QoE Metric Based Resource Allocation for Dynamic Adaptive Streaming over HTTP in LTE Networks

Xiao Zhang, Anyue Wang, Danpu Liu
Beijing Laboratory of Advanced Information Network
Beijing Key Laboratory of Network System Architecture and Convergence
Beijing University of Posts and Telecommunications, Beijing, P.R. China, 100876
zhangxiao9207@163.com,moonwong.way@gmail.com,dpliu@bupt.edu.cn

ABSTRACT

Dynamic Adaptive Streaming over HTTP (DASH) is the main business mode of video service because of its convenient deployment, low cost, and adaptability to different user requirements. Meanwhile Quality of Experience (QoE) has become the main evaluation indicator of video service quality. Considering users are required to periodically feedback QoE metric to base station in the MPEG-DASH protocol, a Resource Allocation algorithm based on QoE Metric Feedback (QMFRA) for LTE networks is proposed in this paper. QMFRA algorithm aims to maximize the weighted sum of all users' data rates. The data rate reflects the user's channel quality, while the weight represents the influence of QoE metric. We consider buffer level, the occurrence of stalling and switch in the design of the weights, in order to avoid stalling as far as possible and enhance user fairness in resource scheduling. Simulation results show that QMFRA algorithm can effectively improve user's Mean Opinion Score (MOS) and reduce the occurrence of stalling, compared with the widely used Multi-Carrier Proportional Fair (MPF) scheduling.

CCS CONCEPTS

• **Networks → Network simulations**;

KEYWORDS

DASH, QoE, resource allocation algorithm, LTE

ACM Reference format:
Xiao Zhang, Anyue Wang, Danpu Liu. 2017. QoE Metric Based Resource Allocation for Dynamic Adaptive Streaming over HTTP in LTE Networks. In *Proceedings of MOBIMEDIA 2017, CHONGQING, R.R.CHINA, July 2017,* 5 pages.
DOI: 10.475/123_4

1 INTRODUCTION

With the vigorous development of mobile intelligent devices and deployment of 4^{th} generation mobile communication network, user demands on mobile video services grow rapidly. According to a traffic forecast published by Cisco, mobile video traffic will account for 80% of mobile traffic in 2020 [1].

MOBIMEDIA 2017, CHONGQING, R.R.CHINA

DOI: 10.475/123_4

DASH, Dynamic Adaptive Streaming over HTTP, is a new international standard of adaptive streaming. It splits video into segments and enables users to choose segment bitrate in accordance with network condition and personal preferences. DASH has become the most popular form of video service because of its low cost, convenient development, adaptability to different user requirements.

As a new evaluation index of video services, QoE (Quality of Experience) has drawn more attention because it concentrates on user's feeling of a service rather than network parameters. There are many factors that affect QoE, including initial delay, stalling, bitrate switching, video quality, video content and so on [2]. Among them, initial delay and stalling are the two factors that have the greatest influence on QoE [3].

Scheduling in wireless system aims to allocate limited bandwidth resources to users to meet their expectations [4]. It is a challenging task in cases where both users requirement on service quality and network traffic grow simultaneously. The QoE-oriented resource allocation has become a popular research field in the context of the increasing emphasis on QoE. In [5], a scheduling algorithm to maximize the Mean Opinion Score (MOS) is proposed. In [6], a resource scheduling algorithm combined with scalable coding is proposed. [7] proposed a QoE-oriented scheduling algorithm for video streaming, where buffered data feedback is introduced to maximize the utility function under the constrains of avoiding stalling. However, none of these studies quantified QoE, so there was no exact data to show that the proposed algorithm performed better on QoE. Moreover, those algorithms only consider the buffered data in scheduling, which is only one of the many influencing factors on QoE.

The MPEG-DASH standard specifies that users should periodically send QoE metric feedback to the base station. The QoE metric contains abundant information about QoE, such as buffered data, stalling conditions, switching conditions, and initial delay. If the information can be fully utilized in scheduling algorithm, there should be a better performance on QoE. Therefore, a QoE metric feedback based resource allocation (QMFRA) algorithm is proposed in this paper. QMFRA aims to maximize the weighted sum of all users' data rates, where the weight represents the influence of QoE metric, such as buffer level, the occurrence of stalling and switch. Simulation results show that the proposed algorithm can effectively improve user' MOS and reduce the occurrence of stalling.

The rest of the paper is organized as follows. Section II describes DASH service and QoE related knowledge in DASH, including QoE metric and the quantitative QoE model. Section III describes our QMFRA algorithm. Section IV shows simulation results and

MOBIMEDIA 2017, July 13-14, Chongqing, People's Republic of China

DOI 10.4108/eai.13-7-2017.2270548

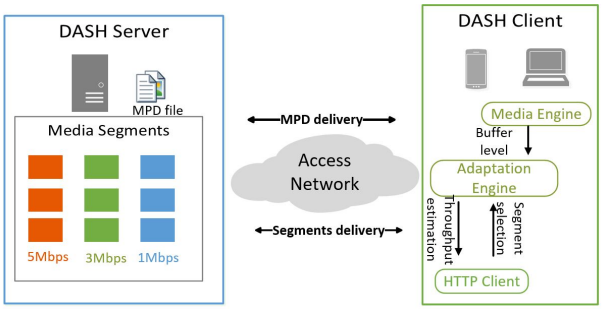

Figure 1: System model of DASH

compares performance of QMFRA and MPF. Section V concludes the paper.

2 APPLICATION SCENARIO

2.1 DASH Overview

Figure 1 depicts how DASH system works. DASH encodes video files with different bit rates and splits video into a sequence of 2-10 seconds long segments [8]. In general, there are more than one representation for each segment in the DASH server, and each representation corresponds to a set of specified encoding factors such as frame rate and resolution [9]. All information of the video is saved in a Media Presentation Description (MPD) file. A user will download the MPD file at the beginning of streaming and apply for a suitable representation of each segment. The selection of representations depends on a certain Rate Adaptation Algorithm (RAA). The client begins to play the video when it has enough buffered data. If the buffered data drops under a threshold, the client will stop playing and continue buffering until there are enough data to resume playing. The process will carry on in a loop during the whole playing period.

At user side, the client will apply for a certain bit rate segment according to RAA. There are basically three types of RAA. The first type is based on the available network capacity prediction, the second one is based on the buffer, and the third one is the combination of both. For simplicity and efficiency, this paper uses a buffer based RAA proposed by [10]. The selected rate is linear dependent to the amount of buffered data at client. This RAA does not require additional cost of monitoring and predicting network capacity. Moreover, the selected rate is continuous, avoiding significant drop of bit rate in adjacent segments, which may notably harm QoE [2].

2.2 QoE for DASH

QoE has become the prime performance criterion of DASH service because DASH provides very flexible service and QoE always concentrates on users' feeling. MPEG-DASH, the international standard of DASH, has foreseen this trend and defines QoE metric in the standard for future optimization of QoE. As shown in table 1, QoE metric contains many QoE-related elements. MPEG-DASH specifies that users may periodically feedback QoE metric to the base station. Feedback is optional, but the user must send the whole QoE metric if feedback is chosen [8]. QoE describes users' perceived

Table 1: QoE Metric Defined by MPEG-DASH

Elements	Discription
HTTP Request/Respond Transactions	Lists of HTTP request/respond transactions, including URL, request/respond time etc.
Representation Switch Events	Lists of representation switch events, including time of the switch event, representation id identifying the switch-to representation etc.
Initial Playout Delay	The initial playout delay is measured as the time in milliseconds from the fetch of the first media segment and the time at which media is retrieved from the client buffer.
Buffer Level	Lists of buffer occupancy level measurements during playout at normal speed, including time of measurement and level of buffer in milliseconds.
Play List	Lists of playback periods. A playback period is the time interval between a user action and whichever occurs soonest of the next user action, the end of playback or a failure that stops playback. It includes timestamp of the start of a playback, type of user action which triggered playback, stop reason, duration of playback, representation id, duration etc.

satisfaction of video service. As a subjective indicator, there are many factors that may impact QoE. To evaluate the QoE performance of a video service and related resource allocation algorithm, a quantitative model is needed to quantify QoE. We employ the model proposed in [11] , which is the most comprehensive one to the extent known of the author. The model is specially derived and validated for DASH video. The model's proposer did sufficient experiments, collected lots of data, derived and trained the model strictly. They investigated three factors which impact QoE: initial delay, stalling and bit rate fluctuation. For each factor, they explored multiple dimensions that may have different effects on QoE. Extensive subjective tests were conducted in which a group of subjects provided subjective evaluation while watching DASH video with one or more artifacts occurring [11]. Based on the tests, they first derived impairment functions to quantify the impairment of three factors, and then combined three impairment functions to an overall QoE model for DASH video.The model can be described as:

$$DASH - MOS = 1 + 0.035R + 7 \times 10^{-6} R(R-60)(100-R) \quad (1)$$

$$R = 100 - I_{ST} - I_{ID} - I_{LV} + C_1 * I_{ID}\sqrt{I_{ST} + I_{LV}} + C_2 * I_{ID}\sqrt{I_{ST} * I_{LV}} \quad (2)$$

where R is a transmission rating factor defined by three impairment functions. I_{ID} is the impairment function of initial delay. I_{ST} is the impairment function of stalling. I_{LV} is the impairment function of level variation. C_1 and C_2 are coefficients. Detailed derivation and validation process can be found in [11].

The model outputs MOS value within the range of [1,4.5]. Tests data showed that the correlation coefficient between the MOS obtained by the model and the MOS obtained by users directly is 0.91,

which means a high reliability. Therefore, we will use this model in following simulation to evaluate resource allocation algorithms' performance on QoE.

3 QMFRA SCHEME

In LTE networks, resource block (RB) is the scheduling unit of resources. For simplicity, we consider the scenario in only one cell. Suppose there are $\mathbf{K} = \{1, \cdots, K\}$ users and $\mathbf{N} = \{1, \cdots, N\}$ RBs in the cell. In each scheduling slot, the resource allocation algorithm determines how to allocate the N RBs among the K users. From classic resource allocation algorithms, such as round robin, proportional fair (PF) and Max C/I, we can infer that typical resource allocation algorithms in wireless networks seeks to optimize a utility function in the case of limited bandwidth. Multicarrier proportional fair (MPF) algorithm reaches a balance between efficiency and fairness, therefore it becomes the most popular scheduling algorithm in the past decades. However, MPF is not QoE-friendly. It cares about the ratio of user's instantaneous transmittable rate to average rate, which has no relevance to QoE. [12] proposed a QoE-aware resource allocation algorithm. It needs a network monitor or proxy to obtain application layer parameters as input of scheduling. However, the relevance between the application layer parameters and QoE is not very clear. [7] modified the utility function of PF algorithm with buffer level feedback and constrained stalling probability in scheduling. But it only considered buffer level and discarded other useful information in QoE metric. As an authoritative foresight research of QoE for DASH, [2] has indicated that future DASH solution should comprehensively consider the factors that may impact QoE. Therefore, we raise the idea of fully using the information in QoE metric to design a resource allocation algorithm to optimize user QoE. QoE metric is a whole and feedback of QoE metric is already standardized in MPEG-DASH, making it very simple and practical. Unlike [7], the proposed QoE metric feedback based resource allocation (QMFRA) algorithm not only considers buffer level but also takes switching and stalling into account, which provides a more comprehensive reflection of QoE in scheduling and can enhance fairness among users directly.

According to [2], to optimize the user's QoE we need to come out a resource allocation algorithm that can:

(1) reduce the number and duration of stalling
(2) improve average bitrate of segments
(3) reduce obvious bitrate fluctuation between adjacent segments
(4) consider fairness among users

For (1), in order to reduce stalling, users with less buffered data should have higher priority in resource allocation. We can also take user's status of stalling into account, which indicates higher priority to users with more stalling. For (2), improving average bitrate is equivalent to improving throughput of the network. Therefore, the utility function should also consider user's channel quality which can be represented by transmittable rate R_k. Users in poor channel quality will probably lose data even though it gets scheduled, which is a waste of resource. For (3), the rate adaptation algorithm described in the previous section has avoided this situation. The algorithm chooses a certain video rate from several discrete candidates based on buffered data amount. The distance between two adjacent video rates provides a natural cushion to absorb rate oscillation [10]. For (4), if the transmittable rate represents efficiency, then there should be a fairness factor in utility function. Providing users with more stallings a higher priority is the reflection of fairness. Meanwhile, users that frequently switch down should also have higher priority for the sake of fairness.

All information considered above can be derived from the QoE metric. As can be seen from Table 1, the amount of buffered data can be obtained from Buffer Level. The details of switching can be obtained from the Representation Switch Events. From a simple calculation on Play List, we can derive the stalling number and total stalling duration.

Based on above considerations, we design the utility function as the sum of all users' weighted rate. The resource allocation problem can be formulated as:

$$\begin{aligned} &\max_{\mu} \sum_{k=1}^{K} \omega_k R_k \\ s.t. \quad &\mu_{k,n} \in 0, 1 \\ &\sum_{k=1}^{K} \mu_{k,n} \leq 1, \forall n \end{aligned} \tag{3}$$

where R_k is the instantaneous transmittable rate of user k, and reflects the channel quality of user k. R_k can be further defined as:

$$R_k = \sum_{k=1}^{K} \mu_{k,n} r_{k,n} \tag{4}$$

where $r_{k,n}$ is the achievable rate of user k at RB n and can be calculated by Shannon formula. $\mu_{k,n}$ is the decision factor, $\mu_{k,n} = 1$ if RB n is allocated to user k, otherwise $\mu_{k,n} = 0$. The following conditions on $\mu_{k,n}$ ensure that one RB can be allocated to only one user in each scheduling slot:

$$\begin{aligned} &\mu_{k,n} \in 0, 1 \\ &\sum_{k=1}^{K} \mu_{k,n} \leq 1, \forall n \end{aligned} \tag{5}$$

ω_k is the priority weight assigned to user k in scheduling slot t and is dynamically changed in each scheduling slot. It shows how QoE metric reacts in scheduling. The specific design goes as follows:

$$\omega_k = \frac{t_k S_k}{(B_k + a)} \tag{6}$$

where t_k is the stalling factor, and defined as

$$t_k = \frac{SD_k}{SD_{total}} \tag{7}$$

where SD_k represents accumulated stalling duration of user k, and SD_{total} represents accumulated stalling duration of all users. Greater stalling factor, which indicates the user has encountered a relatively long time of stalling, leads to higher priority weight. Since stalling is the most harmful event to QoE, giving such users a higher priority may avoid continual deterioration on QoE.

$B_k + a$ is the buffer factor, where B_k is the sum of duration of buffered data of user k in seconds, and the constant parameter a is introduced to prevent zero in denominator. Users with less buffered

data are more vulnerable to stalling. Therefore, they are granted higher priority weight.

S_k is the switching factor and is expressed as:

$$S_k = \begin{cases} \frac{1}{b} & b > \frac{N}{2} \\ -b & -b > \frac{N}{2} \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

We will count the switching times of user k in the past N segments from current scheduling slot as b. Switching up denotes as 1 and switching down denotes as -1. When switching up occurs frequently ($b > N/2$), S_k indicates decreasing priority weight. When frequent switching down occurs frequently ($-b > N/2$), S_k indicates increasing priority weight. Because frequent switching up shows good channel quality, abundant buffered data and a guaranteed QoE. Nevertheless users that switch down frequently are going through deterioration in channel quality, shrink in buffered data and decrease in video rate in segments selection. Based on fairness considerations, the former users are granted lower priority weight and the later users are granted relatively higher priority weight. The calculation of S_k is the cumulative value of the past N segments, which can dynamically reflect the user's channel quality of the passing period.

We solve problem (3) by the following procedure. It's necessary to point out that in the following simulation we solve MPF scheduling with the same method. The complexity of original MPF is $O(K^N)$, which is unpractical when K and N reach the normal scale in real mobile network system. The following method gives a non-optimal but simplified procedure with the complexity of $O(KN)$.

Input: QoE metric feedback, CQI feedback
Output: RB allocation set: $C^{(n)}$
$C^{(n)} = \varnothing$;
for *k=1:K* **do**
  for *n=1:N* **do**
    $\mu = \begin{cases} 1, \mu^* = \underset{\mu}{argmax}\, \omega_k R_k^{\mu} \\ 0, otherwise \end{cases}$
  end
  update $C^{(n)}$
end
return $C^{(n)}$

Algorithm 1: QMFRA

4 PERFORMANCE EVALUATION

In this section, we evaluate the performance of QMFRA algorithm. We convey simulation on an open source LTE system level simulator developed by University of Vienna and compare QMFRA with the widely used MPF scheduling algorithm.

We use the QoE model described in section II to quantify user's QoE as the prime indicator to evaluate performance of the scheduling algorithm. From simulation we can obtain initial delay, stalling and switching data. Applying those data to the QoE model we can obtain user's MOS. In addition, according to [3], stalling is the most influential factor to QoE. Many QoE-oriented researches also use stalling as an indicator of QoE. This paper also compares the performance on stalling of QMFRA and MPF.

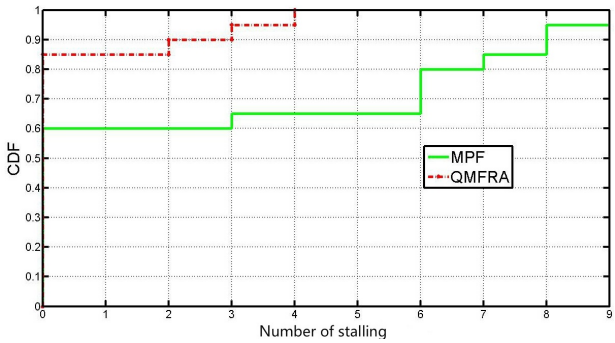

Figure 2: CDF of number of stalling

Figure 1 shows the cumulative distribution function of number of stalling of the two algorithms. The green solid line represents the MPF algorithm and the red dotted line represents the QMFRA algorithm. It can be seen that in the 100 seconds simulation, 60% of the users using MPF have no stalling, 5% have 3 stalling, 15% have 6, 5% have 7, 10% have 8. The maximum number of stalling is 9 and 5% of users reach it. The performance of QMFRA in the number of stalling has significantly improved. 85% of users have no stalling, 5% have 2 stalling, another 5% have 3 stalling. There are also 5% users that reach the maximum number of stalling, i.e. 5. The average number of stalling of MPF users is 2.65. The same index for QMFRA dramatically drops to 0.45.

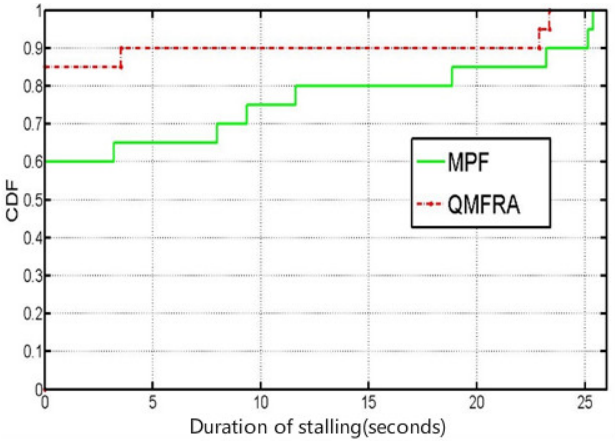

Figure 3: CDF of duration of stalling

Figure 2 shows the cumulative distribution function curves of stalling duration of the two algorithms. QMFRA also significantly performs better. The average duration of stalling for MPF users is 6.24 seconds. The same index for QMFRA is 2.49 seconds. The maximum stalling duration of QMFRA is also less than MPF.

Table 2: Comparison of several average index

Indicator	MPF	QMFRA
Average Stalling Number	2.65	0.45
Average Stalling Durarion	6.24 seconds	2.49 seconds
Average initial delay	2.38 seconds	2.55 seconds
Average video bitrate	340kbps	238kbps
Network throughput	6.792Mbps	4.754Mbps
Average MOS	2.41	2.87

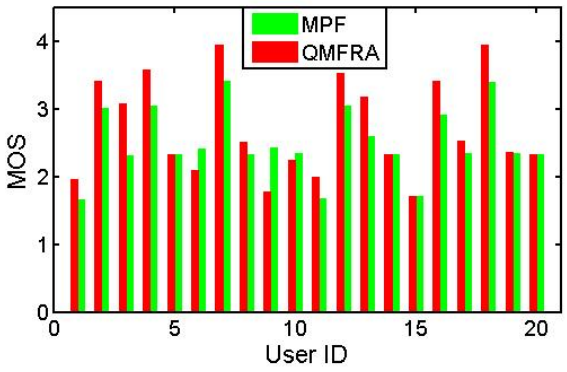

Figure 4: Comparison of user MOS

Figure 3 is a comparison of the user MOS obtained by the two algorithms. Among all the QMFRA users, 55% have obviously higher MOS than MPF and 30% have very close MOS with MPF. The average MOS of QMFRA users is 2.87, 19% higher than MPF's mean MOS 2.41. It testifies that QMFRA can bring better performance on QoE. The result is consistent with the previous discussion. The design of QMFRA algorithm fully utilizes the information in QoE metric and aims to avoid harmful events for QoE such as stalling and switching down frequently. Simulation results of MOS shows QMFRA can effectively improve user's QoE and realize a QoE oriented resource allocation, which is the origin goal of this paper.

Table 2 lists the comparison of several average indicators. We can see that QMFRA significantly outperforms MPF in average stalling number and average staling duration. In average initial delay, the difference is not obvious. Because QMFRA introduces QoE metric feedback to avoid stalling and improve fairness, network throughput of QMFRA is reduced by 2.038 Mbps. Therefore, the rate of the applied video segments is reduced by 100 kbps approximately. However, in the most critical indicator MOS, QMFRA performs better. User's average MOS increases by 0.46. It shows that the idea of this paper is reasonable, using QoE metric in resource allocation can effectively enhance user QoE.

5 CONCLUSIONS

In this paper, the idea of using QoE metric specified by MPEG-DASH in resource allocation is proposed. Based on the idea, we proposed QMFRA algorithm. QMFRA aims to maximize all users' weighted rate. The design of weight introduces buffer factor, stalling factor and switching factor from QoE metric. Simulation results show that compared with the widely used MPF algorithm, QMFRA algorithm can significantly reduce stalling number and stalling duration and improve the user's MOS. This also means that the QMFRA algorithm can enhance user QoE.

6 ACKNOWLEDGMENT

This work was funded by the NSFC under Grant 61271257, 61171107, and supported by "the Fundamental Research Funds for the Central Universities".

REFERENCES

[1] Cisco. *White paper: Cisco VNI Forecast and Methodology, 2015-2020*, Jun 2016.
[2] Slanina M Seufert M, Egger S. A survey on quality of experience of http adaptive streaming. *IEEE Communications Surveys and Tutorials*, 17(1):469–492, 2015.
[3] Sieber C Hossfeld T, Seufert M. Assessing effect sizes of influence factors towards a qoe model for http adaptive streaming. pages 111–116, 2014.
[4] Baker M. Sesia S, Toufik I. *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.
[5] Steinbach E Thakolsri S, Khan S. Qoe-driven cross-layer optimization for high speed downlink packet access. *Journal of Communications*, 4(9), 2009.
[6] Liang J Zhao M, Gong X. Qoe-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over http. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(3):451–465, 2015.
[7] Oyman O Ramamurthi V. Video-qoe aware radio resource allocation for http adaptive streaming. pages 1076–1081, 2014.
[8] Sodagar I. The mpeg-dash standard for multimedia streaming over the internet. *IEEE Multimedia*, 18(4):62–67, 2011.
[9] Veciana G D Joseph V. Nova: Qoe-driven optimization of dash-based video delivery in networks. *Computer Science*, pages 82–90, 2013.
[10] Mckeown N Huang T Y, Johari R. A buffer-based approach to rate adaptation: evidence from a large video streaming service. *Acm Sigcomm Computer Communication Review*, 44(4):187–198, 2014.
[11] Ulupinar F Liu Y, Dey S. Deriving and validating user experience model for dash video streaming. *IEEE Transactions on Broadcasting*, 61(4):651–665, 2015.
[12] Volk M Rugelj M, Sedlar U. Novel cross-layer qoe-aware radio resource allocation algorithms in multiuser ofdma systems. *IEEE Transactions on Communications*, 62(9):3196–3208, 2014.