EURASIP Journal on Image and Video Processing

RESEARCH Open Access

# Content and buffer status aware packet scheduling and resource management framework for video streaming over LTE system

Lijun He* and Fan Li

**Abstract**

With the development of the video encoding and wireless communication technologies, DASH (Dynamic Adaptive Streaming over HTTP) services have an increasing and great share of all the mobile services. However, we find some problems which still need to be addressed for DASH service optimization: (1) the limitation of the video segment representations cannot keep pace with the change of wireless channel states; (2) the characteristics of the video transmission have been not fully utilized in current DASH services; and (3) long interruption time will yield poor QoE (Quality of Experience). To solve these problems, we build a new transmission architecture by modifying the traditional TCP transmission flow. We first add a TCP (Transmission Control Protocol) proxy responsible for discarding the expiry packets automatically before putting them into the TCP sending window and then build a "ACK (acknowledgement) reconstruction module" to reconstruct the received ACK for concealing the packets discarded by the packet scheduler. Based on the new framework, we explore the interdependence among all the packets from the encoder to indicate the importance of each packet and update the interdependence relationship of every scheduling period based on the feedback ACK information. At MAC (Media Access Control) layer, a buffer status estimation module is employed to estimate the client buffer and playback information, which can be used to calculate the packet urgency. Then, a cross-layer design, which consists of an application layer of media server and client, TCP layer, MAC layer, and physical layer, is formulated and the packet scheduling and resource allocation can be jointly optimized. At client, through the analysis of the status of client buffer and MAC queue, an adaptive segment request scheme is developed to determine how and when to send the segment request. Simulation results show that the proposed algorithms can efficiently improve the received video quality as well as the playback continuity compared with other existing algorithms.

**Keywords:** DASH service, Packet importance, Buffer status estimation, Packet scheduling and resource management, Adaptive segment request

## 1 Introduction

### 1.1 Motivation

With the development of the video compression technology and the wireless network, video streaming in wireless system has taken a growing part of mobile service and attracted more and more attention. Representing the next generation in mobile technology, LTE system based on OFDMA (orthogonal frequency division multiple access) and MIMO (multiple-input multiple-output) technologies can promise faster speed and better coverage. Therefore, it is crucial to investigate how to provide satisfactory multiclient video streaming services to mobile clients by using this promising technology.

Researches have been carried out to enable high-quality video services like HTTP streaming over TCP (Transmission Control Protocol), where each video sequence is pre-encoded into many segments which cannot start to play until all the packets of it have been received [1, 2]. However, besides the high received video quality, continuous playback experience is also an important issue to

*Correspondence: jzb2016125@mail.xjtu.edu.cn
School of Electronics and Information Engineering, Xi'an Jiaotong University, Xianning Road, 24105 Xi'an, China

be addressed from the perspective of the end clients. In addition, due to the error concealment strategy employed in the decoding, some packet loss can be tolerated for video applications. Waiting for the arrivals of all the packets inevitably leads to some unnecessary interruptions for the playback. There are some prior works [3–5] in which the playback time of each packet was fixed. Those algorithms can provide continuous playback but may result in large fluctuation of the packet loss rate because of the time-varying channel states. To get a better trade-off between the video quality and the playback continuity, we introduce a maximum allowed waiting time for the playback of each frame. In our multi-client video streaming system, the packets whose arrival times expire the maximum allowed playback time are discarded. Besides time-varying wireless channel states and the limited availability of resources, unique characteristics of video packets such as the packet priority and stringent playback deadline are also considered in our proposed packet scheduling and RB assignment scheme.

To improve the system performance of video streaming, it is equally important to keep the client buffer and MAC queue at a desired level. Conventional strategies usually request segment with fixed time interval [1] or only consider the status of client buffer [6]. All of those algorithms may result in a low level of client buffer but too many packets accumulated in MAC queue due to the bad channel states. In a scenario with limited MAC queue buffer size, the packet loss rate would increase due to the overflow of the MAC queue. As a result, we propose a MAC (Medium Access Control) queue and client buffer aware segment request strategy to determine the appropriate time to send the segment request.

Since only the above two schemes cannot provide the best system performance, to make the packet scheduling more flexible, many prior works have been carried out to select the most appropriate video presentation or even code the video source to match the time-varying channel states [7–10] or control the client buffer length [1]. However, none of them consider the status of MAC queue. Therefore, we propose a rate adaption strategy based on the MAC queue and the client buffer.

### 1.2 Related work

The MLWDF scheduling rule in [11] aimed at maximizing the system throughput by considering the packet delay rather than the queue length. Considering both current channel conditions and the packet delay makes MLWDF more appropriate for real-time applications. The authors in [12–17] paid their attention to DASH service optimization by determining the segment adaptation and resource allocation scheme. The authors in [13] presented a system framework description of DASH service and mentioned many key factors that we should consider in DASH service optimization. The authors in [14] proposed a QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP. They first designed a mapping scheme from scalable video coding layers to DASH layers. Then, they developed a cross-layer design to achieve the resource allocation by considering both the video content and the client buffer status. They also proposed a rate adaptation scheme based on the client buffer status. In general, these works are worthy to be referred and investigated. However, the limitation of the video segment representations cannot keep pace with the change of wireless channel states. Moreover, the characteristics of the video transmission have been not fully utilized in current DASH services and long interruption time will yield poor QoE. Some recent research focused on packet importance and the delay requirements. Our work [18] focused on the video content and estimated the packet importance based on the extracted information from the compressed video. Similarly, [19] employed the video compression characteristics to indicate the packet importance. The wireless channel condition was also considered by using the queue length, input rate, and output rate. However, the packet urgency has not been considered in the above work. In [3–5, 20, 21], the delay requirement depended on the time interval during which the packets can be waiting to be scheduled in MAC queue. They can have a good effect on the video streaming over UDP because the clients experience continuous playback all the time. For instance, the authors in [22] assumed that there is only one frame waiting in MAC queue and the packets of this frame which cannot be transmitted in a certain time interval will be dropped. For the video streaming over TCP, the client will wait until all the packets are received. In this case, different clients will suffer different playback experiences and different playback buffer status. The previous playback experience will affect the following playback condition. Thus, it is inappropriate to use the waiting time in the MAC queue to determine the delay requirement. To accurately examine the packet urgency, the authors in [23–25] analyzed the distribution of the packet display time and proposed a playback control algorithm for real-time video services. In [24], the frame type and its corresponding playback time are considered to improve the received video quality and playback experience. However, their proposed models cannot be adapted in wireless systems because of the varying channel state. Furthermore, since many error concealment strategies can be employed at the decoder and the loss of different video packets will have different effects on the received video quality, it is not necessary to transmit all the video packets for the video service especially in the system with limited wireless resource.

For the segment request, a client usually requests video segments with predetermined time interval (around 2 s for IIS and around 3 s for Netflix in [1]) without considering any information about client in the conventional mechanisms. This may result in some clients having too many frames stored at the client buffers while the others suffer playback buffer underflow. The authors in [6] developed the segment request strategy based on the playback information. But they ignored the information of the MAC queue and the time-varying channel states. With the time when to send the segment request known, how to select the requested segment to adapt to the time-varying channel states becomes a crucial problem to be solved. H.264 SVC (Scalable Video Coding) technology is widely employed in many researches [26–29] to provide different video qualities by encoding each video sequence into many different layers. The authors in [26, 29] exploited the cross-layer design framework for a dynamic scalable video adaptation in varying network capacity. In order to increase the number of the clients that the system can support, the authors in [27] focused on the multi-client video transmission over OFDMA-based communication system by combining the adaptive subcarrier allocation and bit loading with the transmission of the H.264 SVC encoded video sequences. Since scalable video usually consists of multiple scalable layers with different importance which brings high implementation complexity, most of the researches still focus on the pre-encoded video applications. In [7, 8], some studies have been carried out to determine the optimal coding rate that the server should provide. The authors in [7] first allowed the server to drop some unimportant frames by considering the channel states. Then, according to the dropping strategy, the optimal coding rate can be determined by using a simple rate-distortion model. In [8], to aim for maximum QoE (Quality of Experience) for clients, maximum system video throughput, and QoE fairness among clients, a cross-layer optimized coding rate adaptation scheme was proposed. However, both of them focus on the received video quality but neglect another important issue of video streaming: playback continuity. To improve the playback experience, [1] studied a no-reference QoE (quality of experience) monitoring module for adaptive HTTP streaming. The authors showed the playback interruptions can be reduced by changing the QP (quantitative parameters) of the encoder to provide a presentation with a close coding rate to the estimation transmission rate. The authors in [9, 10, 30] formulated the video streaming process with multiple links. Gouache et al. [9] provided a method to estimate the transmission rate of the next scheduling period which can be used to determine the data distribution among the servers. In [10, 30], it is assumed that each client receive the video data from different servers or the other clients in the system. The bit rate can be obtained based on the source node selection strategy. In those algorithms, the bit rate was determined with the assumption that there is no packet in MAC queue waiting to be scheduled. In fact, when we send the next segment request, there may be some packets left in the MAC queue. And due to the different channel states of the end clients, their numbers of the remaining packets maybe different from each other. Since the remaining packets would affect the transmission of the next segment, the status of the clients' MAC queue should be considered in the bit rate determination.

### 1.3 Approach

In summary, we find some problems which still need to be addressed for DASH services over TCP as follows: (1) the limitation of the video segment representations cannot keep pace with the change of wireless channel states; (2) the characteristics of the video transmission have been not fully utilized in current DASH services; (3) long interruption time will yield poor QoE. Inspired by the above prior works, we propose a buffer status and content aware packet management framework in LTE system. Our work is novel in the following aspects.

- *A new transmission framework for DASH services*: We modify the transmission flow of the whole DASH service. With the purpose to transmit the packets which are really important to the received video quality as soon as possible and discard the ones that have little effect on the improvement of the QoE level of all the clients. We added a TCP proxy to control the main functions of the TCP layer. It is responsible for "Packet rearrange" which can discard the expiry packets automatically according to the feedback information from the packet urgency estimation at the MAC layer and the dropped packets will not be put into the "Packet sending window." At the MAC layer, we build a "ACK reconstruction module" to reconstruct the received ACK for concealing the discarded packets. With the aim of maximizing the sum utility of the scheduled packets, some packets with little utility will be discarded instead of wasting lots of time for their arrival at the client. At the client, we introduce a maximum allowed waiting time for each segment. The client does not need to wait until all the video packets arrive at the client. Then, based on the new transmission architecture, we can improve the QoE of DASH service by jointly optimizing the application layer of media server and client, TCP layer, MAC layer, and PHY layer. The strategy of packet scheduling, resource allocation, and segment request can be obtained in our new framework.
- *A new packet importance estimation and update method based on the feedback ACK*: We explore the interdependence among all the packets to indicate the importance of each packet. Specially, we update

the interdependence relationship between the packets waiting to be scheduled and the packets already transmitted or lost based on the feedback ACK information. This update mechanism makes the packet importance more close to the real contribution that the packet can create. Moreover, we take into consideration the playback information to predict the packet urgency.

- *A rate adaptation scheme based on both MAC queue and the client buffer status*: We consider the MAC queue status in our rate adaptation scheme. We use the playback information of client to estimate the continuous playback time that the packets in MAC queue can support. Then, by considering the MAC queue, client buffer, and the estimated transmission rate, we can acquire the urgency of each client and determine when and how to send the new segment request.

The rest of the paper is organized as follows. Section 2.1 describes the system model, and Section 2.2 presents how to calculate the packet utility. Section 2.3 show the problem formulation and solution to our proposed packet scheduling and resource management scheme. Section 2.4 presents a rate adaption scheme to derive when and which presentation should be requested and selected. Section 3 provides the discussion and the experimental results of the proposed algorithms. Finally, we draw the conclusion in Section 4. We provide a list of abbreviations used in this paper as shown in Table 1

**Table 1** A list of abbreviations

| | |
|---|---|
| LTE | Long Term Evolution |
| GoP | Group of pictures |
| MAC | Media Access Control |
| ACK | Acknowledgement |
| AM | Acknowledged mode |
| RB | Resource block |
| OFDMA | Orthogonal frequency division multiple access |
| MIMO | Multiple-input multiple-output |
| TCP | Transmission Control Protocol |
| SVC | Scalable video coding |
| QoE | Quality of Experience |
| CQIs | Channel quality indicator |
| TTI | Transmission time interval |
| MCS | Modulation and coding scheme |
| NS3 | Network Simulation 3 |
| AR | Adaptive request |
| PSNR | Peak signal-to-noise ratio |
| RLC | Radio link control |

## 2 Methods

### 2.1 System Model

Figure 1 shows the framework of this paper. In this system, $K$ video clients can request different video sequences they want. Each video sequence is pre-encoded into a series of video segments with different coding rates (different presentations) by employing the H.264/AVC encoder in advance. The segments are stored at the media server attached to eNodeB via a lossless wired network, and the clients can send the segment requests to the media server. The packet priority can be determined based on the information extracted from the encoder and the feedback ACK information.

We build a TCP proxy to control the main functions of TCP layer. It is responsible for "Packet rearrange," "ACK receiving window," and "Packet sending window." The module "Packet rearrange" can discard the expiry packets according to the feedback information from packet urgency estimation at the MAC layer automatically, and the dropped packets will not be put into the "Packet sending window." In other words, these packets discarded by TCP have no chance to be scheduled and transmitted. The packets not expiry will be put into "Packet sending window" according to the receiving ACK information from the "ACK receiving window."

At the MAC layer, the playback information can be estimated based on the feedback information from the clients. Considering the estimated playback information and the packet priority, we can calculate the importance of each packet. Based on the packet importance and the feedback CQIs (channel quality indicator), different RBs are assigned to different clients to maximize the system performance. We build a "ACK reconstruction module" to reconstruct the received ACK for concealing the discarded packets at the MAC layer. It depends on the received ACK information from the "ACK sending window" at the client and the packet scheduling strategy at the MAC layer. When we consider the packet utility during the resource allocation to maximize the sum utility of the scheduled packets, some packets with little utility will be discarded instead of wasting lots of time for their arrival at the client. The ACK of these discarded packets will be constructed and integrated with the received ACKs by the "ACK reconstruction module" and then sent to the "ACK receiving window" at the TCP layer. And thus, the dropped packets at the MAC layer are concealed during the TCP flow. In addition, at the MAC layer of eNodeB, we focus on the estimation of the playback time that the packets in MAC queue can support by using the estimated buffer status and the predicted transmission rate.

At the client, we build "ACK sending window" and "Packet receiving window" to simulate TCP flow. We give a maximum allowed waiting time for each segment. The

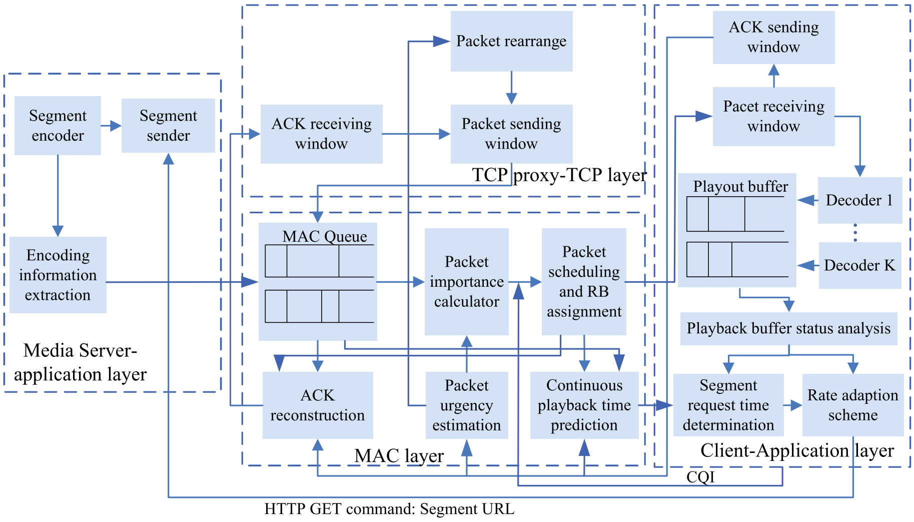
Segment encoder
Segment sender
Encoding information extraction
Media Server-application layer
Packet rearrange
ACK receiving window
Packet sending window
TCP proxy-TCP layer
MAC Queue
Packet importance calculator
Packet scheduling and RB assignment
ACK reconstruction
Packet urgency estimation
Continuous playback time prediction
MAC layer
ACK sending window
Pacet receiving window
Playout buffer
Decoder 1
Decoder K
Playback buffer status analysis
Segment request time determination
Rate adaption scheme
Client-Application layer
CQI
HTTP GET command: Segment URL

**Fig. 1** The proposed system framework

packets violate its deadline will be discarded by the MAC layer and TCP layer. The exact buffer information helps us to easily get how long the remaining buffered frames in client buffer can take. Based on the analysis of the client buffer and MAC queue, an adaptive segment request strategy is developed to send each segment request at an appropriate time. Knowing when to send the segment request, a rate adaption scheme is proposed based on the estimated transmission rate and the status of the client buffer and MAC queue to provide the presentation which should be selected.

### 2.2 Packet utility estimation

Packet utility in our work depends on two factors: the packet priority determined by the encoding information and the packet urgency determined by the playback information. In this paper, the packet priority is calculated in terms of the potential degradation caused in the video quality due to loss of the packet. Since the same method is applied to all the video sequences, we omit the client index in the following analysis. Let $\lambda$ denote the number of B frames inserted into two P frames. Let $D_{f,p}^{\text{loss}}$ denote the distortion caused by the loss of the packet $p$ of frame $f$ ($f$ is the encoding index of the frame), which can be expressed by

$$D_{f,p}^{\text{loss}} = D_{f,p}^{\text{rec}} + D_{f,p}^{\text{ref}} \tag{1}$$

It consists of two parts:

(1) $D_{f,p}^{\text{rec}}$ is the relative distortion caused by the reconstructed packet. Here, the distortion $D_{f,p}^{\text{rec}}$ can be obtained by using the method presented in [31].

(2) $D_{f,p}^{\text{ref}}$ is the relative distortion caused by the packets dependent on the current one. As shown in Fig. 2, if we set the encoding parameter Number Reference Frames = 1, I or P frame will be directly referred according to the following principles:

- The following two B frames ($f+1, \cdots, f+\lambda$) refer the frame $f$ for backward prediction.
- The following P frame $f+\lambda+1$ refers the frame $f$ for forward prediction.

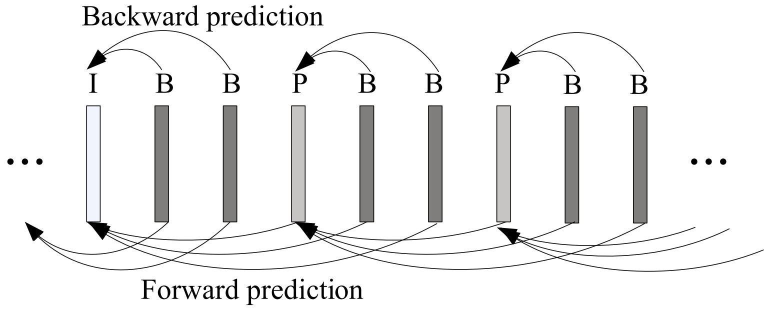

**Fig. 2** The encoding reference relationship with $\lambda = 2$ and Number Reference Frames $= 1$

- The following two B frames ($f+\lambda+2,\cdots,f+2\lambda+1$) refer the frame $f$ for forward prediction.

Let $\gamma_{f,p}$ be the sum times that each of the pixels of packet $p$ of frame $f$ is referenced. To calculate $\gamma_{f,p}$, the information of each packet such as MB (macroblock) partition modes and MVs should be extracted from the encoder. H.264/AVC encoder supports the intra-coded and inter-coded prediction modes and also provides abundant MB partition modes. Intra-coded prediction has 16 × 16 and 4 × 4 partition modes, and the inter-coded prediction mode possesses seven MB partition modes, namely 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8, and 4 × 4 as shown in Fig. 3. Let $\psi_{\eta,\theta}^{\widetilde{f},f,p}$ denote the set of the pixels of packet $p$ of frame $f$ referenced by the SB $\theta$ of the MB $\eta$ of the frame $\widetilde{f}$. The calculation of $\psi_{\eta,\theta}^{\widetilde{f},f,p}$ is described in Table 2.

Note that I frame is directly referenced by its following P frame. If the P frame is not the last P frame in this GoP, it is also referenced by its following P frame. Except for the first P frame, other P frames refer I frame undirectly. Consequently, the loss of I or P frame will cause the quality degradation of all the following frames in the same GoP. For P frame, the smaller frame encoding index is the more important. In this paper, the referenced times of each packet is calculated by considering both direct and undirect reference modes. Let $\beta$ specify the total number of I and P frames in one GoP. Any I or P frame is given another index called *reference frame index* to indicate its index among all the I and P frames in a GoP. Let $\pi_f$ denote the reference frame index of frame $f$, $0 \le \pi_f \le \beta - 1$. Let $\zeta(\bullet)$ denote the number of the elements of the input set. For the packet $p$ of frame $f$:

- If $\pi_f = \beta - 1$, frame $f$ is the last P frame in its GoP and it is only referenced by its following B frames in direct mode. $\gamma_{f,p}$ can be computed by

$$\gamma_{f,p} = \sum_{\widetilde{f}=f+1}^{f+2\lambda+1} \sum_{\eta,\theta} \zeta\left(\psi_{\eta,\theta}^{\widetilde{f},f,p}\right) \tag{2}$$

- If $\pi_f < \beta - 1$, frame $f$ may be referenced by its following B frames and P frames in direct or undirect mode. $\gamma_{f,p}$ can be computed by

$$\gamma_{f,p} = \underbrace{\sum_{\widetilde{f}=f+1}^{f+2\lambda+1} \sum_{\eta,\theta} \zeta\left(\psi_{\eta,\theta}^{\widetilde{f},f,p}\right)}_{\text{direct reference mode}} + \underbrace{\sum_{\widetilde{f}=f+\lambda+2}^{f+\lambda(\beta-\pi_f)+\beta-\pi_f-1} \sum_{\eta,\theta} \zeta\left(\widehat{\psi}_{\eta,\theta}^{\widetilde{f},f,p}\right)}_{\text{undirect reference mode}} \tag{3}$$

where $\widehat{\psi}_{\eta,\theta}^{\widetilde{f},f,p}$ is the region of pixels of packet $p$ of frame $f$ referenced undirectly by SB $\theta$ of MB $\eta$ frame $f$. To calculate $\widehat{\psi}_{\eta,\theta}^{\widetilde{f},f,p}$, we need to calculate the reference region of frame $\widehat{f}$ directly referenced by frame $\widetilde{f}$, $\phi_{\eta,\theta}^{\widetilde{f},\widehat{f}}$ first. Then, according to all the reference relationships between the undirect reference frames such as $\Psi(\widehat{f},\widehat{f}-\lambda-1)$, $\cdots$, $\Psi(f-\lambda-1,f)$, $\widehat{\psi}_{\eta,\theta}^{\widetilde{f},f,p}$ can be obtained finally, where $\Psi(f_1,f_2)$ is the set of the reference relationships for all the SBs between frame $f_1$ and $f_2$, which can be easily acquired

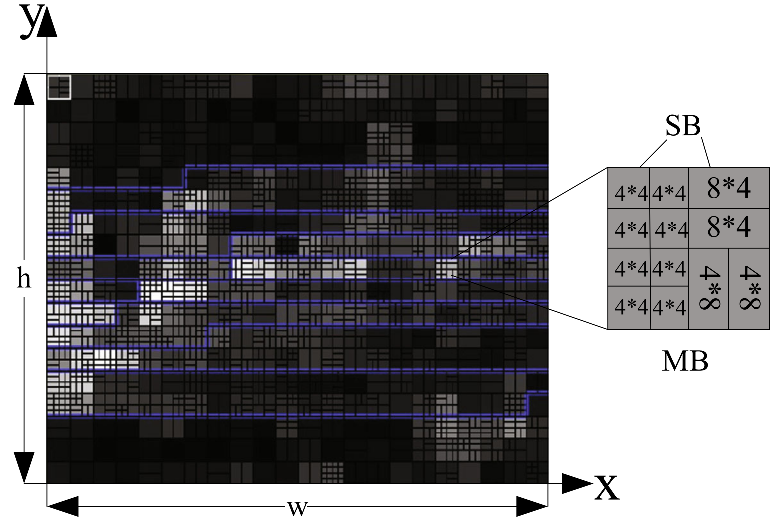

**Fig. 3** The coordinate system of one frame

**Table 2** Calculation of $\psi_{\eta,\theta}^{\widetilde{f},f,p}$

| | |
|---|---|
| Step 1 | Calculate the location of the starting point of the $\eta_{\text{th}}$ MB of $\widetilde{f}_{\times th}$ frame, $\left(m_x^{\widetilde{f}}(\eta), m_y^{\widetilde{f}}(\eta)\right)$. |
| Step 2 | Obtain the position of the SB $\theta$ of MB $\eta$, $\left(s_x^{\widetilde{f}}(\eta,\theta), s_y^{\widetilde{f}}((\eta,\theta))\right)$ based on the extracted MB partition modes. |
| Step 3 | With knowing the motion vector the SB $\theta$ of MB $\eta$ between $f$ and $\widetilde{f}$, $(MVx^{\widetilde{f},f}(\eta,\theta), MVy^{\widetilde{f},f}(\eta,\theta))$, the location of the starting point of the SB used as the reference of frame $f$ can be derived by $\begin{cases} l_x^{\widetilde{f},f}(\eta,\theta) = s_x^{\widetilde{f}}(\eta,\theta) + MVx^{\widetilde{f},f}(\eta,\theta) \\ l_y^{\widetilde{f},f}(\eta,\theta) = s_y^{\widetilde{f}}(\eta,\theta) + MVy^{\widetilde{f},f}(\eta,\theta) \end{cases}$ |
| Step 4 | The region of the pixels of the $f$ frame referenced by the SB $\theta$ of the MB $\eta$ of the frame $\widetilde{f}$ can be expressed by $\phi_{\eta,\theta}^{\widetilde{f},f} = \left\{(x,y) \mid l_x^{\widetilde{f},f}(\eta,\theta) \le x \le l_x^{\widetilde{f},f}(\eta,\theta) + w^{\widetilde{f}}(\eta,\theta) - 1, l_y^{\widetilde{f},f} \le y \le l_y^{\widetilde{f},f} + h^{\widetilde{f}}(\eta,\theta) - 1\right\}$,<br>where $w^{\widetilde{f}}(\eta,\theta)$ and $h^{\widetilde{f}}(\eta,\theta)$ are the width and the height of the SB $\theta$ of MB $\eta$, respectively. |
| Step 5 | Let $\varphi_{f,p}$ specify the region of the pixels of packet $p$ of frame $f$. Eliminate the elements in $\phi_{\eta,\theta}^{\widetilde{f},f}$ that do not belong to the set $\varphi_{f,p}$ and obtain the set of the pixels referenced by the SB $\theta$ of the MB $\eta$ of the frame $\widetilde{f}$, $\psi_{\eta,\theta}^{\widetilde{f},f,p}$, $\psi_{\eta,\theta}^{\widetilde{f},f,p} = \phi_{\eta,\theta}^{\widetilde{f},f} \cap \varphi_{f,p}$. |
| Step 6 | Repeat step 1–step 5 if there are unprocessed SBs of frame $\widetilde{f}$. |

by using the method presented in Table 2. Based on the above analysis, $D_{f,p}^{\text{ref}}$ can be computed by

$$D_{f,p}^{\text{ref}} = \begin{cases} D_{f,p}^{\text{rec}} \frac{\gamma_{f,p}}{\zeta(\varphi_{f,p})}, \text{if } A_{\hat{f},\hat{p}} = 1 \\ D_{f,p}^{\text{con}} \frac{\gamma_{f,p}}{\zeta(\varphi_{f,p})}, \text{if } A_{\hat{f},\hat{p}} = 0 \end{cases} \tag{4}$$

where $A_{\hat{f},\hat{p}}$ is a binary variable which indicates whether the packet $\hat{p}$ of frame $\hat{f}$ is received or lost. $A_{\hat{f},\hat{p}} = 1$ is equal to the successful delivery and $A_{\hat{f},\hat{p}} = 0$ otherwise. The value of $A_{\hat{f},\hat{p}}$ depends on the feedback ACK information. $D_{f,p}^{\text{con}}$ is the distortion caused by the loss of its reference packet $\hat{p}$ of frame $\hat{f}$. Finally, $D_{f,p}^{\text{loss}}$ can be obtained according to (1). In the following, $D_{f,p}^{\text{loss}}$ is regarded as the priority of packet $p$ of frame $f$ and at the RLC (radio link control) layer of LTE system, each video packet is fragmented into many small packets. It is assumed that the packets attributed to the same packet have the same priority. In MAC queue, we can acquire the packet priority from the packet header denoted by $\text{pri}_{k,m}$.

We use the feedback information such as ACKs of the received packets from the clients to predict the playback information and then examine the packet urgency. Let $\widetilde{F}_c(k)$ denote the estimated index of the frame being played of client $k$. Since the frames of one segment should be decoded simultaneously, the packet urgency of all the video packets of the frames of the same segment should be the same. Thus, the packet urgency $\mu_{k,m}$ can be expressed by

$$\mu_{k,m} = P_{k,m} - \widetilde{F}_c(k) \tag{5}$$

where $P_{k,m}$ is the index of the first frame of the segment waiting in the MAC queue. For the packets of the same segment, the values of $P_{k,m}$ are the same. By employing the method proposed in [3], the scheduling probability of packet $m$ for client $k$, $\rho_{k,m}$, can be defined as $\rho_{k,m} = e^{-\mu_{k,m}}$.

Then, considering the packet priority and the packet urgency, the utility of packet $m$ of client $k$, $U_{k,m}$, can be calculated by

$$U_{k,m} = e^{-\mu_{k,m}} \times \text{pri}_{k,m} \tag{6}$$

## 2.3 Problem formulation and solution

### 2.3.1 Problem formulation

We formulate the packet scheduling and resource management into a new mathematical model with the objective to maximize the weighted sum of utilities of all the scheduled packets of all the clients:

$$\begin{aligned} &\max \sum_{k=1}^{K} w_k \sum_{m=1}^{M_k} U_{k,m}\tau_{k,m} \\ &s.t. \\ &(c1) \sum_{n\in\Omega} a_{k,n} \sum_{j=1}^{q(N_k)} b_{k,j} r_j \ge \sum_{m=1}^{M_k} \tau_{k,m} R_{k,m}, 1 \le k \le K \\ &(c2) \tau_{k,m} \in \{0,1\}, 1 \le m \le M_k, 1 \le k \le K \\ &(c3) \sum_{j=1}^{q(N_k)} b_{k,j} = 1,\ b_{k,j} \in \{0,1\}, 1 \le k \le K \\ &(c4) \sum_{k=1}^{K} a_{k,n} = 1,\ a_{k,n} \in \{0,1\}, 1 \le n \le N \end{aligned} \tag{7}$$

where

- $\Omega$: the set of the total available RBs.
- $N$: the number of the available RBs in this system.
- $M_k$: the number of the packets in client $k$'s MAC queue.

- $w_k$: client $k$'s weight, a positive constant calculated by $w_k = \sum_{k=1}^{K} (\widetilde{N}_b(k))/\widetilde{N}_b(k)$, such that $\sum_{k=1}^{K} (w_k) = 1$. $\widetilde{N}b(k)$ is the predicted number of frames waiting to be played at client buffer. The aim is to balance the playback continuity experience among the clients. $\widetilde{N}b(k)$ can be predicted by using the feedback ACK information with RLC mode AM. It can also be fedback from the client by employing the playback buffer status analysis module at client.
- $\tau_{k,m}$: $\tau_{m,k} = 1$ means the packet $m$ of client $k$ is selected to be scheduled and $\tau_{m,k} = 0$ otherwise.
- $a_{k,n}$: $a_{k,n} = 1$ indicates the RB $n$ is assigned to client $k$ and $a_{k,n} = 0$ otherwise.
- $N_k$: the set of RBs assigned to client $k$.
- $q(N_k)$: the maximum MCS (Modulation and Coding Scheme) that client $k$ can support based on $N_k$.
- $b_{k,j}$: $b_{k,j} = 1$ means client $k$ uses MCS $j$ and $b_{k,j} = 0$ otherwise.
- $r_j$: the transmission capacity of each RB by using MCS $j$.
- $R_{k,m}$: the size of the packet $m$ of client $k$.

Constraint (c1) shows that the transmission capacity of client $k$ should be enough to transmit its scheduled packets. We assume that all the scheduled packets can be transmitted to the client successfully by advanced channel coding. Constraint (c2) indicates that each packet can be scheduled or not at each scheduling period. Constraint (c3) means that each client can only choose one MCS. The last constraint implies that one RB can only be assigned to one client.

#### 2.3.2 *Packet scheduling and resource management scheme*

From the mathematical expression in Eq. (7), the relationship between the receive time and other parameters cannot be explicitly evaluated without restrictive assumptions on the arrival process and the time-varying channel states. Thus, we can not obtain a mathematically optimal solution of the above problem in its most generic form. Due to this, we develop a heuristic suboptimal solution, which jointly performs RB assignment and packet scheduling by employing our proposed utility function.

To solve the problem (7), a cost function $C_{k,n}$ is developed to present the increment of the sum utilities of the clients brought by assigning RB $n$ to client $k$. $C_{k,n}$ can be expressed by

$$C_{k,n} = f(\tau_{k,m}, a_{k,n}, b_{k,j}, N_k \cup \{n\}) - f(\tau_{k,m}, a_{k,n}, b_{k,j}, N_k) \tag{8}$$

where $N_k$ is the set of the RBs already assigned to client $k$. The function $f(\bullet)$ can be expressed as

$$\begin{aligned} & f(\tau_{k,m}, a_{k,n}, b_{k,j}, N_k) = \max\ w_k \sum_{m=1}^{M_k} U_{k,m}\tau_{k,m} \\ & s.t. \\ & (c1)\ \sum_{n\in N_k} a_{k,n} \sum_{j=1}^{q(N_k)} b_{k,j} r_j \geq \sum_{m=1}^{M_k} \tau_{k,m} R_{k,m} \\ & (c2)\tau_{k,m} \in \{0,1\}, 1 \leq m \leq M_k \\ & (c3)\ \sum_{j=1}^{q(N_k)} b_{k,j} = 1,\ b_{k,j} \in \{0,1\} \end{aligned} \tag{9}$$

Achieving problem (7), we should assign RB $n$ to the client with the maximum $C_{k,n}$. To get $C_{k,n}$, we need to solve $f(\tau_{k,m}, a_{k,n}, b_{k,j}, N_k)$ first. Next, we will provide how to calculate $f(\tau_{k,m}, a_{k,n}, b_{k,j}, N_k)$.

For a certain client with fixed RB set, the utility is an increasing function of the transmission capacity. The maximum capacity these RBs can support can be calculated by solving the following problem:

$$\begin{aligned} & \max \sum_{n\in N_k} a_{k,n} \sum_{j=1}^{q(N_k)} b_{k,j} r_j \\ & s.t. \\ & (c1)\ \sum_{j=1}^{q(N_k)} b_{k,j} = 1,\ b_{k,j} \in \{0,1\} \\ & (c2)\ \sum_{k=1}^{K} a_{k,n} = 1,\ a_{k,n} \in \{0,1\}, \{n\} \in N_k \end{aligned} \tag{10}$$

Then, we can get the optimal $\left(a^*_{k,n}, b^*_{k,j}\right)$, and then the transmission capacity $\sum_{n\in N_k} a^*_{k,n} \sum_{j=1}^{q(N_k)} b^*_{k,j} r_j$ can be obtained according to [32]. Knowing the transmission capacity, the optimal packet scheduling strategy $\tau^*_{k,m}$ can be obtained by solving the following problem:

$$\begin{aligned} & \max\ w_k \sum_{m=1}^{M_k} U_{k,m}\tau_{k,m} \\ & s.t. \\ & (1)\ \sum_{n\in N_k} a^*_{k,n} \sum_{j=1}^{q(N_k)} b^*_{k,j} r_j \geq \sum_{m=1}^{M_k} \tau_{k,m} R_{k,m} \\ & (2)\tau_{k,m} \in \{0,1\} \end{aligned} \tag{11}$$

Optimal $\tau^*_{k,m}$ can be easily acquired by our method proposed in [33]. Then, we can have

$$f\left(\tau^*_{k,m}, a^*_{k,n}, b^*_{k,j}, N_k\right) = \sum_{m=1}^{M_k} \tau^*_{k,m} \tag{12}$$

The detailed process of the packet scheduling and resource management scheme is described in Table 3.

### 2.4 Adaptive segment request scheme

In this section, an adaptive segment request strategy is proposed to determine when to send the segment request and which presentation should be selected.

**Table 3** Packet scheduling and resource management scheme

```
1   **Initialize** the total RB set $\Omega = \{1, ..., N\}$ and the RBs determined to be assigned to client $k$, $N_k = \emptyset$, for $1 \le k \le K$. $a_{k,n} = 0, b_{k,j} = 0$, for $1 \le k \le K, n \in \Omega$.
2   **While** $(\Omega \ne \emptyset)$
3   **For** $n \in \Omega$
5   **For** $1 \le k \le K$
6   Update $N_k = N_k \cup \{n\}$ and $a_{k,n} = 1$.
7   Solve the problem (10), update $a^*_{k,n} = 1$ and $b^*_{k,j} = 1$.
8   Solve the problem (11), get $\tau^*_{k,m}$.
9   Calculate $f\left(\tau^*_{k,m}, a^*_{k,n}, b^*_{k,j}, N_k\right) = \sum_{m=1}^{M_k} \tau^*_{k,m}$ and $C_{k,n}$
10  **End** for For
11  $k^* = \underset{1 \le k \le K}{argmax}\, C_{k,n}$
12  Update $N_k = N_k \setminus \{n\}$, $a_{k,n} = 0$ and $b_{k,j^*} = 0$ for $k \ne k^*$. $\Omega = \Omega \setminus \{n\}$
13  **End** for For
14  **End** for while
15  **Output**optimal packet scheduling $\tau^*_{k,m}$ and RB assignment $N^*_k$.
```

**Table 4** Procedure of estimation of $T_{\mathrm{mac}}(k)$

```
1   Initialize $RP_k = \{1, \cdots, M_k\}$, $T_{\mathrm{mac}}(k) = 0$ and $\widehat{T}_{\mathrm{mac}}(k) = T_b(k)$, estimate the transmission rate $\widehat{r}_k$.
2   Calculate the transmission capacity $c_k = \widehat{T}_{\mathrm{mac}}(k) \times \widehat{r}_k$.
3   Determine optimal packet scheduling strategy $\tau_k$ for the packets in $RP_k$.
4   Update the set of unscheduled packets $RP_k$ according to $\tau_k$ and determine the number of frames that these new scheduled packets are attributed to, $N_{\mathrm{mac}}(k)$.
5   Calculate the playback time that these new received packets can support, $\widehat{T}_{\mathrm{mac}}(k) = N_{\mathrm{mac}}(k)/F_{pr}$. $F_{pr}$ is the playback frame rate.
6   Update the playback time that the packets in the MAC queue can support, $T_{\mathrm{mac}}(k) = T_{\mathrm{mac}}(k) + \widehat{T}_{\mathrm{mac}}(k)$.
7   **If** $\widehat{T}_{\mathrm{mac}}(k) > 0$ and $RP_k \ne \emptyset$, it means that during the time interval $\widehat{T}_{\mathrm{mac}}(k)$, the packets left in $RP_k$ still have the chance to be scheduled.
8   Do step 2, 3, 4, 5, 6.
9   **End for If**
10  Output $RP_k$ and $T_{\mathrm{mac}}(k)$
```

#### *2.4.1 Segment request time determination*

The segment request time determination is carried out at the client side, so we can get the accurate number of the unplayed but successfully decoded frames stored in client $k$'s playback buffer $N_b(k)$ and how long continuous playback these frames can support $T_b(k) = N_b(k)/F_{pr}$, where $F_{pr}$ is the frame rate. Since some packets in the MAC queue can be transmitted to the client side during $T_b(k)$ and therefore can bring extra continuous playback time, immediately sending the new segment request can not bring benefit to the client but may occupy the wireless resource which is really needed by other clients. Letting the packets in MAC queue are the ones that are really needed by the clients is the most important.

Let $T_{\mathrm{mac}}(k)$ denote how long the continuous playback time interval that the packets in MAC queue can support and $RP_k$ denote the set of the packets in MAC queue of client $k$ after the estimation. The estimation of $T_{\mathrm{mac}}(k)$ is shown in Table 4.

If $RP_k = \emptyset$, it means that all the packets in the MAC queue can arrive at the client on time and the client can enjoy $T_{\mathrm{mac}}(k) + T_b(k)$ continuous playback back at least. If $RP_k \ne \emptyset$, it means that the client $k$ will suffer interruption after the time interval $T_{\mathrm{mac}}(k) + T_b(k)$. Let $T_l(k)$ denote the playback time that the remaining packets in MAC queue will take, which can be calculated by $T_l(k) = N_l(k)/F_{pr}$, where $N_l(k)$ is the number of the remaining frames in the MAC queue. The transmission time they will take can be estimated by $T_r(k) = \mathrm{SM}(k)/\widehat{r}_k$, where $\mathrm{SM}(k)$ indicates the sum size of the left packets in the MAC queue. A guard time interval $T_g(k)$ is introduced to protect the client against suffering playback interruption. The appropriate time interval needed to send the next segment request, $T_i(k)$, can be computed by

$$T_i(k) = \max(T_{\mathrm{mac}}(k) + T_b(k) + T_l(k) + T_r(k) - T_g(k), 0) \tag{13}$$

#### *2.4.2 Rate adaption at client*

With the time when to send the segment request known, we need to determine which presentation should be selected. To keep the continuous playback, the next requested segment should be transmitted to the client during the time we estimated above. Based on the estimated transmission rate, we can determine the optimal coding rate we should request. With the optimal coding rate known, we can select the segment with most closest coding rate. Using the metadata contained in MPD (Media Presentation Description), the unique URL of the requested segment can be constructed. Then, the request will be sent to the media server by HTTP GET command. The media server will pick the segment with the unique URL and send it to the DASH client.

Let $S_{\mathrm{mac}}(k)$ denote the sum size of the packets in client $k$'s MAC queue, $S_{\mathrm{mac}}(k) = \sum_{m=1}^{M_k} R_{k,m}$. The transmission time taken by those packets, $\widehat{T}_{\mathrm{mac}}(k)$, can be estimated by

$$\widehat{T}_{\mathrm{mac}}(k) = \frac{S_{\mathrm{mac}}(k)}{\widehat{r}_k} \tag{14}$$

There are two possible transmission cases according to the value of $T_l(k)$:

- If $T_l(k) = 0$, all the packets in MAC queue can be transmitted to the client during the time interval $T_b(k) + T_{\text{mac}}(k)$. Therefore, to keep the continuous playback of client $k$, the next segment should be transmitted to client during the interval $T_b(k)+T_{\text{mac}}(k)-\widehat{T}_{\text{mac}}(k)$. Let $T_{\text{seg}}(k)$ be the duration of a segment for client $k$. Then, the coding rate that the client should recommend can be computed by

$$\text{pr}_k = \widehat{r}_k \frac{T_b(k) + T_{\text{mac}}(k) - \widehat{T}_{\text{mac}}(k)}{T_{\text{seg}}(k)} \qquad (15)$$

- If $T_l(k) > 0$, client $k$ would suffer interruption after the time interval $T_b(k) + T_{\text{mac}}(k)$. Since the left packets also take some time $\widehat{T}_l(k)$ to be transmitted and played, the next segment should arrive at client during the time interval $\widehat{T}_l(k)$. Then, the recommended coding rate can be acquired by

$$\text{pr}_k = \widehat{r}_k \frac{T_l(k)}{T_{\text{seg}}(k)} \qquad (16)$$

In general, the recommended coding rate can be determined by

$$\text{pr}_k = \begin{cases} \widehat{r}_k \frac{T_l(k)}{T_{\text{seg}}(k)}, \ \textit{if} \ T_l(k) > 0 \\ \widehat{r}_k \frac{T_b(k)+T_{\text{mac}}(k)-\widehat{T}_{\text{mac}}(k)}{T_{\text{seg}}(k)}, \ \text{otherwise} \end{cases} \qquad (17)$$

Obviously, the recommended coding rate may be not equal to the real coding rate of the segments stored at the media server. Thus, the client should determine which segment should be selected based on the recommended rate. In this paper, we will pick the representation with the most close coding rate to $\text{pr}_k$ from all the pre-encoded representations. The encoding information of the representation can be found in MPD.

Assume that there be $\varepsilon_k$ representations of the video sequence requested by client $k$ and let $\text{PR}_k$ indicate the set of those representations, $\text{PR}_k = \{\text{PR}_{k,1}, \cdots, \text{PR}_{k,\varepsilon_k}\}$ with $\text{PR}_{k,i} < \text{PR}_{k,i+1}$. If the recommended rate $\text{pr}_k$ is lower or higher than the coding rates of all the encoded representations, the DASH client will pick the representation with the lowest or the highest coding rate of $\text{PR}_k$. The principle of the final decision on the selected representation $\text{sp}_k$ can be summarized as follows:

$$\text{sp}_k = \begin{cases} \text{PR}_{k,1}, \ \text{if } \text{pr}_k < \text{PR}_{k,1} \\ \text{PR}_{k,i}, \ \text{if } \text{PR}_{k,i} \le \text{pr}_k < \text{PR}_{k,i+1} \\ \text{PR}_{k,\varepsilon_k}, \ \text{if } \text{pr}_k \ge \text{PR}_{k,\varepsilon_k} \end{cases} \qquad (18)$$

From Eq. (18), we can determine which segment should be requested. As long as the segment going to be requested is obtained, the unique URL of the segment can be constructed by using the metadata contained in MPD. Then, the request will be sent to the media server by HTTP GET command and the media server will pick the segment with the unique URL and send it to the DASH client.

## 3 Results and discussion

We provide our experimental results and discussion in this section. All the experiments are carried out in LTE system with the Network Simulation 3 (NS3) simulator [34]. eNodeB can obtain the CQIs through the feedback of all UEs. Each scheduling period called TTI (Transmission Time Interval) consists of two consecutive RBs and the time slot duration of each RB is 0.5 ms. Therefore, CQIs will be periodically updated and transmitted to eNodeB in the feedback information every 1 ms. It is assumed that the power be equally allocated among the RBs. Since we need the ACK information to predict the playback buffer status of the DASH client, the RLC mode is set to be AM (acknowledged mode). All the video sequence have been encoded into a series of the video segments with five different coding rates (presentations). Each segment consists of 60 frames with the time duration of 2 s. The detailed system configuration is shown in Table 5. The encoding information about the average coding rate, PSNR, and the corresponding presentation is shown in Table 6. For each segment, there is no need to receive all the video packets especially the packets with little effect on the received video quality. In our simulation, each video packet should be transmitted to the client before its deadline which depends on the maximum allowed waiting time of each video packet. We define $Ta_k$ as the maximum allowed waiting time for the packet of client $k$. It means that from the time the packet should be started to be played, the client only wait $Ta_k$ for the arrival of the video packet at most. In our simulation, $Ta_k = 5$ s, $1 \le k \le K$. Using this parameter, we can determine which packet has the chance to be transmitted and which should be discarded automatically. The clients can dynamically switch their request

**Table 5** Description of the experimental environment

| | |
|---|---|
| RLC mode | AM |
| TTI | 1 ms |
| The number of frames in one segment | 60 |
| The time duration of one segment | 2 s |
| The maximum allowed waiting time | 5 s |
| The number of the representations | 5 |
| The number of the clients | 6 |
| The number of RBs | 15 |
| Video sequences | Crew, soccer, coastguard, flower, foreman, bus |
| Resolutions | 352 × 288 |
| BER (bit error ratio) | 10% |

**Table 6** The encoding information of different video sequences

| Sequence | Client | | Presentations | | | | |
|---|---|---|---|---|---|---|---|
| | index | | 1 | 2 | 3 | 4 | 5 |
| Crew | 1 | Rate (kbps) | 466.7208 | 608.524 | 711.7496 | 801.0336 | 923.3712 |
| | | PSNR (dB) | 34.8681 | 35.5819 | 36.2521 | 36.8612 | 37.5994 |
| Soccer | 2 | Rate (kbps) | 423.2584 | 551.7096 | 636.772 | 718.7632 | 829.6016 |
| | | PSNR (dB) | 33.952 | 34.6657 | 35.2741 | 35.9167 | 36.6319 |
| Coastguard | 3 | Rate (kbps) | 471.1336 | 672.8216 | 810.1552 | 940.3784 | 1123.4392 |
| | | PSNR (dB) | 32.3916 | 33.1998 | 33.871 | 34.5482 | 35.3562 |
| Flower | 4 | Rate (kbps) | 999.0232 | 1153.0616 | 1337.796 | 1490.62 | 1699.604 |
| | | PSNR (dB) | 31.8602 | 32.851 | 33.692 | 34.5294 | 35.5316 |
| Foreman | 5 | Rate (kbps) | 235.1504 | 365.7448 | 419.1232 | 476.512 | 554.4792 |
| | | PSNR (dB) | 34.2313 | 34.9516 | 35.5223 | 36.1295 | 36.8365 |
| Bus | 6 | Rate (kbps) | 671.0488 | 940.0936 | 1090.0216 | 1226.1664 | 1411.5 |
| | | PSNR (dB) | 32.3691 | 33.2327 | 33.9662 | 34.6956 | 35.5563 |

presentations at the times that they send their segment requests.

"PROPOSED" is the algorithm which consists of both the proposed packet scheduling and resource management scheme. "AR" (adaptive request) only indicates the adaptive segment request scheme. When AR is not used at the client, the clients will send the segment request with a fixed time interval and the rate adaption scheme is to request the segment whose rate is the most closest to the estimated transmission rate. The algorithm proposed in this paper will be compared with the following four algorithms:

- *QTRMA* in [15]: It investigated QoE optimization approaches for adapting the adaptive HTTP video delivery. They considered the client buffer status and proposed a novel playback buffer-dependent approach that determines for each client the streaming rate for future video segments according to its buffer time and the achievable QoE under current radio conditions.
- *ADCLA* in [18]: It developed a packet importance model based on the information extracted from the compressed video and proposed a content-aware scheduling scheme for resource allocation. The waiting time in MAC queue is used to calculate the packet urgency.
- *MAXCI* in [35]: It is a purely channel-dependent scheduling scheme, in which the clients with good channel states always occupy the channel most of time even if they are well ahead of their video decoding deadlines.
- *PF* in [36]: It allocates the limited network resources by considering both the experienced channel states and the past client throughput. Its goal is to maximize the system throughput and to guarantee fairness among end clients.

In Fig. 4, we provide the average PSNR of each frame of all the clients for different algorithms respectively. From this figure, we can see that our proposed algorithm can outperform other existing algorithms for all the clients. This indicates that considering the packet importance and the packet urgency can make the packet scheduling and resource allocation strategy more reasonable. Due to taking into consideration the packet importance, ADCLA+AR has higher PSNR than MAXCI+AR and PF+AR. We can also observe that QTRMA has more stable performance than other algorithms. This is because for QTRMA, there is no packet loss, the client will wait until all the video packets of each segment arrive at the client.

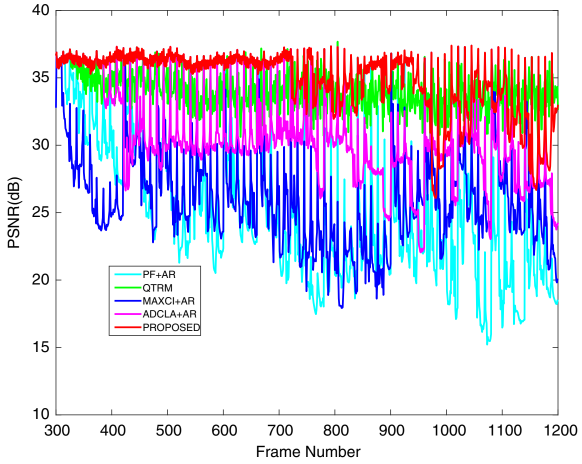

**Fig. 4** Average PSNR of all the clients for different algorithms

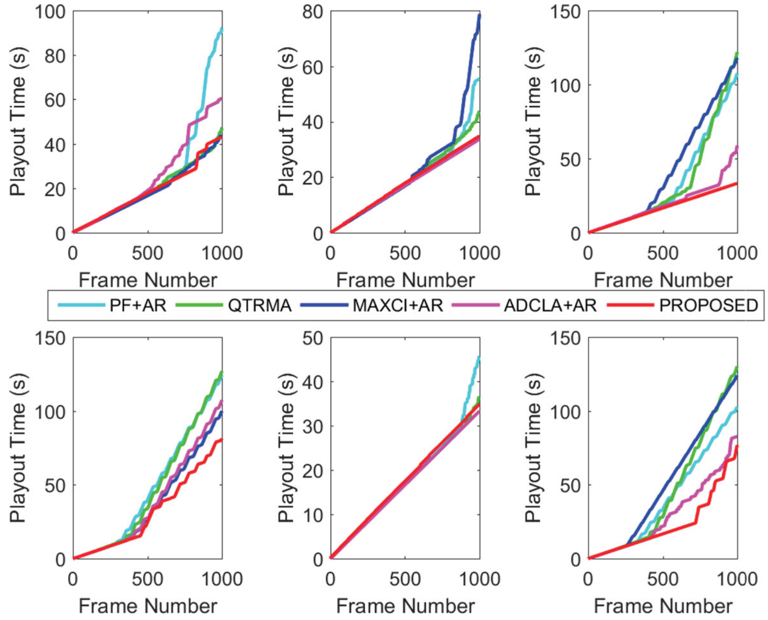

**Fig. 5** Playback time of each frame of each client for different algorithms

No packet loss means not so serious quality fluctuation during one segment. But transmitting all the video packets may lead to requesting the segment with low bit rate which will also degrade the value of the average PSNR. In fact, the packet with little importance should be discarded not scheduled.

Figures 5 and 6 shows the corresponding playback time of each frame of each client and the average playback time of each frame of all the clients. In the simulation, each client should experience a rebuffering time interval until all the packets of two segments are received. From the two figures, we can see that some continuous playback can be guaranteed for all the clients with different algorithms. From Fig. 6, we can deserve that the client can finish the whole video in a shortest time among all the clients by employing our proposed method. It means that the clients suffer the fewest rebuffering events and shortest interruption time which brings most satisfactory playback experience to the clients. From Fig. 5, we can

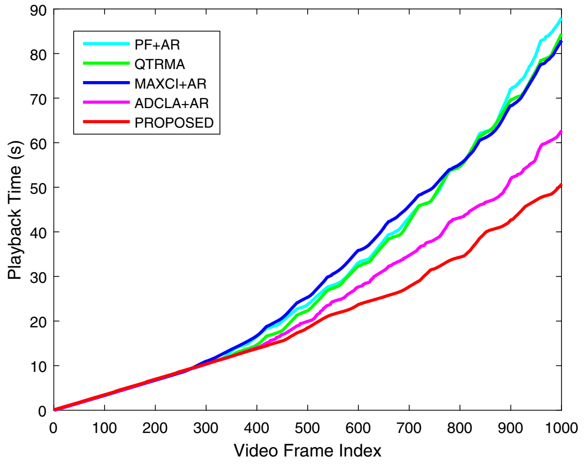

**Fig. 6** Average playback time of all the clients for each frame for different algorithms

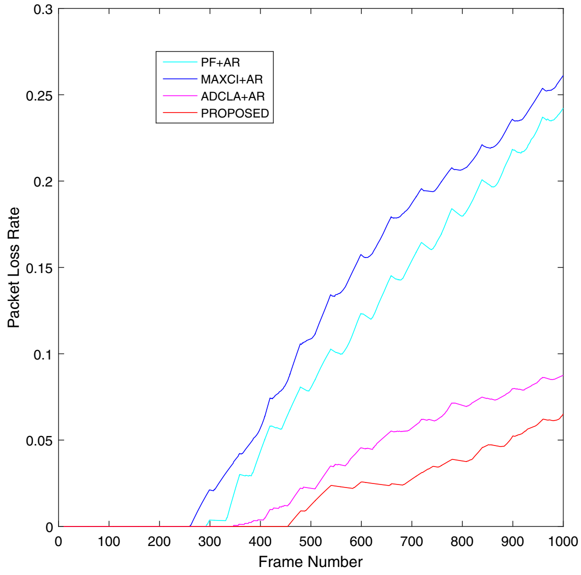

**Fig. 7** Average packet loss rate of all the clients for each frame for different algorithms

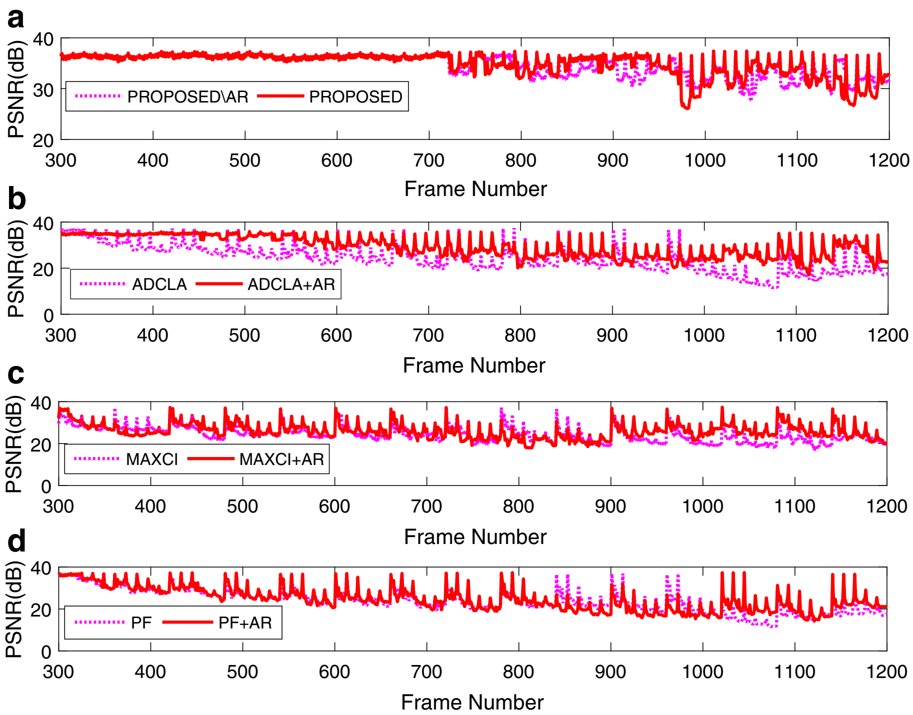

**Fig. 8** The comparison of the average PSNR of all the clients for each frame for different packet scheduling and resource management algorithms by employing our proposed segment request scheme AR or not. **a** Average PSNR of all the clients for each frame for PROPOSED vs. PROPOSED-AR. **b** Average PSNR of all the clients for each frame for ADCLA+AR vs. ADCLA. **c** Average PSNR of all the clients for each frame for MAXCI+AR vs. MAXCI. **d** Average PSNR of all the clients for each frame for PF+AR vs. PF

discover that the performance of our proposed method is not always the best among the algorithms for each frame. This is because the objective of our proposed algorithm is to maximize the sum of the utilities of all the clients not aiming at optimizing an individual client. We can also see that the algorithm QTRMA yields longer interruption time than the PROPOSED and ADCLA+AR. This is because QTRMA has to start to play until all the video packets of each segment arrive at the client while the former algorithms will wait for only $Ta_k$. It is unreasonable to wait for the arrival of the packets with little importance especially in a wireless network with a bad channel state.

To examine the performance of the packet scheduling strategy of our proposed scheme, we present the average packet loss rate of the clients for each frame in Fig. 7. From the figure, we can see that the packet loss rate can be kept to zero for some time because a rebuffering process exists for all the videos at the beginning of the playback. In addition, the packet loss rate can be restricted in a lowest range by using our proposed algorithm. The number of the frames with no packet loss is larger than other compared algorithms too. Lower packet loss is equal to the higher PSNR especially when we consider the packet utility in the packet scheduling and resource management. The packet with higher utility will be scheduled with higher priority. Reasonable packet scheduling strategy can improve the received video quality by allocating the limited resource to the clients who really need. Together with the results in Figs. 4 and 6, we can conclude that our proposed algorithm can acquire higher video quality as well as more continuous playback experience.

The simulation result of the average PSNR of all the clients for each frame for different packet scheduling and resource management algorithms integrating with our proposed segment request scheme AR is shown in Fig. 8, and the interruption time experienced by each client by employing our proposed segment request scheme AR is shown in Fig. 9. Note that “PROPOSED-AR” means the resource allocation scheme used is the algorithm proposed in this paper but the segment will be requested with a fixed time interval. Since QTRMA has its unique segment request scheme, we did not employ AR for it. From Fig. 8, we can see that the video quality based on AR is higher than that without AR for the most of the frames. This is because the segment adaption scheme can make the segment switch more flexible and adaptive to the varying channel states. Without rate adaptation, the rate of the requested segment may exceed the transmission rate that

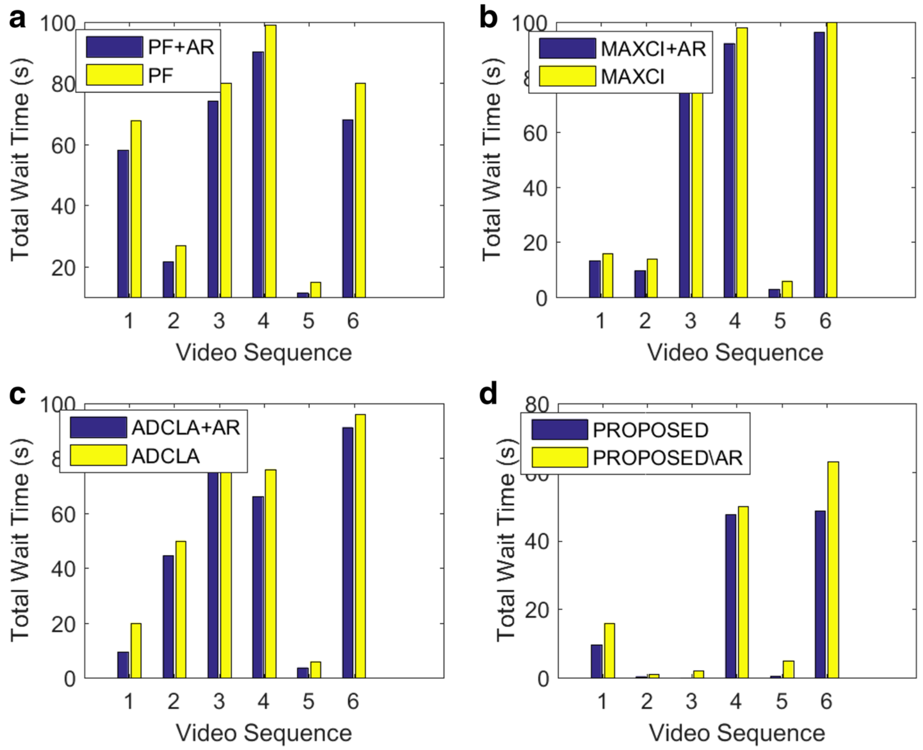

**Fig. 9** The comparison of interruption time experienced by each client for different packet scheduling and resource management algorithms by employing our proposed segment request scheme AR or not. **a** Interruption time experienced by each client for PROPOSED vs. PROPOSED-AR. **b** Interruption time experienced by each client for ADCLA+AR vs. ADCLA. **c** Interruption time experienced by each client for MAXCI+AR vs. MAXCI. **d** Interruption time experienced by each client for PF+AR vs. PF

the wireless channel can support. By using AR, the limited resource can be allocated to the packets which are suitable to be transmitted. As shown in Fig. 9, AR can bring the playback experience improvement apparently for all the packet scheduling and resource management schemes. Different schemes earn different achievements. In summary, QoE of the clients can be improved by employing AR.

## 4 Conclusions

A content and buffer status aware packet scheduling and resource management framework is proposed for DASH video streaming over LTE system. We first modify the transmission flow to make the whole framework tolerant of the loss of unimportant packets. Then, depending on the encoding information extracted from the encoder, we calculate the priorities of the video packets. Based on the feedback information from the clients we can estimate the packet urgency. The priorities of the video packets, the feedback channel states and the packet urgencies are considered to optimize the packet scheduling and RB assignment. Instead of requesting the video segments with fixed time interval, we propose an adaptive segment request time determination strategy which can effectively control the status of the client buffer and MAC queue so as to reduce the packet loss rate caused by the overflow of the client buffer and MAC queue. To get a better trade-off between the received video quality and playback continuity, we further propose a novel rate adaption algorithm which depends on the status of the client buffer, the MAC queue and the predicted transmission rate. Numerical simulation results show that our proposed client buffer and content aware packet scheduling framework can provide superior performance in terms of the PSNR of the received video and playback continuity.

**Funding**
This work is supported by the National Natural Science Foundation of China, No.61701389.

**Availability of data and materials**
Not applicable.

**Authors' contributions**
LH and FL conceived and designed the research. LH performed the experiments and analyzed the data. LH and FL wrote and edited the manuscript. Both authors read and approved the final manuscript.

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

### Competing interests

The authors declare that they have no competing interests.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 9 June 2017 Accepted: 27 October 2017
Published online: 14 November 2017

### References

1. O Oyman, S Singh, Quality of experience for HTTP adaptive streaming services. IEEE Commun. Mag. **50**(4), 20–27 (2012)
2. LD Cicco, S Mascolo, An adaptive video streaming control system: modeling, validation, and performance evaluation. IEEE/ACM Trans. Netw. **22**(2), 526–539 (2013)
3. Q Ai, P Wang, F Liu, Y Wang, F Yang, J Xu, QoS-guaranteed cross-layer resource allocation algorithm for multiclass services in downlink LTE system. Int. Conf. Wireless Commun. Signal Process, 1–4 (2010)
4. J Huang, W Lin, H Ko, in *IEEE Region 10 Conf*. A resource allocation algorithm for maximizing packet transmissions in downlink lte cellular systems, (2011), pp. 445–449
5. H Ramli, K Sandrasegaran, R Basukala, R Patachaianand, X Minjie, L Chung, Resource allocation technique for video streaming applications in the LTE system. Wirel. Opt. Commun. Conf. 1–5 (2010)
6. T Kupka, P Halvorsen, C Griwodz, An evaluation of live adaptive HTTP segment streaming request strategies. IEEE Conf. Local Comput. Netw, 604–612 (2011)
7. M Kalman, B Girod, P Beek, Optimized transcoding rate selection and packet scheduling for transmitting multiple video streams over a shared channel. IEEE Int. Conf. Image Process. **1**, 165–168 (2005)
8. J Xing, W Fan, Z Lu, A cross-layer scheduling scheme for video streaming based on fuzzy decision-making. IEEE Int. Conf. Commun. Technol, 84–87 (2011)
9. S Gouache, G Bichot, A Bsila, C Howson, Distributed and adaptive HTTP streaming. IEEE Int. Conf. Multimedia Expo, 1–6 (2011)
10. M Xing, S Xiang, L Cai, Rate adaptation strategy for video streaming over multiple wireless access networks. IEEE Global Commun. Conf, 5745–5750 (2012)
11. M Andrews, K Kumaran, K Ramanan, A Stolyar, P Whiting, R Vijayakumar, Providing quality of service over a shared wireless link. IEEE Commun. Mag. **39**(2), 150–154 (2001)
12. T Cong, H Le, H Nguyen, A Pham, R Man, An evaluation of bitrate adaptation methods for HTTP live streaming. IEEE J. Sel. Areas Commun. **32**(4), 693–705 (2014)
13. J Jiang, V Sekar, H Zhang, Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive. IEEE/ACM Trans. Netw. **22**(1), 326–340 (2012)
14. M Zhao, X Gong, J Liang, W Wang, X Que, S Cheng, QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP. IEEE Trans. Circ. Syst. Video Technol. **25**(3), 451–465 (2015)
15. A Essaili, D Schroeder, E Steinbach, D Staehle, M Shehada, QoE-based traffic and resource management for adaptive HTTP video delivery in LTE. IEEE Trans. Circ. Syst. Video Techn. **25**(6), 988–1001 (2015)
16. V Ramamurthi, O Oyman, Video-QoE aware radio resource allocation for HTTP adaptive streaming. IEEE Int. Conf. Commun, 1076–1081 (2014)
17. N Bouten, S Latre, J Famaey, In-network quality optimization for adaptive video streaming services. IEEE Trans. Multi. **16**(8), 2281–2293 (2014)
18. F Li, G Liu, L He, Application-driven cross-layer approaches to video transmission over downlink OFDMA networks. IEEE Globecom Workshops, 1–6 (2009)
19. M Zhao, X Gong, X Que, et al, Context-aware adaptive active queue management mechanism for improving video transmission over IEEE 802.11E WLAN. J. China Univ. Posts Telecommun. **19**(11), 65–72 (2012)
20. G Liebl, M Kalman, B Girod, Deadline-aware scheduling for wireless video streaming. IEEE Int. Conf. Multimedia Expo, 1–4 (2005)
21. R Seungwan, R Byunghan, S Hyunhwa, S Mooyong, Urgency and efficiency based packet scheduling algorithm for OFDMA wireless system. IEEE Int. Conf. Communications. **4**, 2779–2785 (2005)
22. A Dua, C Chan, N Bambos, J Apostolopoulos, Channel, deadline, and distortion ($CD^2$) aware scheduling for video streams over wireless. IEEE Trans. Wireless Commun. **9**(3), 1001–1011 (2010)
23. K Fujimoto, S Ata, M Murata, Playout control for streaming applications by statistical delay analysis. IEEE Int. Conf. Commun. **8**, 2337–2342 (2001)
24. S Wee, T tian, J Apostolopoulos, M Etoh, Optimized video streaming for networks with varying delay. IEEE Int. Conf. Multimedia Expo. **2**, 89–92 (2002)
25. C Boutremans, J Boudec, Adaptive joint playout buffer and FEC adjustment for internet telephony. IEEE Int. Conf. Comput. Commun. **1**, 652–662 (2003)
26. E Piri, M Uitto, J Vehkapera, T Sutinen, Dynamic cross-layer adaptation of scalable video in wireless networking. IEEE Global Commun. Conf, 1–5 (2010)
27. S Islam, M Hossain, A wireless video streaming system based on OFDMA with multi-layer H.264 coding and adaptive radio resource allocation. Int. Conf. Intell. Inf. Process, 1–6 (2011)
28. K Tappayuthpijarn, T Stockhammer, E Steinbach, HTTP-based scalable video streaming over mobile networks. IEEE Int. Conf. Image Process, 2193–2196 (2011)
29. R Radhakrishnan, A Nayak, Cross layer design for efficient video streaming over LTE using scalable video coding. IEEE Int. Conf. Commun, 6509–6513 (2012)
30. W Pu, Z Zou, C Chen, Video adaptation proxy for wireless dynamic adaptive streaming over HTTP. Packet Video Workshop, 65–70 (2012)
31. F Li, G Liu, L He, A cross-layer scheduling algorithm for H.264 video transmission over wireless networks. Int. Workshop Cross Layer Design, 1–6 (2009)
32. TS 36.213, Evolved Universal Terrestrial Radio Access (E-UTRA): physical layer procedures, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2427. Accessed 23 March 2017
33. L He, G Liu, Quality-driven cross-layer design for H.264/AVC video transmission over OFDMA system. IEEE Trans. Wirel. Commun. **13**(12), 6768–6782 (2014)
34. G Piro, L Grieco, G Boggia, F Capozzi, P Camarda, Simulating LTE cellular systems: an open-source framework. IEEE Trans. Veh. Technol. **60**(2), 498–513 (2011)
35. S Borst, User-level performance of channel-aware scheduling algorithms in wireless data networks. IEEE Int. Conf. Comput. Commun. **1**, 321–331 (2003)
36. A Jalali, R Padovani, R Pankaj, Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. IEEE Vehicular Technol. Conf. **3**, 1854–1858 (2000)