Caching Policy Optimization for Video on Demand

Hao Wang, Shengqian Han, and Chenyang Yang
School of Electronics and Information Engineering, Beihang University, China, 100191
Email: {wang.hao, sqhan, cyyang}@buaa.edu.cn

***Abstract*—Caching popular contents at the base stations is an effective way to address the challenging data traffic demand driven by video-on-demand (VoD) service. Initial delay is one of most important factors affecting the quality of experience (QoE) of users for VoD service. Existing work has optimized the caching policy to reduce the average delay for file downloading service. However, considering the fact that the QoE for VoD service does not linearly decrease with the initial delay, the caching policy minimizing the average delay is not optimal for VoD service. In this paper we optimize the caching policy to control the initial delay, aimed at maximizing the average QoE of a user with random requests of videos from a file library. By properly approximating the non-smooth non-concave QoE function, we obtain efficient caching policies for VoD service, based on which the impact of file popularity on the caching policy is analyzed. Simulation results demonstrate the advantages of the proposed caching policy.**

I. Introduction

The telecommunication industry has been witnessing an explosion of wireless data traffic driven by video-on-demand (VoD) service [1]. To meet the challenging traffic demands and guarantee the quality of experience (QoE) of users, ultra dense network has emerged as an candidate solution, where small base stations (SBS) are deployed to bring the network close to the users. However, a drawback of this technology is the requirement of high-capacity and low-latency backhaul for each SBS, which leads to prohibitive cost with the increase of the density of SBSs. To overcome this dilemma, a concept of femtocaching was proposed in [2] to use the cheap storage to replace the expensive backhaul. By caching the popular video files at the SBSs without backhaul link (also referred to as "helpers") and streaming the non-cached files from the macro BS, femtocaching is able to dramatically improve the network throughput.

Caching at the wireless edge has been extensively studied in the literature, e.g., [3], [4], [5] and references therein. Nevertheless, most existing caching policies are designed for file downloading service, while the optimization of caching policy for VoD service aimed at improving the QoE of users by taking into account the features of VoD has received little attention. Considering the adaptation of video rate, the Least Recently Used (LRU) based reactive caching policy was studied in [6] and [7], and the proactive caching policy for multi-rate VoD was investigated for a single video file in [8] and for multiple video files in [9] and [10]. Except for video rate adaptation, initial delay is another important factor affecting the QoE of VoD service. Existing works have studied the optimal caching policy aimed at minimizing the average delay for file downloading service [2], [11], and showed that caching the most popular contents at the helper is the optimal policy in the single-helper scenario.

However, the minimization of average delay is not equivalent to the maximization of QoE for VoD service, because the QoE has been shown as a non-linear function of initial delay [12], [13]. A simple example to show the non-optimality of the existing policy designed for file downloading is given as follows, where the feature of VoD service allowing a certain initial delay is considered. Let us consider the caching of three video files, f_1, f_2, f_3, with increasing popularity. Suppose that the three videos have decreasing file sizes so that the helper with limited storage size can either only cache f_1 or cache both f_2 and f_3. Assume that the three videos respectively require to pre-buffer $2B, B, B$ bits during the initial phase, the downloading rates from a macro BS and the helper are B and $2B$ bit/s, respectively, and the maximal acceptable initial delay is 1 second, within which the QoE of users does not degrade. Then, according to the optimal caching policy for file downloading, the two most popular videos, i.e., f_2 and f_3, should be cached, which leads to the initial delays for the three files as 2, 0.5 and 0.5 seconds. Such caching policy reduces the QoE of users requesting f_1 because its initial delay exceeds the maximal acceptable initial dealy. An alternative caching policy is to only cache f_1, resulting in the initial delay of 1, 1 and 1 second for the three files, which does not lead to any QoE degradation.

In this paper we optimize the caching policy by taking into account the impact of initial delay on the QoE of VoD service. Considering that the QoE is a non-smooth non-concave function of the initial delay, which is difficult to optimize, we first propose an approximate QoE function, based on which efficient caching policies are obtained and the impact of file popularity on the caching policies is analyzed. Simulation results show that the proposed caching policies can effectively improve the QoE of users compared to the existing policy.

II. System Model

Consider a femtocaching system where a macro BS and a helper serve a single user. The macro BS is assumed to have ideal backhaul, connected to the core network to gather the requested video files without delay. Each helper has no backhaul but is equipped with a cache with the storage size of C bits. Suppose that the user requests the video from a library of F files. Let q_f, R_f, and B_f denote the popularity (i.e., request probability), playback rate, and size of the f-th video file, respectively, $f = 1, \ldots, F$. Under the constraint on storage size of the helper, partial caching is allowed in the

paper, i.e., the helper may only cache a part of a video file. This means that a user may stream the cached part of a video from the helper and the uncached part from the macro BS via backhaul, respectively. Let x_f denote the cached portion of the f-th video file. Then, the caching policy is to determine how much portion of which files should be cached at the helper.

A video file is composed of a sequence of chunks, each of which is encoded and decoded as a stand-alone unit. The duration of a chunk is generally much longer than the coherent time of the small-scale fading channel, i.e., the channel coding can span across a large number of independent fading channels [14]. Therefore, it is safe to assume that the ergodic capacity is achievable. Let $R_{\rm B}$ and $R_{\rm H}$ denote the ergodic capacity from the macro BS and the helper to the user, which are referred to as downloading rates. Moreover, consider $R_{\rm B} \leq R_{\rm H}$ since the helper is usually closer to the user and can allocate a larger bandwidth to the user compared to the macro BS by noting that the number of users accessing to the helper is smaller.

Regarding the impact of initial delay on QoE, the logarithmic relationship between them has been validated by experiment for file downloading service in [15]. Moreover, it is shown in [12] that there exists a threshold for initial delay blow which the QoE does not degrade. In [13], the impact of page loading delay on the QoE of web browsing service is characterized by the following function

$$\eta(\tau) = \begin{cases} \alpha_0, & 0 \leq \tau < \tau_0 \\ \alpha_1 - \kappa \log(\tau), & \tau \leq \tau_0, \end{cases} \tag{1}$$

where the constants α_0, α_1, and κ satisfy $\alpha_1 - \kappa \log(\tau_0) = \alpha_0$ to ensure the function continuous. We can see that the web browsing service allows a delay less than τ_0 without degrading the QoE, beyond which the QoE degrades in a logarithmic law. According to the analysis in [13], such a QoE function coincides with the Weber-Fechner Law, a key principle in psychophysics describing the general relationship between the magnitude of a physical stimulus and its perceived intensity within the human sensory system, and therefore can be used to evaluate the impact of the initial delay on the QoE of VoD service.

III. Delay-Aware Video Caching

In this section we optimize the caching policy by taking into account the impact of the initial delay of VoD service.

A. Initial Delay

The initial delay, τ_f, for the user requesting the f-th video file is defined as the minimal pre-buffer duration after which the user can play the whole video without stalling. This requires that at any moment t after the user starts to play the video, the cumulative bits received by the user should be no less than the cumulative bits required by displaying the video. It is obvious that the initial delay τ_f depends on how much portion of the f-th video is cached, i.e., x_f. We next show that τ_f is also affected by which portion of the video is cached given x_f. To see this, we consider two special cases, where the helper caches the first or final x_f portion of the f-th video, respectively, which are denoted by "Cache-first" and "Cache-final" for notational simplicity.

1) Cache-final: In this case, the user streams the first $1-x_f$ portion of the f-th video from the macro BS, and streams the remaining x_f portion from the helper. Let $\tau_{{\rm B},f}$ and $\tau_{{\rm H},f}$ denote the pre-buffer durations from the macro BS and the helper, respectively.

In order to ensure non-stalling for the first portion of the f-th video lasting $\frac{B_f(1-x_f)}{R_f}$ seconds, which is streamed from the macro BS, the following condition needs to be satisfied for all $t \in [0, \frac{B_f(1-x_f)}{R_f}]$:

$$\min\left(B_f(1-x_f), R_{\rm B}\tau_{{\rm B},f} + R_{\rm B}t\right) \geq R_f t. \tag{2}$$

From (2), we can find that the pre-buffer duration $\tau_{{\rm B},f}$ should be long enough so that (2) is satisfied when $t = \frac{B_f(1-x_f)}{R_f}$. Further considering that $\tau_{{\rm B},f} \geq 0$ and the pre-buffered data $R_{\rm B}\tau_{{\rm B},f}$ should be no more than all data included in the first portion of the video, we can obtain that $\tau_{{\rm B},f}$ is bounded by

$$\left(\frac{B_f}{R_{\rm B}} - \frac{B_f}{R_f}\right)^+ (1-x_f) \leq \tau_{{\rm B},f} \leq \frac{B_f(1-x_f)}{R_{\rm B}}, \tag{3}$$

where $(x)^+ \triangleq \max(x, 0)$. Moreover, the time required to download the non-pre-buffered data of the first portion can be computed from (2) as

$$T_f = \frac{B_f(1-x_f)}{R_{\rm B}} - \tau_{{\rm B},f}. \tag{4}$$

When the first portion has been downloaded, the user can start to download the second portion from the helper. To ensure non-stalling during streaming the second portion, the following condition needs to be satisfied for all $t \in [\frac{B_f(1-x_f)}{R_f}, \frac{B_f}{R_f}]$:

$$\begin{aligned} &R_{\rm B}\tau_{{\rm B},f} + R_{\rm H}\tau_{{\rm H},f} + R_{\rm B}T_f + R_{\rm H}(t - T_f) \\ &= \frac{(R_{\rm B}-R_{\rm H})B_f(1-x_f)}{R_{\rm B}} + R_{\rm H}(\tau_{{\rm B},f}+\tau_{{\rm H},f}) + R_{\rm H}t \geq R_f t, \end{aligned} \tag{5}$$

where the equality follows upon substituting (4).

From (5), we can obtain that the pre-buffer duration $\tau_{{\rm B},f} + \tau_{{\rm H},f}$ should be long enough so that (5) is satisfied when $t = \frac{B_f}{R_f}$, which leads to

$$\tau_{{\rm B},f} + \tau_{{\rm H},f} \geq \left(\frac{B_f}{R_{\rm B}} - \frac{B_f}{R_f}\right)(1-x_f) + \left(\frac{B_f}{R_{\rm H}} - \frac{B_f}{R_f}\right)x_f. \tag{6}$$

Further considering (3) and $\tau_{{\rm H},f} \geq 0$, after some manipulations, we can obtain the initial delay τ_f as

$$\begin{aligned} \tau_f &= \min\ \tau_{{\rm B},f} + \tau_{{\rm H},f} \\ &= \max\left(\left(\frac{B_f}{R_{\rm B}} - \frac{B_f}{R_f}\right)(1-x_f) + \left(\frac{B_f}{R_{\rm H}} - \frac{B_f}{R_f}\right)x_f, \right. \\ &\qquad \left. \left(\frac{B_f}{R_{\rm B}} - \frac{B_f}{R_f}\right)^+ (1-x_f)\right) \\ &\triangleq \max\left(\Delta_1 + \Delta_2, (\Delta_1)^+\right). \end{aligned} \tag{7}$$

Noting $R_{\mathrm{B}} \leq R_{\mathrm{H}}$ so that $\Delta_2 > 0$ leads to $\Delta_1 > 0$, we can obtain τ_f from (7) as

$$\tau_f = \begin{cases} (\Delta_1)^+, & \Delta_2 < 0 \\ \Delta_1 + \Delta_2, & \Delta_2 > 0 \text{ and hence } \Delta_1 > 0, \end{cases} \tag{8}$$

which can be rewritten in a compact form as

$$\begin{aligned} \tau_f =& (\Delta_1)^+ + (\Delta_2)^+ \\ =& \left(\frac{B_f}{R_{\mathrm{B}}} - \frac{B_f}{R_f}\right)^+ (1-x_f) + \left(\frac{B_f}{R_{\mathrm{H}}} - \frac{B_f}{R_f}\right)^+ x_f. \end{aligned} \tag{9}$$

2) Cache-first: In this case, the user streams the first portion from the helper and the second portion from the macro BS. Similar to the cache-final case, we can obtain the initial delay as

$$\begin{aligned} \tau_f =& \max\left(\Delta_1 + \Delta_2, (\Delta_2)^+\right) \\ =& \begin{cases} (\Delta_1 + \Delta_2)^+, & \Delta_2 < 0 \\ \Delta_1 + \Delta_2, & \Delta_2 > 0 \text{ and hence } \Delta_1 > 0, \end{cases} \end{aligned} \tag{10}$$

which can be rewritten as

$$\tau_f = (\Delta_1 + \Delta_2)^+ = \left(\frac{B_f x_f}{R_{\mathrm{H}}} + \frac{B_f(1-x_f)}{R_{\mathrm{B}}} - \frac{B_f}{R_f}\right)^+. \tag{11}$$

We next discuss the initial delay in general cases where the helper arbitrarily selects x_f portion of the f-th video to cache.

First, comparing (9) and (11), we can find that the cache-first case needs shorter initial delay than the cache-final case if $\Delta_1 > 0$ and $\Delta_2 < 0$, i.e., $R_{\mathrm{B}} < R_f < R_{\mathrm{H}}$. In this scenario, for the cached portion the downloading time is shorter than the playback time, so that the user can use the time difference to buffer the not-yet-played uncached portion. This reduces the amount of data that are required to be pre-buffered and hence leads to the reduction of initial delay. Meanwhile, the benefits of exploiting the high downloading rate from helper depends on how much uncached data remain, e.g., the cache-final case benefits nothing because there are no data left to stream within the above mentioned time difference. This suggests that caching earlier portion of a video can benefit more. Thus, the cache-first and cache-final cases provide the lower and upper bounds for the initial delay in general cases, respectively.

Second, the cache-first and cache-final cases need the same initial delay if $\Delta_1 \geq 0$ and $\Delta_2 \geq 0$ or $\Delta_1 \leq 0$ and $\Delta_2 \leq 0$, i.e., $R_{\mathrm{B}} \leq R_f$ and $R_{\mathrm{H}} \leq R_f$ or $R_{\mathrm{B}} \geq R_f$ and $R_{\mathrm{H}} \geq R_f$. In the former scenario, both the downloading rates are not larger than the playback rate, then the benefits due to high downloading rate described in the last paragraph does not exist. Thus, in this scenario the same initial delay is required no matter which portion is chosen to cache. In the latter scenario, both the downloading rates are not smaller than the playback rate, then pre-buffering is not necessary and the initial delay is zero.

In summary, the initial delay in general cases is bounded by or equal to those in the cache-final and cache-first cases. Therefore, in the following we focus on the two cases for the optimization of caching policies.

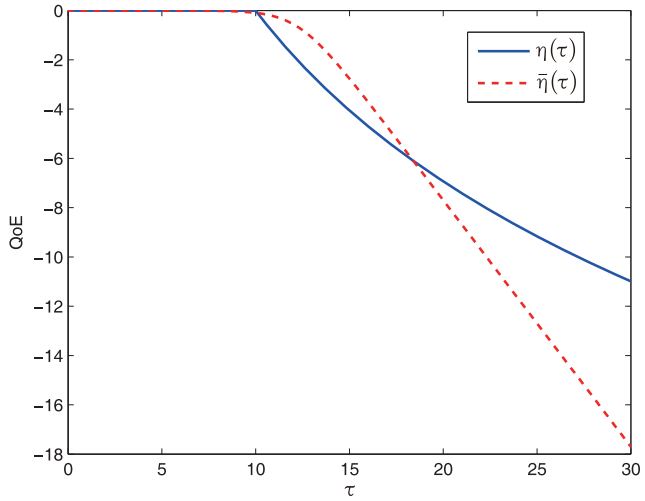

Fig. 1. Comparison of two QoE functions with $\tau_0 = 10$, $\alpha_0 = 0$, $\kappa = 10$, $\alpha_1 = 10\log 10$, $\rho = 10^{-5}$, and $\xi = \log(1 + 10^{-5})$.

B. Approximation of QoE Function

We strive to optimize the caching policy aimed at maximizing the average QoE of the user subject to storage size constraint of the helper, where the average is taken over the random request to the F video files. However, the QoE function given in (1) is non-smooth and non-concave, which is difficult to optimize. Thus, we introduce a smooth and concave approximate QoE function for optimization, which is

$$\overline{\eta}(\tau) = \xi - \log(1 + \rho e^{\tau}), \ \tau \geq 0, \tag{12}$$

where the parameter ρ is a positive constant with small value, and $\xi = \log(1+\rho) + \alpha_0$ ensures that the approximate QoE function has the same value as the real one when $\tau = 0$.

The comparison of the two QoE functions is shown in Fig. 1. We can observe that the approximate QoE function $\overline{\eta}(\tau)$ is able to reflect the nonlinear impact of initial delay on QoE, i.e., a small initial delay does not reduce QoE. For small and medium values of τ, the approximation is reasonable in general, but large difference between the two functions appears when τ is large. Specifically, $\eta(\tau)$ decreases logarithmically for large τ, while $\overline{\eta}(\tau)$ decreases linearly. Fortunately, in practical VoD applications, different schemes such as playback rate adaptation, adaptive resource allocation, admission control, can control the initial delay not too long, which makes using $\overline{\eta}(\tau)$ to approximate $\eta(\tau)$ reasonable.

C. Optimization of Caching Policy

The caching policy optimization problem can be formulated as follows.

$$\max_{x_f} \quad \sum_{f=1}^{F} q_f \xi - q_f \log(1 + \rho e^{\tau_f}) \tag{13a}$$

$$s.t. \quad \sum_{f=1}^{F} B_f x_f \leq C \tag{13b}$$

$$0 \leq x_f \leq 1, \forall f, \tag{13c}$$

where the objective function is the approximate average QoE with random video requests, the initial delay τ_f is given

in (9) and (11) for the cache-final and cache-first cases, respectively, and (13b) is the constraint of storage size at the helper. We focus on the scenario where the storage size of the helper is small so that it cannot cache all video files, i.e., $C < \sum_{f=1}^{F} B_f$; otherwise, the optimization problem is trivial. It is obvious that the optimal caching policy must fully use all storage resources, i.e., constraint (13b) holds with equality.

We next solve problem (13) in the cache-final and cache-first cases, respectively, whose performance can serve as lower and upper bounds for other general cases as analyzed before.

In the cache-final case, we can find from (9) that τ_f is a linear function of x_f, and hence problem (13) is a convex problem. By analyzing the Karush-Kuhn-Tucker (KKT) conditions of the problem, we can obtain the optimal solution of x_f in closed form (see the Appendix for detailed derivations):

$$x_f^\star = \begin{cases} 1, & 0 \le \nu \le -\frac{g_f(1)}{B_f} \\ -\frac{1}{Q_f}\log\left(\frac{t_f(q_f Q_f + \nu B_f)}{-\nu B_f}\right), & -\frac{g_f(1)}{B_f} < \nu < -\frac{g_f(0)}{B_f} \\ 0, & \nu \ge -\frac{g_f(0)}{B_f}, \end{cases} \quad (14)$$

where $Q_f = \left(\frac{B_f}{R_{\mathrm{H}}} - \frac{B_f}{R_f}\right)^+ - \left(\frac{B_f}{R_{\mathrm{B}}} - \frac{B_f}{R_f}\right)^+ \le 0$, $t_f = \rho \exp\left(\frac{B_f}{R_{\mathrm{B}}} - \frac{B_f}{R_f}\right)^+$, $g_f(x) = \frac{q_f t_f Q_f e^{xQ_f}}{1 + t_f e^{xQ_f}}$, and ν is the Lagrangian multiplier. We can find that $x_f^\star$ is a non-increasing function of ν, therefore the optimal ν can be easily obtained by using a bisection method to ensure constraint (13b) holding with equality.

In the cache-first case, we can see from (11) that τ_f is a convex function of x_f. Further considering $\overline{\eta}(\tau_f)$ is a concave and non-increasing function of τ_f, then based on the composition rule of convex functions [16], we know that the objective function (13a) is a concave function of x_f. Therefore, problem (13) is convex in this case, whose globally optimal solution can be obtained efficiently with standard convex optimization algorithms [16].

D. Impact of File Popularity

In this subsection we analyze the impact of file popularity q_f on the optimal caching policy $x_f^\star$. Since only the caching-final case has the closed-form solution, we focus on the analysis in this case based on (14). For the caching-first case, we find by simulations that the conclusions drawn for the caching-final case is also valid (the simulation result is not presented due to the lack of space).

It is easy to see from (14) that $x_f^\star$ is lower and upper bounded by 0 and 1, and within the bounds its behavior is determined by the term $-\frac{1}{Q_f}\log\left(-\frac{t_f(q_f Q_f + \nu B_f)}{\nu B_f}\right) \triangleq \bar{x}_f^\star$. Thus, we focus on the analysis of $\bar{x}_f^\star$ herein.

To see the impact of q_f, we rewrite $\bar{x}_f^\star$ as

$$\bar{x}_f^\star = -\frac{1}{Q_f}\log\left(-q_f Q_f - \nu B_f\right) + \frac{1}{Q_f}\log\left(\nu B_f\right) - \frac{1}{Q_f}\log\rho - \frac{1}{Q_f}\left(\frac{B_f}{R_{\mathrm{B}}} - \frac{B_f}{R_f}\right)^+. \quad (15)$$

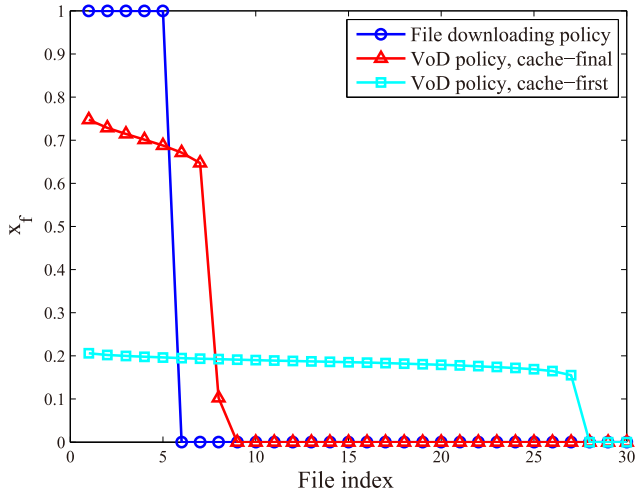

Fig. 2. Caching policies designed for VoD and file downloading, $C = 50$ MB.

When given the size B_f, playback rate R_f, and popularity q_f of the F video files, the Lagrangian multiplier ν can be determined accordingly, which is common for all files. Thus, the final three terms in (15) are constants, and the cached portion $\bar{x}_f^\star$ is determined by the first term in the right-hand side of (15). Then, we can obtain the following observation.

***Observation**: The cached portion $\bar{x}_f^\star$ of the f-th video file logarithmically increases with its popularity q_f.*

The logarithm law means that the cached portions for a highly popular video and a medium popular video may have small difference, which implies that only caching the most popular video files is no longer optimal for VoD service.

IV. Simulation Results

In this section we evaluate the performance of the proposed caching policy for VoD service. For comparison, we also simulate the optimal caching policy for file downloading service proposed in [3] for the single helper scenario as we considered in the paper, i.e., caching the most popular files until the storage resource is used up.

In the simulations, the spectral efficiency of the macro BS and the helper are set as 3 and 5 bps/Hz, respectively, as in [11]. Considering that the macro BS covers more users than the helper, the bandwidth allocated by the macro BS and the helper to the user are set as 0.2 and 2 MHz, respectively, which leads to the downloading rates from the macro BS and the helper as $R_{\mathrm{B}} = 0.6$ Mbps and $R_{\mathrm{H}} = 10$ Mbps. We consider $F = 30$ video files, whose popularity follows Zipf distribution with the parameter of 0.56 [11]. The parameters in the QoE function are set as $\tau_0 = 10$ seconds, $\kappa = 10$, $\alpha_0 = 0$, and $\alpha_1 = 10\log 10$. As for the approximate QoE function, we choose $\xi = \log(1 + 10^{-4})$ and $\rho = 10^{-4}$, which can well approximate the real QoE function. The file size and playback rate are set as $B_f = 10$ MB and $R_f = 0.8$ Mbps, respectively, $f = 1, \ldots, F$, i.e., the playback duration of each video is 100 seconds.

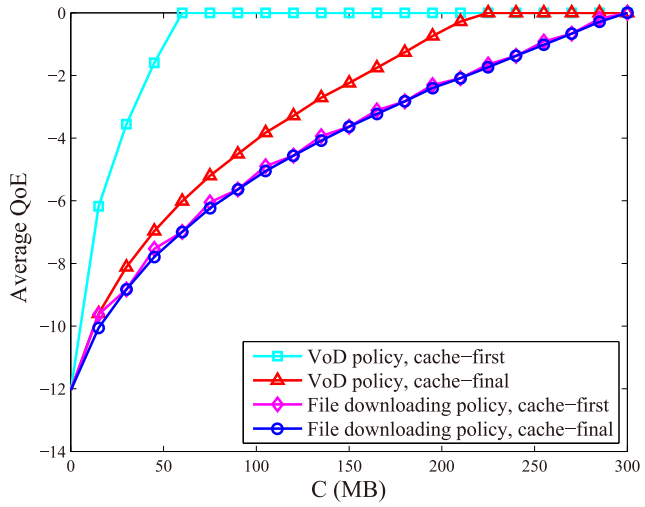

Fig. 3. Average QoE achieved by the proposed and existing caching policies.

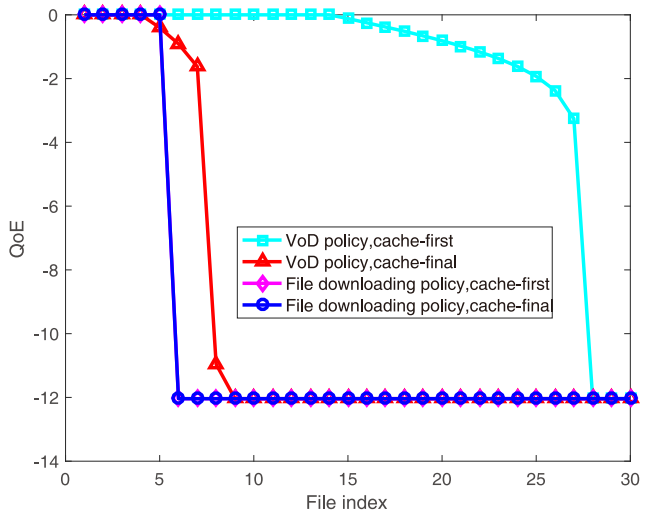

Fig. 4. QoE of the user when requesting every file under the proposed and existing caching policies, $C = 50$ MB.

Figure 2 shows the caching results of the existing caching policy for file downloading and the proposed caching policies for both cache-final and cache-first cases, where the storage size of the helper is set as $C = 50$ MB. The x-axis is the index of the video files, ranking from one (the most popular) to F (the least popular) based on popularity, and y-axis is the cached portion of every video file. We can observe that in contrast to the cache policy for file downloading, which only caches the five most popular videos, the proposed caching policies prefer to cache more videos with smaller portion. The cache-first case can stores more files than the cache-final case. This is because the cache-first case can achieve shorter initial delay than the cache-final case given the same x_f as analyzed before, or inversely, to ensure the same initial delay, the cache-first case only needs to cache a smaller portion and therefore more videos can be cached.

Figure 3 compares the average QoE achieved by the proposed and existing caching policies as a function of the storage size of the helper, where the y-axis shows the values of the real QoE rather than the approximate one. We consider cache-final and cache-first cases for both the proposed caching policy and the policy for file downloading when a video file cannot be fully cached due to the limited storage size. We can see that the impact of cache-first and cache-final is very small for file downloading. The proposed policies perform the same as the policy for file downloading when the storage size is small so that very few videos can be cached or when the storage size is very large so that all video files can be cached. For medium storage size, the proposed policies can achieve higher average QoE, and the gain in the cache-first case is remarkable because it can reduce the initial delay more effectively than the cache-final case.

Although the proposed polices can improve the average QoE when the user randomly requests the F video files, it is not clear whether or not the QoE of the user can be improved for every video file. Figure 4 shows the QoE of the user when requesting every video file, where both the proposed and existing caching policies in cache-first and cache-final cases are considered, and $C = 50$ MB. The x-axis is the index of the video files and y-axis is the QoE of the user. It can be observed that compared to the file downloading policy, the proposed caching policy in the cache-final case can largely improve the QoE for the sixth and seventh video files with a small QoE degradation of the fifth video file, while in the cache-first case the improvement of QoE is dramatic for most video files.

V. Conclusions

In this paper we optimized the caching policy for VoD service by taking into account the impact of initial delay on the QoE. We analyzed the initial delays for the cache-final and cache-first cases, and introduced an approximate QoE function to overcome the difficulty for caching policy optimization caused by the original non-smooth non-concave QoE function. We obtained the optimal caching policy for the caching-final case in closed form and for the caching-first case numerically, and then analyzed the impact of file popularity on the caching policy. Simulation results show the performance gain of the proposed caching policies.

VI. Acknowledgement

This work was supported in part by National High Technology Research and Development Program of China (No. 2014AA01A705) and by National Natural Science Foundation of China (No. 61429101).

Appendix
Optimal Solution to (13) in Cache-final Case

By defining $t_f = \rho \exp\left(\frac{B_f}{R_{\rm B}} - \frac{B_f}{R_f}\right)^+$ and $Q_f = \left(\frac{B_f}{R_{\rm H}} - \frac{B_f}{R_f}\right)^+ - \left(\frac{B_f}{R_{\rm B}} - \frac{B_f}{R_f}\right)^+$, we can write the KKT condi-

tions of problem (13) as follows:

$$0 \le x_f^\star \le 1, \lambda_f \ge 0,\ \mu_f \ge 0, \nu \ge 0,\ \forall f \tag{16a}$$

$$\nu \left(\sum_{f=1}^{F} B_f x_f^\star - C \right) = 0 \tag{16b}$$

$$\lambda_f x_f^\star = 0, \forall f \tag{16c}$$

$$\mu_f (x_f^\star - 1) = 0, \forall f \tag{16d}$$

$$\frac{q_f t_f Q_f e^{x_f^\star Q_f}}{1 + t_f e^{x_f^\star Q_f}} - \lambda_f + \mu_f + \nu B_f = 0, \forall f, \tag{16e}$$

where λ_f, μ_f, and ν are Lagrangian multipliers, and $x_f^\star$ denotes the optimal solution.

Define $g_f(x_f^\star) = \frac{q_f t_f Q_f e^{x_f^\star Q_f}}{1+t_f e^{x_f^\star Q_f}}$, which is a monotonically increasing function of $x_f^\star$. Then, we can obtain from (16e) that

$$\lambda_f = g_f(x^\star) + \mu_f + \nu B_f. \tag{17}$$

Substitute (17) into (16c) and considering $\lambda_f \ge 0$, we can obtain

$$x_f^\star \left(g_f(x_f^\star) + \mu_f + \nu B_f \right) = 0, \tag{18}$$

$$\mu_f \ge -g_f(x_f^\star) - \nu B_f. \tag{19}$$

1) If $\nu \ge -\frac{g_f(0)}{B_f}$, i.e., $g_f(0) + \nu B_f \ge 0$, then $g_f(0) + \mu_f + \nu B_f < 0$ will not hold because $\mu_f \ge 0$ according to (16a). Thus, we have $g_f(0) + \mu_f + \nu B_f \ge 0$, then $g_f(x_f^\star) + \mu_f + \nu B_f \ge 0$ since $x_f^\star \ge 0$ and $g_f(x_f^\star)$ is an increasing function. If $x_f^\star > 0$, then $g_f(x_f^\star) + \mu_f + \nu B_f > 0$, which makes (18) infeasible. Therefore, we have $x_f^\star = 0$ in this case.

2) If $\nu < -\frac{g_f(0)}{B_f}$, i.e., $g_f(0) + \nu B_f < 0$, we can show that $g_f(0) + \mu_f + \nu B_f \ge 0$ will not hold because this inequality leads to $\mu_f > 0$ and $x_f^\star = 0$ as analyzed in the above case, which does not satisfy (16d). Thus, in this case we have $g_f(0) + \mu_f + \nu B_f < 0$. Since $g_f(x_f^\star)$ is an increasing function and $x_f^\star \ge 0$, we know that (19) holds only when $x_f^\star > 0$. Then, from (18) we have

$$g_f(x_f^\star) + \mu_f + \nu B_f = 0. \tag{20}$$

We have $\mu_f = -g_f(x_f^\star) - \nu B_f$ from (20). By substituting it into (16d) and considering $\mu_f \ge 0$, we obtain

$$(g_f(x_f^\star) + \nu B_f)(x_f^\star - 1) = 0, \tag{21}$$

$$\nu \le -\frac{g_f(x_f^\star)}{B_f}. \tag{22}$$

i) If $0 \le \nu \le -\frac{g_f(1)}{B_f}$, i.e., $g_f(1) + \nu B_f \le 0$, then $g_f(x_f^\star) + \nu B_f \le 0$ since $x_f^\star \le 1$ and $g_f(x_f^\star)$ increases with $x_f^\star$. If $x_f^\star < 1$, then $g_f(x_f^\star) + \nu B_f < 0$, which makes (21) infeasible. Thus, we have $x_f^\star = 1$ in this case.

ii) If $\nu > -\frac{g_f(1)}{B_f}$, i.e., $g_f(1) + \nu B_f > 0$, (22) holds only when $x_f^\star < 1$ considering that $g_f(x_f^\star)$ is an increasing function. Then, we can obtain from (21) that

$$g_f(x_f^\star) + \nu B_f = 0, \tag{23}$$

from which $x_f^\star$ can be solved as

$$x_f^\star = -\frac{1}{Q_f} \log \left(\frac{t_f (q_f Q_f + \nu B_f)}{-\nu B_f} \right) \triangleq h(\nu). \tag{24}$$

Based on the above analysis, we obtain $x_f^\star$ as

$$x_f^\star = \begin{cases} 1, & 0 \le \nu \le -\frac{g_f(1)}{B_f} \\ h(\nu), & -\frac{g_f(1)}{B_f} < \nu < -\frac{g_f(0)}{B_f} \\ 0, & \nu \ge -\frac{g_f(0)}{B_f} \end{cases} \tag{25}$$

which gives rise to (14).

REFERENCES

[1] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[2] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, 2012.

[3] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[4] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[5] C. Yang, Y. Yao, Z. Chen, and B. Xia, "Analysis on cache-enabled wireless heterogeneous networks," *IEEE Trans. on Wireless Communications*, vol. 15, no. 1, pp. 131–145, Jan. 2016.

[6] H. Ahlehagh and S. Dey, "Adaptive bit rate capable video caching and scheduling," in *Proc. IEEE WCNC*, 2013.

[7] ——, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Trans. on Networking*, vol. 22, no. 5, pp. 1444–1462, Oct. 2014.

[8] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1431–1445, Oct. 2013.

[9] Y. Wang, X. Zhou, M. Sun, L. Zhang, and X. Wu, "A new QoE-driven video cache management scheme with wireless cloud computing in cellular networks," *Mobile Networks and Applications*, Feb. 2016. [Online]. Available: http://dx.doi.org/10.1007/s11036-016-0689-5

[10] H. Su, S. Han, and C. Yang, "Caching policy optimization for rate adaptive video streaming," in *Proc. IEEE Globalsip*, 2016.

[11] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[12] C. Quadros, A. Santos, M. Gerla, and E. Cerqueira, "QoE-driven dissemination of real-time videos over vehicular networks," *Computer Communications*, vol. 91, pp. 133–147, 2016.

[13] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: Where microeconomics meets psychophysics and quality of experience," *Telecommun. Syst.*, pp. 1–14, 2011.

[14] D. Bethanabhotla, G. Caire, and M. J. Neely, "Adaptive video streaming for wireless networks with multiple users and helpers," *IEEE Transactions on Communications*, vol. 63, no. 1, pp. 268–285, Jan. 2015.

[15] S. Egger, P. Reichl, T. Hoßfeld, and R. Schatz, "'Time is bandwidth'? narrowing the gap between subjective time perception and quality of experience," in *Proc. IEEE ICC*, 2012.

[16] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.