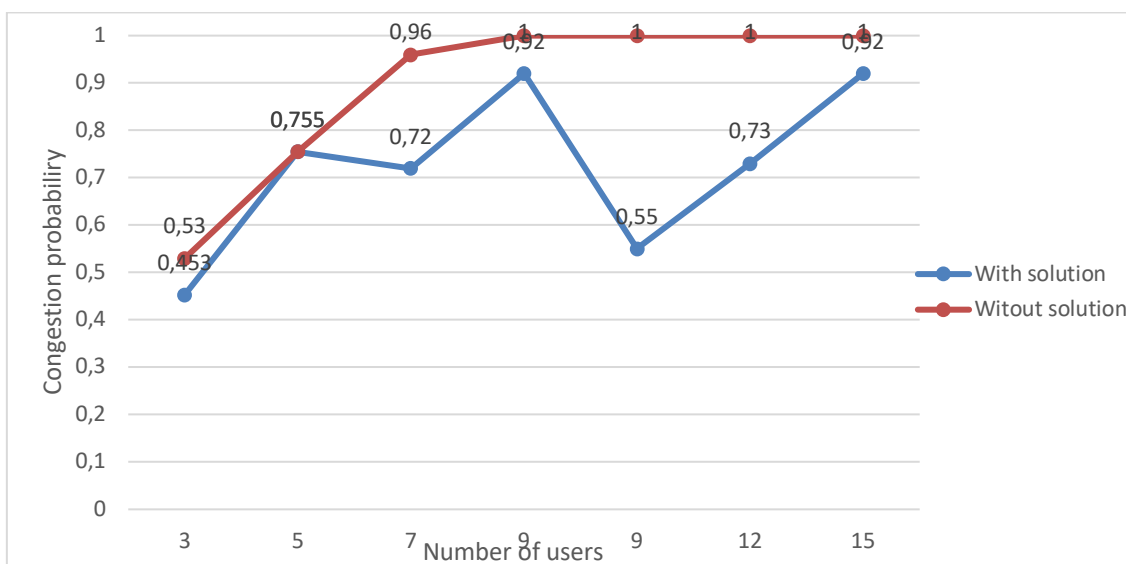


Fig.6 Congestion probability evolution according to the number of users with and without applying the solution

The figure 6 shows the evolution of network congestion probability with and without applying our solution. We can observe that our solution avoids early network congestion and keeps the network working for a longer time using the optimization mechanism (with users' devices parameters) and the congestion avoidance mechanism through the removal of the highest video qualities when the

congestion value reaches the threshold. However, without the use of the proposed solution, once congestion occurs, negative impacts on the video streaming service and the users' satisfaction are observed because of the absence of a mechanism that may reduce those negative impacts.

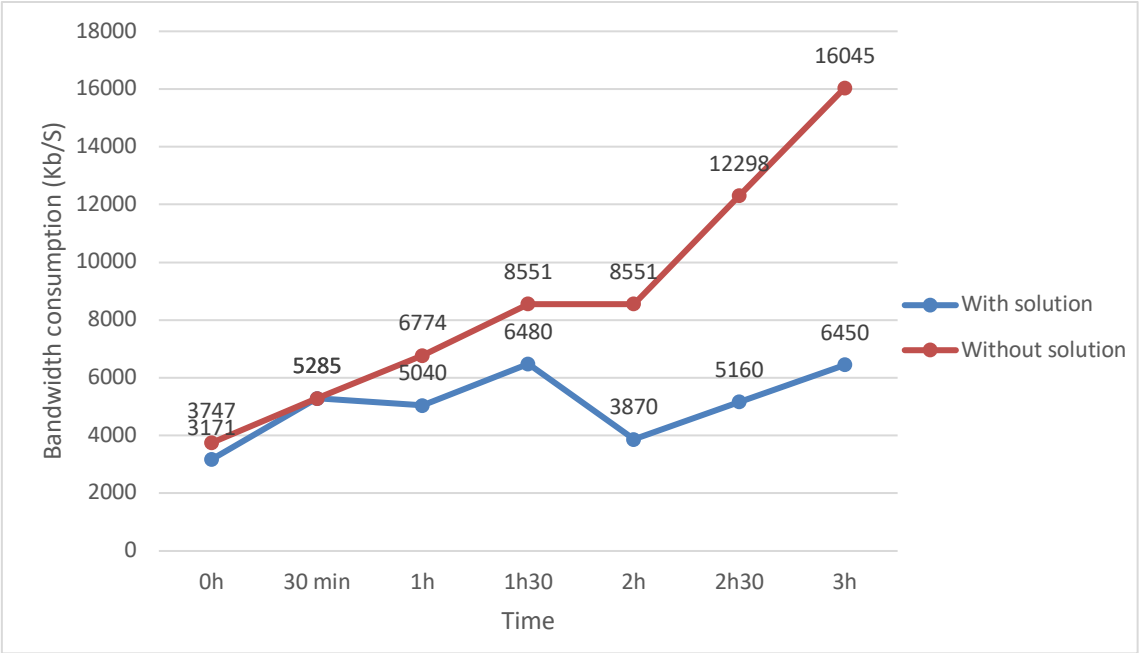


Fig.7 Adaptive video streaming bandwidth consumption evolution with and without applying the proposed solution

The comparison between the evolution of the bandwidth consumption with and without applying the proposed solution was described in figure 7. Our solution offers less bandwidth consumption, which gives users better data optimization. This bandwidth optimization is the results of adapting the maximum received video quality with the terminal parameters and taking into consideration the user's feeling while choosing the best video quality.

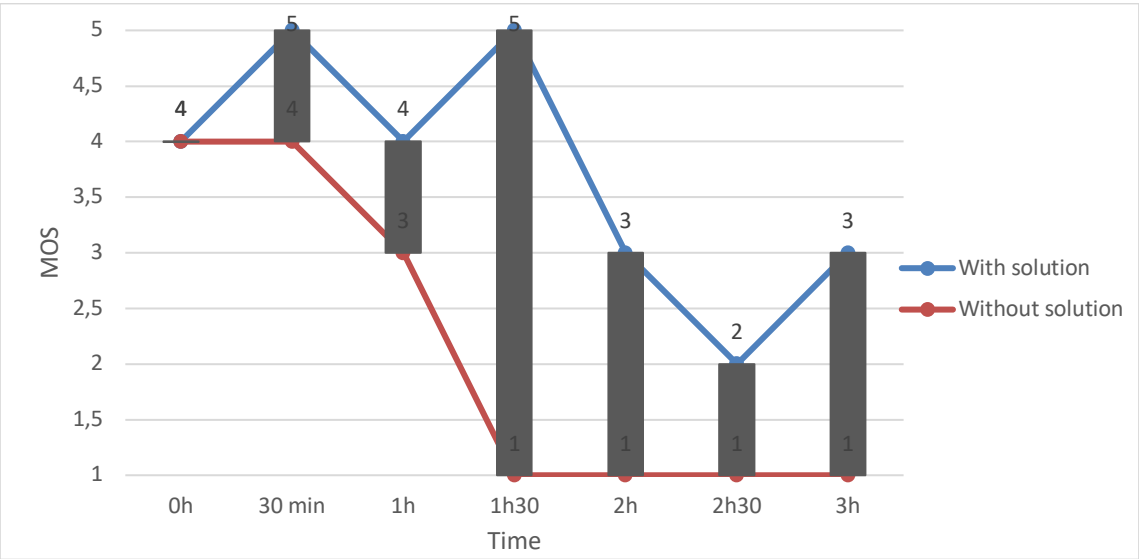


Fig.8 Average value of MOS evolution according to the number of users with proposed solution applied and without

Finally, figure 8 shows the differences between the MOS average value between the two conditions (with and without applying the solution). When network parameters are good, both solutions give good users satisfaction with a little advantage to our solution due to the adaptation with users' parameters which optimize users' resources. However, when the network parameters begin to fail, the gap between the studied solutions become more and more important. This is due to the congestion avoidance mechanism. In time (1h 30) when our solution was not used, the network became congested, the video streaming stopped and the users started showing signs of dissatisfaction. But with our solution (time :1h30) the video streaming kept working even though a weak performance was observed. However, the users were less satisfied than if they were in a good network condition but at least the video streaming kept working.

5. Conclusion

Across this paper, we presented an improvement of an existing adaptive video streaming solution. The improvement was realized in a MEC architecture. The main idea behind this proposal was to relieve users and servers from any additional tasks. All additional tasks are adapted and transferred to the MEC mobile agent to optimize users and servers' resources. A new congestion control mechanism was integrated. Emulation was used to test our proposal. The obtained results show how far our proposal improves the user's QoE and network QoS. As perspectives, we plan to improve the user manager system and their parameters, fix batteries impact update in our system, add more user's parameters and finally do more tests with additional terminals and more users (emulation with real cases).

Conflict of interest statement: We have no conflicts of interest to disclose.

DATA AVAILABILITY STATEMENT: Data sharing not applicable to this article as no datasets were generated or analyzed during the current study

References

1. Y Douga, M Bourenane, Y Hadjadj-Aoul, A Mellouk (2015) TCP based-user control for adaptive video streaming. *Multimedia Tools and applications*. <https://doi.org/10.1007/s11042-015-2857-1>.
2. L Yue, P Frangoudis, Y Hadjadj-Aoul, B Philippe (2016) A Mobile Edge Computing–Based Architecture for Improved Adaptive HTTP Video Delivery. *IEEE Conference on Standards for Communications and Networking (CSCN)*. <https://doi.org/10.1109/CSCN.2016.7784892>
3. L Roberto, S Marta, T Matías (2019) A Network-Layer QoE Model for YouTube Live in Wireless Networks. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2019.2918433>.
4. S Schwarzmann, C Cassales, M Bosk, H Liu, R Trivisonno, T Zinner (2019) Estimating Video Streaming QoE in the 5G Architecture Using Machine Learning. *Proceedings of the 4th Internet-QoE Workshop on QoE-based Analysis and Management of Data Communication Networks*. <https://doi.org/10.1145/3349611.3355547>.
5. M T Vega, C Perra, F De Turc, A Liotta (2018) A Review of Predictive Quality of Experience Management in Video Streaming Services *IEEE transactions on broadcasting*. <https://doi.org/10.1109/TBC.2018.2822869>.
6. K Minsu, C Kwangsue (2020) Edge Computing Assisted Adaptive Streaming Scheme for Mobile Networks. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2020.3047373>.
7. S Wanxin, L Qing, Z Ruishan, S Gengbiao, J Yong, Y Zhenhui, M Gabriel-Miro (2021) QoE Ready to Respond: A QoE-aware MEC Selection Scheme for DASH-based Adaptive Video Streaming to Mobile Users. *MM '21: Proceedings of the 29th ACM International Conference on Multimedia*. <https://doi.org/10.1145/3474085.3475325>.
8. A Mehrabi, M Siekkinen, A Yla-Jaaski (2019) Edge Computing Assisted Adaptive Mobile Video Streaming. *IEEE transactions on mobile computing*. <https://doi.org/10.1109/TMC.2018.2850026>.
9. N Barman, M G Martini (2019) QoE Modeling for HTTP Adaptive Video Streaming-A Survey and Open Challenges. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2019.2901778>.
10. M Till Beck, M Werner, S Feld, T Schimper (2014) : A Taxonomy. *AFIN 2014: The Sixth International Conference on Advances in Future Internet*.

11. R Xie, X Jia, K Wu (2020) Adaptive Online Decision Method for Initial Congestion Window in 5G Mobile Edge Computing Using Deep Reinforcement Learning. IEEE journal on selected areas in communications. <https://doi.org/10.1109/JSAC.2019.2959187>.
12. R Eman, N Arvind, K D Udhaya, A K Rostand, Q Feng, Z Zhi-Li (2021) Case for 5G-aware video streaming applications. 5G-MeMU '21: Proceedings of the 1st Workshop on 5G Measurements, Modeling, and Use Cases. <https://doi.org/10.1145/3472771.3474036>.
13. U Mikko, H Antti (2021) Evaluation of Live Video Streaming Performance for Low Latency Use Cases in 5G. Joint European Conference on Networks and Communications & 6G Summit. <https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482605>.
14. B Li, S Yu, J Su, J Ou, D Fan (2022) Computation Offloading in Multi-UAV-Enhanced Mobile Edge Networks: A Deep Reinforcement Learning Approach. Wireless communications and Mobile Computing. <https://doi.org/10.1155/2022/6216372>.
15. Y Gao, W Wu, J Dong, Y Yin, P Si (2020) Deep Reinforcement Learning based Node Pairing Scheme in Edge-chain for IoT Applications. 2020 IEEE Global Communications Conference. <https://doi.org/10.1109/GLOBECOM42002.2020.9322205>.
16. I A Elgendy, W Zhang, H He, H B Gupta, A Abd El-Latif (2021) Joint computation offloading and task caching for multi-user and multi-task MEC systems: reinforcement learning-based algorithms. Wireless Networks. <https://doi.org/10.1007/s11276-021-02554-w>.
17. M Kim, K Chung (2022) Reinforcement Learning-Based Adaptive Streaming Scheme with Edge Computing Assistance. Procedia Computer Science. <https://doi.org/10.3390/s22062171>.
18. ISO/IEC JTC1/SC29/WG11, MPEG-DASH (2012) The Standard for Multimedia Streaming Over Internet.
19. V Adhikari, Y Guo, F Hao, M Varvello, V Hilt, M Steiner, Z Zhang (2012) Unreeling netflix: Understanding and improving multi-CDN movie delivery. 2012 Proceedings IEEE INFOCOM. <https://doi.org/10.1109/INFOCOM.2012.6195531>.



Yassine Douga is an associate professor in computer science University Saad Dahleb Blida Algeria. He is member and researcher in the LRDSI laboratory, his main research topics are: wireless networks, WSN, network protocols, video streaming, QoS, QoE.



Yassine Hadjadj-Aoul is currently working as an associate professor at the University of Rennes 1, France, where he is also a member of the IRISA Laboratory and the INRIA project-team Dionysos. He received a B.Sc. in computer engineering with high honours from Mohamed Boudiaf University, Oran, Algeria, in 1999. Dr. Hadjadj received his Master's and Ph.D. degrees in computer science from the University of Versailles, France, in 2002 and 2007, respectively. He was an assistant professor at the University of Versailles from 2005 to 2007, where he was involved in several national and European projects such as NMS, IST-ATHENA, and ISTIMOSAN. He was also a post-doctoral fellow at the University of Lille 1 and a research fellow, under the EUFP6 EIF Marie Curie Action, at the National University of Dublin (UCD), where he was involved in the DOM'COM and IST-CARMEN projects, which aim at developing mixed Wi-Fi/WiMAX wireless mesh networks to support carrier grade

services. His main research interests concern the fields of wireless networking, multimedia streaming architectures and protocols, congestion control protocols and QoS provisioning, and satellite and space communications. Dr. Hadjadj has been on the technical program committee of different IEEE conferences, including Globecom, ICC, VTC, PIMRC and IWCMC. His work on multimedia and wireless communications has led to more than 40 technical papers in journals and international conference proceedings.



Malika Bourenane received her diploma (doctorate of state) in computer science from University Es-Senia Oran Algeria. Since 1984 she has been a permanent teacher at department of computer science. She is a researcher in the computer science laboratory and her main research interests include mobile ad-hoc networks, quality of service networks, queuing and scheduling theory, dynamic routing and Networks policy-based management and Real Time Application. She is member of international scientific committees of international conferences and journal.



Abdelhamid Mellouk is Full Professor at University of Paris-Est C-VdM (UPEC), Networks & Telecommunications (N&T) Department and LiSSi Laboratory, France. Head of several executive national and international positions, he was the founder of the Network Control Research activity at UPEC with extensive international academic and industrial collaborations. His general area of research is in adaptive real-time control for high-speed new generation dynamic wired/wireless networks in order to maintain acceptable Quality of Service/Experience for added-value services. He is an active member of the IEEE Communications Society and has held several offices including leadership positions in IEEE Communications Society Technical Committees.