This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

Markov Decision Policies for Dynamic Video Delivery in Wireless Caching Networks

Minseok Choi, Albert No, Mingyue Ji, *Member, IEEE,* and Joongheon Kim, *Senior Member, IEEE,*

***Abstract*—This paper proposes a video delivery strategy for dynamic streaming services which maximizes time-average streaming quality under a playback delay constraint in wireless caching networks. The network where popular videos encoded by scalable video coding are already stored in randomly distributed caching nodes is considered under adaptive video streaming concepts, and distance-based interference management is investigated in this paper. In this network model, a streaming user makes delay-constrained decisions depending on stochastic network states: 1) caching node for video delivery, 2) video quality, and 3) the quantity of video chunks to receive. Since wireless link activation for video delivery may introduce delays, different timescales for updating caching node association, video quality adaptation, and chunk amounts are considered. After associating with a caching node for video delivery, the streaming user chooses combinations of quality and chunk amounts in the small timescale. The dynamic decision making process for video quality and chunk amounts at each slot is modeled using Markov decision process, and the caching node decision is made based on the framework of Lyapunov optimization. Our intensive simulations verify that the proposed video delivery algorithm works reliably and also can control the tradeoff between video quality and playback latency.**

***Index Terms*—Wireless caching network, video delivery, stochastic network optimization, Lyapunov optimization, Markov decision process**

I. Introduction

Within few years, it has been expected that tens of exabytes of global data traffic be handled on daily basis, and on-demand video streaming will account for about 70% of them [1]. In on-demand video streaming services, a relatively small number of popular contents is requested at ultra high rates and playback delay is one of the most important measurement criteria of goodness [2], [3]. To deal with the characteristics, wireless caching technologies have been studied for video streaming services by storing popular videos in caching helpers located nearby users during off-peak time [4]–[6]. Therefore, it is obvious that storing and streaming of video files are of major research interests in wireless caching networks.

There have been major research results for caching popular files in stochastic wireless caching networks [7]–[9]. The major goal of those research results was to design the optimal caching policies according to the popularity distribution of contents and wireless network topology. The probabilistic caching policy was proposed in [7] to adapt characteristics of the stochastic network. Many probabilistic caching methods have been proposed depending on various optimization goals, e.g., maximization of cache hit probability [7], average success probability of content delivery [8], and delivery rate of multiple content requests [9]. However, these works on the caching policy do not consider the identical content with different qualities.

Since video files can be encoded to multiple versions which differ in the quality levels, the video caching policies having different quality levels have been widely studied in [10], [11]. Many researchers have proposed the static content placement policies under the consideration of differentiated quality requests for the same content, given probabilistic quality requests [10] or minimum quality requirements [11]. The above works are focused only on the content placement problem with different qualities, however, the delivery policy of contents with different qualities has not yet been studied much.

For video delivery/streaming, there are some necessary decisions to be made: 1) which caching node will deliver the video, 2) which quality of video will be provided, and 3) how many video chunks will be transmitted. The first one is called *node association problem*, and in most research contributions that do not consider different quality levels for the same file, the file-requesting user is allowed to receive the desired video from the caching node under the strongest channel condition [8], [12]. The node associations for video delivery in heterogeneous caching networks have been studied in [13]–[15]. On the other hand, when videos with different qualities are independently cached, more elaborate node association algorithm is necessary, because decision on the content quality relies on the node association. In this case, the video delivery policy was proposed in [16] to pursue time-average video quality maximization while avoiding playback delays.

Since dynamic video streaming allows each chunk to have a different quality depending on time-varying network conditions, some researchers addressed transmission schemes which serve the video by dynamically selecting the quality level [17]. In [18] and [19], the scheduling policies which maximize the network utility function of time-averaged video quality in small-cell networks and device-to-device networks were pro-

This research was supported in part by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00170, Virtual Presence in Moving Objects through 5G), National Science Foundation (NSF) CCF-1817154, and NSF SpecEES-1824558.

M. Choi is with the Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089, The United States, e-mail: choimins@usc.edu

A. No is with the Department of Electronic and Electrical Engineering, Hongik University, Seoul 04066, South Korea, e-mail: albertno@hongik.ac.kr.

M. Ji is with the Department of Electronic and Computer Engineering, University of Utah, Salt Lake City, UT 84112, The United States, e-mail: mingyue.ji@utah.edu.

J. Kim is with the School of Electrical Engineering, Korea University, Seoul 02841, South Korea, e-mail: joongheon@korea.ac.kr.

J. Kim and A. No are corresponding authors of this paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

posed. The authors of [20] considered scalable video coding (SVC) and proposed dynamic resource allocation and quality selection under the pricing strategy for interference. While the video delivery policies of [17]–[20] are operated at the base station (BS) side, the decision policy of video quality level at user sides was not considered. This scenario is consistent with the practical real-world software implementation of dynamic adaptive streaming over HTTP (DASH) [21], in which users dynamically choose the most appropriate video quality. Even though the work of [16] can choose the video quality at the user side, however it cannot dynamically change the video quality without updates of node association.

Further, control of the amount of receiving chunks depending on stochastic network states has been largely neglected in above existing researches about video delivery. Even though the authors of [16] and [20] maximize the long-term time-average video quality under the various constraints, their metrics representing video quality is obtained by averaging the number of quality selections at each time slot. This method would be not enough to evaluate the user's quality of service, especially when the transmission rate varies over the video streaming service time. In practice, when channel experiences deep fading and only the low-quality video is available, it would not be the best choice to receive as many chunks as the channel condition can provide. Rather than receiving many low-quality chunks, the user could prefer to wait channel conditions to be better and then to receive high-quality chunks, if it guarantees no playback delay. Therefore, by considering decision process of combinations of video quality and chunk amounts, we can formulate the optimization problem which maximizes the average video quality per each received chunk. There has been not many researches yet considering the average video quality per chunk as a performance metric as well as aiming at adjustments of receiving chunk amounts depending on caching node and user states.

This paper addresses dynamic video delivery in two different timescales for wireless caching networks with consideration of user mobility. There are some existing works of mobility-aware caching [22]–[24] in which user mobility is known at the centralized server or randomness of user mobility is probabilistically modeled. Meanwhile, content delivery in wireless caching networks with moving users has not been explored yet. When the delivery decisions are made at the user side, it is reasonable to utilize the knowledge of user mobility for dynamic content delivery. In addition, joint optimization of long-term content caching and short-term delivery was studied in [20], [25]; however, content delivery decisions in different timescales has not been investigated.

This paper proposes a video delivery policy in the wireless caching network for dynamic streaming services. The main contributions are as follows:

- This paper proposes a dynamic video delivery policy depending on stochastic network states. The proposed policy makes three different but necessary decisions for the streaming user: 1) caching node for video delivery, 2) video quality and 3) the quantity of video chunks to receive. To the best of the authors' knowledge, no research has yet considered all of those video delivery decisions.
- Caching node association and decisions of video quality and amounts of receiving chunks are conducted in different timescales. Since wireless link activation for video delivery is time-consuming, it is reasonable that caching node association is performed slower than decisions of video quality and amounts of receiving chunks. The optimization framework of video delivery policy in two different timescales is constructed based on frame-based Lypunov optimization theory [26] and Markov decision process (MDP).
- The proposed content delivery method reflects user mobility. Therefore, the user enables to associate with the caching node based on estimation of the short-term decisions on quality and receiving chunk amounts in future. Note that the knowledge of user mobility can be obtained because delivery decisions are made at the user side.
- The proposed technique maximizes the average streaming quality while averting playback latency, and can control the tradeoff between video quality and playback delay. Different from [16] and [20], we adopt the long-term average video quality per each received chunk as a performance metric.
- We perform simulations to verify the proposed video delivery policy and to show the advantages of using Lyapunov optimization theory and MDP.

The rest of the paper is organized as follows. The wireless video caching network model is described in Sec. II. The optimization problem for dynamic video delivery is formulated in Sec. III. The rule of caching node association and control policies of quality level and receiving chunk amounts are proposed in Sec. IV and Sec. V. Simulation results are presented in Sec. VI and Sec. VII concludes this paper.

II. Network Model

The wireless caching network model is described in this section. In addition, the user queue model and channel model are introduced and the dependency of the necessary video delivery decisions on the user queue and channel models is explained. The inter-user interference is roughly included by using distance-based interference management.

A. Wireless caching network model

This paper considers wireless caching network model where a user requests certain video file for one of caching nodes around the user, as shown in Fig. 1. The BS has already pushed popular video files during off-peak hours to caching nodes which have the finite storage size. Since we focus on video delivery, the caching policy is out of scope for this paper and only the desired video is considered. Suppose that the desired video has L quality levels. Therefore, there are L types of caching nodes, and the type-l caching nodes can deliver the video of any quality $q \in \mathcal{L}_l$, where $\mathcal{L}_l = \{1, \cdots, l\}$ is the set of qualities which the type-l caching node can provide. Thus, the type-L caching nodes can provide all quality levels from the quality set $\mathcal{L}_L$. Note that simple definition of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

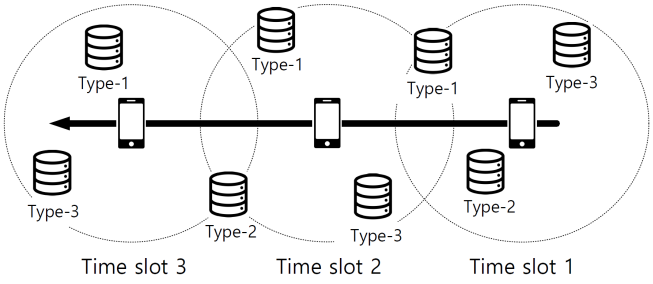

Fig. 1. Network Model

$\mathcal{L}_l = \{1, \cdots, l\}$ is assumed, but the proposed technique can be coordinated with any arbitrary quality set as long as multiple versions of the same video having different qualities are stored in caching nodes.

The identical files of different qualities are stored in multiple caching nodes, and the type-l caching nodes are distributed by the independent Poisson Point Processes (PPPs) with intensity λp_q [7], where p_q indicates the caching probability of the requested video encoded to provide any quality $q \in \{1, \cdots, l\}$. Suppose that the caching policy is already determined, i.e., all p_q for all q are given. In addition, videos of different qualities have different file sizes and N_q denotes the file size of quality q in bits, satisfying $N_m < N_q$ for all $m, q \in \mathcal{L}_L$ and $m < q$.

User mobility is also captured in network model. The user is moving towards certain direction and periodically searches for a caching node to receive the desired video file. As shown in Fig. 1, geological distribution of caching nodes around the user varies at each time slot, so the caching node decision should be appropriately updated. Further, this paper also considers how many chunks of which quality level to be requested from the user depending on the stochastic network environment. When there are other users who exploit the wireless caching network with the same resource, the target streaming user is interfering with them. We adopt the distance-based interference management to limit the interference power lower than certain threshold, and details are explained in Section II-C.

B. User queue model and channel model

A video file consists of many sequential chunks. The user receives the video file from a caching node and processes data for the streaming service in units of chunks. Each chunk of a file is responsible for some playback time of the entire stream. As long as all chunks are in correct sequence, each chunk can have different quality in dynamic streaming. Therefore, the user can dynamically choose video quality level in each chunk processing time. By using the queueing model, it can be said that the playback delay occurs when the queue does not have the chunk to be played. In this sense, receiver queue dynamics collectively reflects the various factors which cause the playback delay.

In general, the user model has its own arrival and departure processes. The user queue dynamics in each discrete time slot $t \in \{0, 1, \cdots\}$ can be represented as follows:

$$Q(t+1) = \max\{Q(t) - c, 0\} + M(t) \text{ and } Q(0) = 0, \quad (1)$$

where $Q(t)$ stands for the queue backlog at time t. In addition, the departure c is a constant because the streaming user does not change the video playback rate in general. The arrival $M(t)$ denotes the number of received chunks at time t.

Let the caching node which the user chooses for video delivery be α. Then, $h(\alpha, t) = \sqrt{D(\alpha, t)w(t)}u(t)$ represents the Rayleigh fading channel between the user and the caching node α at time t, where $D(\alpha, t) = 1/d(\alpha, t)^\beta$ controls path loss, where $d(\alpha, t)$ is the user-caching node distance at time t, β is the path loss exponenet, and $w(t)$ is the shadowing effect and follows normal distribution with variance σ_w^2, i.e., $w \sim \mathcal{N}(0, \sigma_w^2)$. In addition, u represents the fast fading component having a complex Gaussian distribution, $u \sim \mathcal{CN}(0, 1)$. The link rate can be simply given by $R(\alpha, t) = \mathcal{W} \log_2 \left(1 + \frac{\Psi |h(\alpha,t)|^2}{\Upsilon + 1}\right)$, where $\mathcal{W}$, Ψ, and Υ are bandwidth, transmit SNR, and interference-noise-ratio (INR), respectively.

The number of received chunks necessarily depends on the caching node decision α and its link rate. In this paper, each slot interval is determined to be channel coherence time t_c. Then, the number of received chunks $M(t)$ is constrained by

$$M(t) N_{q(t)} \leq t_c R(\alpha, t). \quad (2)$$

Since $M(t)$ and $N_{q(t)}$ are nonnegative integers,

$$M(t) N_{q(t)} \leq B(\alpha, t) = \lfloor t_c R(\alpha, t) \rfloor. \quad (3)$$

Therefore, the decision of $M(t)$ depends on the decisions of $\alpha(t)$ and $q(t)$ and the random network event $R(\alpha, t)$.

C. Distance-based interference management

Although many existing works have investigated complex interference management schemes such as interference alignment and interference cancellation, most of researches on the wireless caching and delivery policy have still used simple interference avoidance based interference management schemes, e.g., by spectrum sharing [28] or assuming the protocol model [29]. For simplicity, this paper considers the distance-based interference control for node association (i.e., link activation) for video delivery. The design ideas can be extended to other more sophisticated interference management schemes [30], [31].

Activation of the new link for video delivery in the wireless caching network means that the network allows the new streaming user to interfere with existing users. A new user causes two types of interference, 1) from the caching nodes already serving existing users to the new user, and 2) from the caching node associated with the new user to existing users. Therefore, we define R_U and R_N as the safety distances for streaming users and their associated caching nodes respectively to keep the interference levels below the predetermined threshold of ρ. In other words, a new streaming user who wants to exploit the wireless caching network should be generated outside the radius R_N of all caching nodes associated with the existing users. In addition, the new user has to find the caching node to receive the desired content outside the radius R_U of all existing users. The safety distances of R_U and R_N should be carefully chosen, and then a new pair of a caching node and a user can be generated only when their interference

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

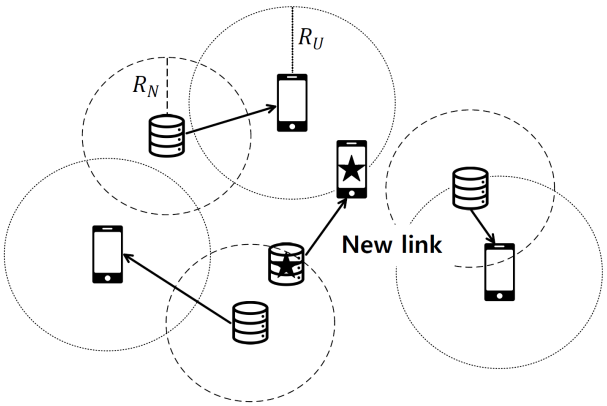

Fig. 2. Safety radius and activation of new link for video delivery

power is acceptable for every existing video delivery link, as shown in Fig. 2.

In this regard, a new pair of a caching node and a streaming user is allowed for video delivery through following two steps. The first step is to confirm the INR, say Υ_0, at the new streaming user to be lower than ρ. Here, Υ_0 is the ratio of the aggregated interference power from all the activated caching nodes to noise variance. If $\Upsilon_0 > \rho$, the system does not allow the new user to exploit the wireless caching network, and the new user should directly request the desired content from the server which has a whole file library or wait for content delivery in future. For example, suppose that interference power from the nearest interfering caching node to the new user dominates Υ_0. We further let Ψ_0 be the transmit SNR of the interfering node, and d_n be the distance from the interfering node to the new user. Then, INR becomes $\Upsilon_0 = \frac{\Psi_0}{d_n^2}$, and $R_N \geq \sqrt{\Psi_0/\rho}$ to guarantee $\Upsilon_0 \leq \rho$.

Although the interference power at the new user is safe to exploit the wireless caching network, i.e., $\Upsilon_0 \leq \rho$, the caching node associated with the new user will be able to degrade the signal-to-interference-plus-noise ratios (SINRs) of existing users. Thus, the second step is required, in which the new user should find the caching node to receive the desired content with sufficiently large link rate as well as not to significantly interfere with other users. Let one of existing users have a margin of INR to guarantee ρ before activating the new link, denoted by $\delta = \rho - \Upsilon_0$. Since the interference signal from the new caching node is independent on other interfering nodes, $R_U = \sqrt{\Psi_0/\delta}$ is obtained similar to the case of R_N. Therefore, the new caching node should be chosen outside the radius of R_U from every existing user whose margin of INR would be different from each other.

Even though the new user and its caching node are found while limiting all interference levels at users lower than ρ, the newly generated link between them could not be enough to provide reliable content transmissions due to bad channel conditions. Therefore, we investigate the existence of the caching node around the new user which stores the requested content and can deliver the content reliably. Let the minimum SINR for reliable video delivery denoted by $\gamma_{\min}$. Then, the probability that at least one caching node can successfully deliver the desired content to the new user is represented by

$$\eta = \Pr\left\{ \frac{\Psi |h_{n,1}|^2}{\Upsilon + 1} \geq \gamma_{\min} \right\}, \tag{4}$$

where $h_{n,1}$ is the channel gain between the new user and the caching node whose channel condition is the strongest among the nodes storing the desired content of the user. According to order statistics and [8], the cumulative distribution function of the smallest reciprocal of channel power is $F_{\xi_{n,1}}(\xi) = 1 - e^{-\pi\Gamma(2)\lambda_n \xi}$, where $\xi_{n,1} = 1/|h_{n,1}|^2$ and λ_n is the intensity of PPP of nodes caching the desired content. According to (4), η can be found by

$$\eta = 1 - \exp\left\{ -\pi\Gamma(2)\lambda_n \frac{\Psi}{\gamma_{\min}(\Upsilon + 1)} \right\}. \tag{5}$$

Then, by introducing the minimum probability of finding at least one caching node for reliable video delivery denoted by $\eta_{\min}$, a set of $\{\gamma_{\min}, \eta_{\min}\}$ can be considered as a criterion for new reliable link activation which satisfies $\eta \geq \eta_{\min}$. In this regard, we can verify how much interference power is acceptable to satisfy the criterion of $\{\gamma_{\min}, \eta_{\min}\}$, as follows:

$$\begin{aligned} &1 - \exp\left\{ -\pi\Gamma(2)\lambda_n \frac{\Psi}{\gamma_{\min}(\Upsilon + 1)} \right\} \geq \eta_{\min} \\ &\iff \frac{\pi\Gamma(2)\lambda_n \Psi}{\gamma_{\min} \ln(\frac{1}{1-\eta_{\min}})} - 1 \geq \Upsilon. \end{aligned} \tag{6}$$

Thus, if all network parameters are given, the threshold of interference power can be determined by

$$\rho = \frac{\pi\Gamma(2)\lambda_n \Psi}{\gamma_{\min} \ln(\frac{1}{1-\eta_{\min}})} - 1. \tag{7}$$

On the other hand, if the network requires the target criterion of interference management, i.e., ρ, $\gamma_{\min}$, and $\eta_{\min}$ are given, the system can determine how much transmit power is required and/or how many caching nodes store the desired video.

In this paper, the minimum SINR threshold is set so that the chunk of the smallest size (i.e., the lowest quality) is deliverable at least, i.e., $t_0 \mathcal{W} \log_2(1 + \gamma_{\min}) = N_1$. Then, we can say that caching nodes which store the desired content should be distributed with the intensity of $\lambda_{\min}$ at least, as follows:

$$\lambda_n \geq \lambda_{\min} = \frac{(2^{\frac{N_1}{t_0 \mathcal{W}}} - 1) \ln(\frac{1}{1-\eta_{\min}})(1 + \Upsilon)}{\pi\Gamma(2)\Psi}. \tag{8}$$

III. Dynamic Video Delivery Policies

The dynamic video delivery policy in two different timescales is described in this section. When the data rate of the delivery link not in outage is varying, i.e., the number of receiving video chunks at each discrete slot is not fixed, video quality per received chunk is the reasonable performance metric. Therefore, the problem that maximizes quality per received chunk with the constraints for playback delay and channel capacity is formulated.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

A. Video delivery decisions

The goal of this paper is to find the appropriate three decisions at each slot t in the network model illustrated in Section II: 1) caching node for video delivery $\alpha(t)$, 2) video quality level $q(t)$, and 3) the quantity of receiving chunks $M(t)$. However, to update the caching node association, the time-consuming process is required in which the user sends the request signal for video delivery and the caching node approves it. Therefore, new caching node association is hardly performed as frequent as receiving chunks, and we suppose that the decision on $\alpha(t)$ is made at larger timescale than decisions on $q(t)$ and $M(t)$.

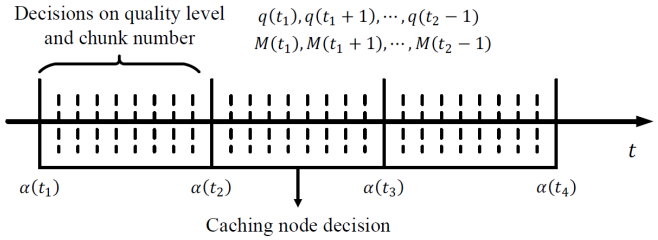

Fig. 3. Different timescales for decisions on $\alpha(t)$, $q(t)$ and $M(t)$

In this sense, the user decides $q(t)$ and $M(t)$ at time slots $t \in \{0, 1, 2 \cdots\}$, but caching node decisions are performed at time slots $t = 0, T, 2T, \cdots$, where T is the time interval for caching node association. The time slot for the k-th caching node decision is denoted by $t_k = (k-1)T$ for $k \in \{1, 2, \cdots\}$. Different timescales of decisions on $\alpha(t)$, $q(t)$ and $M(t)$ are described in Fig. 3. Let the k-th frame for caching node decision be $\mathcal{T}_k = \{t_k, t_k + 1, \cdots, t_k + T - 1\}$. As shown in Fig. 3, after associating with the caching node $\alpha(t_k)$ at time t_k, decisions on quality level $q(t)$ and chunk amounts $M(t)$ are performed over $t \in \mathcal{T}_k$ to receive the desired video from $\alpha(t_k)$. Therefore, $q(t) \in \mathcal{L}_{l(\alpha(t_k))}$ and $M(t)N_{q(t)} \leq B(\alpha(t_k), t)$ should be satisfied for $t \in \mathcal{T}_k$, where $l(\alpha(t_k))$ is the type of the caching node $\alpha(t_k)$.

The user can make the candidate set of caching nodes denoted by $\mathcal{A}(t_k)$, and $\alpha(t_k) \in \mathcal{A}(t_k)$. All caching nodes in $\mathcal{A}(t_k)$ should be outside the radius R_U of all existing users to limit the interference power lower than ρ. To avoid the situation in which no caching node can deliver the desired video, i.e., $\mathcal{A}(t_k) > 0$, the caching nodes which provide SINRs larger than γ_{min} are assumed to be outside the radius R_U of all existing users. $\mathcal{A}(t_k)$ consists of up to L caching nodes, i.e., $|\mathcal{A}(t_k)| \leq L$, in which each caching node in $\mathcal{A}(t_k)$ belongs to different types. If there are several nodes of type-l, the user takes one of them whose channel condition is the strongest. There is no reason to choose another type-l caching node while leaving the node with the strongest channel if another streaming user does not request the video from that strongest node. In addition, $|\mathcal{A}(t_k)| < L$ means that $L - |\mathcal{A}(t_k)|$ caching node types do not exist around the user. Suppose that the new streaming user is already generated outside the radius R_N of all existing caching nodes and the INR Υ is observed at the new user. Also, another user's link activation is banned around the target user due to the interference issue. Then, we just consider the node association problem of the new streaming user with respect to the candidate set $\mathcal{A}(t_k)$ while the INR Υ is observed.

B. Problem formulation

For determining the appropriate video delivery policy, three performance metrics are considered: playback delay, average streaming quality, and video quality fluctuations. The dynamic streaming service enables to provide the differentiated quality requirements to users; therefore, video quality per received chunk becomes a performance metric. The playback delay is obviously the most important factor for the streaming user's satisfaction. In addition, for streaming services, users are very difficult to endure the dramatic fluctuations of video quality. Therefore, this paper supposes that the user pursues the high-quality video unless the delay constraint is not suffered and quality does not vary extremely during video playback.

Based on these goals, we can formulate the optimization problem which minimizes the quality degradation constrained on averting queue emptiness and extreme quality fluctuations as follows:

$$\{\boldsymbol{\alpha}, \boldsymbol{q}, \boldsymbol{M}\} = \underset{\alpha \in \mathcal{A},\ q \in \mathcal{L}_{l(\alpha)}}{\text{argmin}} \lim_{K \to \infty} \mathbb{E}\left[\frac{1}{KT} \sum_{t=0}^{KT-1} (\bar{\mathcal{P}} - \mathcal{P}(q(t))) \cdot M(t)\right] \tag{9}$$

$$\text{s.t.} \lim_{t' \to \infty} \frac{1}{t'} \sum\nolimits_{t=0}^{t'-1} \mathbb{E}[Z(t)] < \infty \tag{10}$$

$$M(t)N_{q(t)} \leq B(\alpha, t) \tag{11}$$

$$|q(t) - q(t-1)| \leq 1 \tag{12}$$

$$\alpha(t_k) \in \mathcal{A}(t_k),\ \forall k \in \{1, 2, \cdots\} \tag{13}$$

$$q(t) \in \mathcal{L}_{l(\alpha(t_k))},\ \forall t \in \mathcal{T}_k,\ k \in \{1, 2, \cdots\}, \tag{14}$$

where $\mathcal{P}(q(t))$ is quality measure of $q(t)$ and $\bar{\mathcal{P}}$ is the maximum quality measure, i.e., equation (9) is the time averaged video quality degradation. Decision vectors are represented as $\boldsymbol{\alpha} = [\alpha(t_1), \cdots, \alpha(t_K)]$, $\boldsymbol{q} = [q(0), q(1), \cdots, q(KT-1)]$ and $\boldsymbol{M} = [M(0), M(1), \cdots, M(KT-1)]$. Specifically, the expectation of (9) is with respect to random channel realizations and stochastic distributions of caching nodes. The constraint (10) is for limiting the playback latency (i.e., queueing delay), and the constraint (11) is came from (3), which demonstrates that decisions on $q(t)$ and $M(t)$ depend on the channel state of the associated caching node α. The dramatic quality fluctuations can be avoided by addressing the constraint (12). Lastly, the constraints (13) and (14) give the candidate sets of caching node and video quality.

As mentioned earlier, playback delay occurs when the next chunk is not arrived in the queue. If the user is assumed to receive chunks in order, the playback latency is generated when $Q(t) = 0$. Then, the long-term time-averaged playback delay occurrence rate can be limited by taking the constraint of

$$\lim_{t' \to \infty} \mathbb{E}[Q(t)] > 0. \tag{15}$$

Here, $Z(t) = \tilde{Q} - Q(t)$ is introduced to convert (15) to the strong stability form of the queuing system. Taking a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

sufficiently large constant $\tilde{Q}$ which affects the maximal queue backlog, (15) can be approximately replaced by the constraint (10) that has a role of avoiding queue emptiness. Based on the Lyapunov optimization theory, the upper bound on the time-average queue length of $Z(t)$ is also derived by using the algorithm which minimizes the Lyapunov drift [33]; finally, it makes chunks accumulated in the user queue enough to avoid the playback delay, i.e., to achieve queue stability in (15).

From (1), the queue dynamics of $Z(t)$ can be represented as follows:

$$Z(t+1) = \min\{Z(t) + c, \tilde{Q}\} - M(t) \text{ and } Z(0) = \tilde{Q}. \quad (16)$$

Even though the update rules of $Q(t)$ and $Z(t)$ are different, both queue dynamics mean the same video chunk processing. Therefore, playback delay due to emptiness of $Q(t)$ can be explained by queueing delay of $Z(t)$. By Littles' Law [32], the expected value of $Z(t)$ is proportional to the time-averaged queueing delay. The similar method to reduce the video playback delay by introducing $Z(t)$ and pursuing its stability was used in the work of [16].

From the optimization problem (9)–(11), we can intuitively see how decisions are made depending on $Q(t)$. Suppose that the queue is almost empty. In this case, the user prefers the caching node whose channel condition is strong, pursues low-quality file, and tries to receive as many chunks as possible to stack many chunks in the queue. However, all of those decisions could degrade the average streaming quality. When the caching node with the strongest channel condition belongs to type-1, it can be better to associate with the caching node of another type in terms of average quality. In addition, when low quality is chosen, receiving too many chunks may not be a good choice. The user would prefer to receive the small number of chunks in current time-step and wait the better channel condition. If the channel condition is improved at the next time-step, the user can request many chunks of high-quality video. Thus, those decisions are strongly dependent on the queue state $Q(t)$, the caching node distribution, and channel conditions of caching node candidates.

IV. Caching Node Decision Policy

For avoiding the queue emptiness, i.e., pursuing queue stability of $Z(t)$, the optimization problem of (9)–(11) are solved based on the Lyapunov optimization theory. However, since the timescale of decision on $\boldsymbol{\alpha}$ is larger than that of decisions on $\boldsymbol{q}$ and $\boldsymbol{M}$, the frame-based Lyapunov optimization theory [26] is used for caching node decision. Lyapunov function $L(t)$ can be defined as $L(t) = \frac{1}{2}Z(t)^2$. Then, let $\Delta(.)$ be a frame-based conditional Lyapunov function that can be formulated as $\Delta(t_k) = \mathbb{E}[L(t_k + T) - L(t_k)|Z(t_k)]$, i.e., the drift over the time interval T. The dynamic policy is designed to solve the given optimization problem of (9)–(11) by observing the current queue state, $Z(t_k)$, and determining the caching node to minimize a upper bound on frame-based *drift-plus-penalty* [33]:

$$\Delta(t_k) + V\mathbb{E}\left[\sum\nolimits_{t=t_k}^{t_k+T-1}(\bar{\mathcal{P}} - \mathcal{P}(q(t))) \cdot M(t)\middle| Z(t_k)\right], \quad (17)$$

where V is an importance weight for quality improvement.

First of all, the upper bound on the drift can be found in the Lyapunov function.

$$\begin{aligned} L(t+1) - L(t) &= \frac{1}{2}\Big\{Z(t+1)^2 - Z(t)^2\Big\} \\ &= \frac{1}{2}\Big\{\min\{Z(t) - M(t) + c, \tilde{Q} - M(t)\}^2 - Z(t)^2\Big\} \\ &\le \frac{1}{2}\Big\{(Z(t) - M(t) + c)^2 - Z(t)^2\Big\} \end{aligned} \quad (18)$$

By summing (18) over $t = t_k, \cdots, t_k + T - 1$, the upper bound in the frame-based Lyapunov function is obtained by

$$\begin{aligned} &L(t_k + T) - L(t_k) \\ &\quad \le \sum_{t=t_k}^{t_k+T-1}\left\{Z(t)(c - M(t)) + \frac{1}{2}(c - M(t))^2\right\}. \end{aligned} \quad (19)$$

Thus, according to (17), minimizing a bound on frame-based *drift-plus-penalty* is consistent with minimizing

$$\begin{aligned} &\mathcal{D}(\alpha(t_k), Q(t_k), \boldsymbol{q}_k, \boldsymbol{M}_k) = \\ &\quad \mathbb{E}\Bigg[\sum\nolimits_{t=t_k}^{t_k+T-1}\Big\{Z(t)(c - M(t)) + \frac{1}{2}(c - M(t))^2 \\ &\qquad + V(\bar{\mathcal{P}} - \mathcal{P}(q(t))) \cdot M(t)\Big\}\Big| Z(t_k)\Bigg], \end{aligned} \quad (20)$$

where $\boldsymbol{q}_k = [q(t_k), q(t_k + 1), \cdots, q(t_k + T - 1)]$, $\boldsymbol{M}_k = [M(t_k), M(t_k + 1), \cdots, M(t_k + T - 1)]$ and recall that $Z(t) = \tilde{Q} - Q(t)$. The above minimum is conditioned on $M(t)N_{q(t)} \le B(\alpha(t_k), t)$ for all $t \in \mathcal{T}_k$. This frame-based algorithm is shown to satisfy the queue stability constraint of (10) while minimizing the objective function of (9) in [26]. For any $\alpha(t_k) \in \mathcal{A}(t_k)$, the minimum bound on frame-based drift-plus-penalty can be obtained by

$$\mathcal{D}(\alpha(t_k), Q(t_k)) = \min_{\boldsymbol{q}_k, \boldsymbol{M}_k} \mathcal{D}(\alpha(t_k), Q(t_k), \boldsymbol{q}_k, \boldsymbol{M}_k). \quad (21)$$

In Section V, we will provide an efficient method to find the minimum achieving $\boldsymbol{q}_k$ and $\boldsymbol{M}_k$.

System parameter V in (20) is a weight factor for the term representing the measure of video quality degradation. The value of V is important to control the queue backlogs and quality measures at every time. The appropriate initial value of V needs to be obtained by experiment because it depends on the distribution of caching nodes, the channel environments, the playback rate c, and $\tilde{Q}$. Also, $V \ge 0$ should be satisfied. If $V < 0$, the optimization goal is converted into maximizing the measure of video quality degradation. Moreover, in the case of $V = 0$, the user only aims at stacking queue backlogs without consideration of video quality. On the other hand, when $V \to \infty$, users do not consider the queue state, and thus they just pursue to minimize the video quality degradation. V can be regarded as the parameter to control the trade-off between quality and delay, which captures the fact that the user can stack many low-quality chunks or relatively the small number of high-quality chunks in the queue, under the given channel condition. Even though the weight factor V is utilized for Lyapunov optimization as a deterministic system parameter, in the system in which the queue length should be smaller than a predetermined threshold, the appropriate value

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

of V can be adaptively chosen depending on the network state. However, since the statistics of the queue length are very difficult to derive, the distribution of V is hard to be mathematically obtained in advance; therefore, the work of [34] dynamically changes V with the list of predetermined values according to the queue length. However, a constant V is used in this paper for simplicity.

From (20), we can anticipate how the algorithm works. When the queue is almost empty, i.e. $Z(t) \simeq \tilde{Q}$, the large arrivals $M(t)$ are necessary for the user not to wait the next chunk. In this case, the user prefers the caching node which can deliver many chunks. On the other hand, when the queue backlogs are stacked enough to avoid playback delay, i.e. $Z(t) \simeq 0$, the user would request the high quality level of $\mathcal{P}(q(t))$ without worrying about playback latency.

With the initial condition of $Q(t_k)$, the user computes $\mathcal{D}(\alpha(t_k), Q(t_k))$ for all $\alpha(t_k) \in \mathcal{A}(t_k)$. Then, the caching node which minimizes $\mathcal{D}(\alpha(t_k), Q(t_k))$ is chosen at the user,

$$\alpha^*(t_k) = \underset{\alpha(t_k)\in\mathcal{A}(t_k)}{\operatorname{argmin}} \mathcal{D}(\alpha(t_k), Q(t_k)). \tag{22}$$

However, the user should estimate the average function value of future queue states $Z(t)$ and decisions of $q(t)$ and $M(t)$ for $t \in \mathcal{T}_k$. For finding (21), the frame-based algorithm is formulated based on MDP [26], and it can be solved by dynamic programming as following section.

V. Decisions on Quality Level and Receiving Chunk Amounts

The goal of this section is to compute $\mathcal{D}(\alpha(t_k), Q(t_k))$, given the associated caching node $\alpha(t_k)$ and initial queue backlogs $Q(t_k)$.

A. Stochastic shortest path problem

According to (20), we can formulate the drift-plus-penalty algorithm of the k-th frame as follows:

$$\{\boldsymbol{q}_k, \boldsymbol{M}_k\} = \underset{q,M}{\operatorname{argmin}} \mathcal{D}(\alpha(t_k), Q(t_k), \boldsymbol{q}, \boldsymbol{M}) \tag{23}$$

$$\text{s.t. } M(t)N_{q(t)} \le B_k(t) \tag{24}$$

$$|q(t) - q(t-1)| \le 1 \tag{25}$$

$$q(t) \in \mathcal{L}_{l(\alpha(t_k))}, \tag{26}$$

where $B_k(t) \triangleq B(\alpha(t_k), t)$. The problem of (23)–(26) is similar to the stochastic shortest path problem based on MDP. In the network model, $B_k(t)$ and $Z(t)$ (i.e., $Q(t)$) are given before making decisions of $q_k(t)$ and $M_k(t)$ at every time t.

The queue backlog $Z(t)$ represents the current state which satisfies the Markov property and $Z(t) \in \mathcal{Z} = \{0, 1, \cdots, \tilde{Q}\}$ because $Q(t)$ is a nonnegative integer and we assume that $Q(t) \le \tilde{Q}$. It is reasonable to set $\tilde{Q}$ be the arbitrarily predefined maximum queue backlog because the queue size is finite in the practical system. In addition, the quality fluctuation constraint (25) sets a limit on the quality decision $q(t)$. Therefore, we can define $\mathcal{S} = \mathcal{Z} \times \mathcal{L}_{l(\alpha(t_k))}$ as the state space of MDP and let the state set $S(t) = \{Z(t), q(t-1)\} \in \mathcal{S}$. The action set of MDP is defined as $\Theta(t) = \{M(t), q(t)\}$. Then, incurred cost at $t \in \mathcal{T}_k$ can be formulated by

$$g_k(S(t), \Theta(t)) = Z(t)(c - M(t)) + \frac{1}{2}(c - M(t))^2 + V(\bar{\mathcal{P}} - \mathcal{P}(q(t)))M(t). \tag{27}$$

The transition probabilities from $S(t)$ to $S(t+1)$ can be defined for all states s and s' as

$$P_{s's}(\theta, b) = P\{S(t+1) = s' | S(t) = s, \Theta(t) = \theta, B_k(t) = b\}. \tag{28}$$

Since the next state $S(t+1)$ is deterministic given the current state $S(t)$ and action $\Theta(t)$, it can be seen that $P_{s's}(\theta, b) \in \{0, 1\}$. Note that even though transition probabilities are deterministic to the current state and action set, a random event, i.e., channel gain information, sets a limit on the feasible next action space; therefore, the system is not deterministic.

B. Probability mass function of $B_k(t)$

The constraint (24) indicates that the maximum number of chunks which the user can receive depends on the random network event $B_k(t)$ and the decision of quality $q(t)$. It notifies that decisions on $q(t)$ and $M(t)$ should jointly made as well as the probability distribution of the random network event $B_k(t)$ is required.

Define a random variable $Y = \log_2(1 + aX)$, where X is a chi-square random variable and a is a constant. Then, we can obtain $P\{Y \ge y\}$, as given by

$$P\{Y \ge y\} = \exp\Big\{ -\frac{1}{2a}(2^y - 1)\Big\}. \tag{29}$$

Since $|h(\alpha, t)|^2$ follows the chi-squared distribution and a random variable $B_k(t)$ is a nonnegative integer, the probability mass function of $B_k(t)$ is found as follows:

$$P\{B_k(t) = b\} = P\{t_c R(\alpha(t_k), t) \ge b\} - P\{t_c R(\alpha(t_k), t) \ge b + 1\} \tag{30}$$

$$= e^{\frac{\Upsilon+1}{2\Psi D(\alpha,t)}} \left\{ \exp\left\{ -\frac{(\Upsilon+1)2^{b/t_c\mathcal{W}}}{2\Psi D(\alpha,t)} \right\} - \exp\left\{ -\frac{(\Upsilon+1)2^{(b+1)/t_c\mathcal{W}}}{2\Psi D(\alpha,t)} \right\} \right\}. \tag{31}$$

C. Dynamic programming

Given $\alpha(t_k)$ and $S(t_k)$, the user observes the state $S(\tau)$ and the random network event $B_k(\tau)$, and decides the action $\Theta(\tau)$ for each time slot $\tau \in \mathcal{T}_k$. Then, the minimum incurred cost based on measurements of $B_k(\tau)$ and $Z(\tau)$ is

$$G_k(\tau, s_0, b_0) = \min_{\Theta} \mathbb{E}\left[\sum_{t=\tau}^{t_k+T-1} g_k(S(t), \Theta(t)) \Big| S(\tau) = s_0, B_k(\tau) = b_0 \right], \tag{32}$$

conditioned on $M(\tau)N_{q(\tau)} \le B_k(\tau)$, where $s_0 \triangleq \{z_0, q_0\}$ and $S(\tau) = s_0$ means that $Z(\tau) = z_0$ and $q(\tau - 1) = q_0$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

Let $J_k(\tau, s_0)$ be the marginalized function of $G_k(\tau, s_0, b_0)$ over all possible $B_k(\tau) = b_0$, and it can be approximated into

$$J_k(\tau, s_0) = \sum_{b_0=0}^{B_{\max}} P\{B_k(\tau) = b_0\} G_k(\tau, s_0, b_0), \tag{33}$$

where $B_{\max}$ is a nonnegative integer such that $P\{B \geq B_{\max}\} \approx 0$. The dynamic programming provides the action that minimizes the following cost as given by [27]

$$G_k(\tau, s_0, b_0) = \min_{\Theta} \mathbb{E}\Big[g_k(S(\tau) = s_0, \Theta(\tau)) + \sum_{y \in \mathcal{S}} P_{y,s_0}(\Theta(\tau), b_0) \cdot G_k(\tau + 1, y, B_k(\tau + 1))\Big] \tag{34}$$

$$= \min_{\Theta}\Big[g_k(S(\tau) = s_0, \Theta(\tau)) + J_k(\tau + 1, S(\tau + 1))\Big], \tag{35}$$

where the expectation of (34) is with respect to $\{B_k(t) : \tau + 1 \leq t \leq T - 1\}$, $Z(\tau + 1) = \min\{z_0 + c, \tilde{Q}\} - M(\tau)$, and $|q(\tau + 1) - q_0| \leq 1$. The minimum cost is obtained over all $\Theta(\tau)$ such that $M(\tau)N_{q(\tau)} \leq B_k(\tau) = b_0$.

Given $B_k(t) = b_0$, the user can find the minimum value of (35) by greedily testing all joint combinations of decisions on $q(t)$ and $M(t)$. For example, let there exists $L = 2$ quality levels and $q \in \{1, 2\}$ correspond to the file size of $N \in \{10, 20\}$. If $B_k(t) \in [20{:}30)$ Kbits and $q(t - 1) = 1$ for simplicity, where $[a{:}b) \triangleq \{a, a + 1, \cdots, b - 1\}$, then there are four possible decisions: 1) $M(t) = 0$, 2) $q(t) = 1, M(t) = 1$, 3) $q(t) = 1, M(t) = 2$, and 4) $q(t) = 2, M(t) = 1$. The user computes costs for all those possible decision cases and picks the minimum one as an optimal cost.

We set the end time slot of the k-th frame as $t_{k+1} = t_k + T$, which is the start time of the $k+1$-th frame. To find the optimal costs $\mathbf{J}_k(t) = \{J_k(t, s) : s \in \mathcal{S}\}$ for $t \in \mathcal{T}_k$ by using dynamic programming equation (35), the end costs of $\mathbf{J}_k(t_{k+1})$ are required. Since the playback delay occurs at the end state when the accumulated chunk amounts are smaller than the departure quantity, i.e., $Q(t_{k+1}) < c$ and $Z(t_{k+1}) > \tilde{Q} - c$. Therefore, the end costs for those states, i.e., $J_k(t_{k+1}, \{z, q\})$ for $z \in \{\tilde{Q} - c + 1, \cdots, \tilde{Q}\}$ should be very large. Even when $Q(t_{k+1}) \geq c$, the more chunks are accumulated, the more likely there will be no playback delays in the following frame. In this sense, $J_k(t_{k+1}, \{i, q\}) \geq J_k(t_{k+1}, \{j, q\})$ for $i \geq j$ is preferred for all $i, j \in \{0, \cdots, \tilde{Q} - c\}$. Especially, as a large number of chunks are received in the queue, the effect of additional chunks to avert queue emptiness would be significantly decreased. Therefore, $J_k(t_{k+1}, \{i, q\})$ for $i = \{0, 1, \cdots \tilde{Q}\}$ are arbitrarily modeled as the truncated form of exponential distribution. Thus, we can set the end costs for all $q \in \mathcal{L}_{l(\alpha(t_k))}$ as follows:

$$J_k(t_{k+1}, \{q, z\}) = A, \ \forall z \in \{\tilde{Q} - c + 1, \cdots, \tilde{Q}\} \tag{36}$$

$$J_k(t_{k+1}, \{q, i\}) = 10^{-3} \cdot A\mu e^{-\mu \cdot (\tilde{Q} - i)}, \ \forall i \in \{0, \cdots, \tilde{Q} - c\}, \tag{37}$$

where A is a predefined large constant to give penalties for playback delay occurrences and μ is the exponential distribution coefficient.

Given the end costs $\mathbf{J}_k(t_{k+1})$, the optimal costs $\mathbf{J}_k(t)$ for all $t \in \mathcal{T}_k$ can be obtained by backtracking the shortest path based on the dynamic programming equation (35). Then, when the queue backlog at time $t = t_k$ is $Z(t_k)$ and the previous quality selection is $q(t_k - 1)$, $J_k(t_k, S(t_k))$ becomes the minimum drift-plus-penalty term given $\alpha(t_k)$ and $Q(t_k)$, i.e., (21). Then, after finding the averaged drift-plus-penalty terms for all $\alpha \in \mathcal{A}(t)$ by using dynamic programming, the user determines the caching node to receive the desired video file by comparing all drift-plus-penalty terms, as described in (22).

D. Decisions of quality and chunk amounts

After determining the caching node $\alpha(t_k)$, the user should choose the video quality and the number of chunks to receive for every time slot $t \in \mathcal{T}_k$, depending on time-varying channel conditions and its queue state. For this goal, we can simply use the principle of optimality in the dynamic programming algorithm [27], which argues that if the optimal policy $\Theta^* = \{\Theta^*(t_k), \cdots, \Theta^*(t_k + T - 1)\}$ is a solution of the stochastic shortest path problem, then the truncated policy $\{\Theta^*(t_k + j), \cdots, \Theta^*(t_k + T - 1)\}$ is optimal for the subproblem over $t \in \{t_k + j, \cdots, t_k + T - 1\}$, where $0 \leq j \leq T - 1$.

Based on this principle of optimality, the user can make the optimal decisions of $q^*(t)$ and $M^*(t)$ for $t \in \mathcal{T}_k$ by using the minimum costs obtained while performing dynamic programming for caching node decision $\alpha^*(t_k)$. When deciding $q^*(t)$ and $M^*(t)$, the channel gain can be observed, e.g. $B(\alpha(t_k), t) = b_0$; therefore, the optimal action $\Theta^*(t)$ is deterministic given $S(t) = s_0$ and $B_k(t) = b_0$ which provide the minimum cost $G_k(t, s_0, b_0)$, as given by

$$\Theta_k^*(t, s_0, b_0) = \underset{\Theta}{\operatorname{argmin}} \Big[g_k(S(t) = s_0, \Theta) + J_k(t + 1, S(t + 1))\Big], \tag{38}$$

conditioned on $M(t)N_{q(t)} \leq b_0$ for $t \in \mathcal{T}_k$.

Thus, the user should store the optimal actions $\Theta_k^*(t, s, b)$ for all $t \in \mathcal{T}_k$, $s \in \mathcal{S}$ and $b \in \mathcal{B} = \{0, 1, \cdots, B_{\max}\}$ to deal with all possible random network events. Simply, $T \cdot \tilde{Q}L \cdot (B_{\max} + 1)$ actions are required, but some of channel realizations can give the same optimal action. Again, consider the example of $L = 2$ quality levels and $q \in \{1, 2\}$ corresponding to the file size of $N \in \{10, 20\}$ in Kbits. When $q(t - 1) = 1$, any $B = b \in [20{:}30)$ Kbits allows four combinations of decisions of $q(t)$ and $M(t)$, as explained in Section V-C, and the user is enough to store the only one optimal action for all $B = b \in [20{:}30)$ Kbits. In this sense, define $\mathcal{N}_B$ subsets of $\mathcal{B}$ denoted by $\mathcal{B}_n$ for $n \in \{1, \cdots, \mathcal{N}_B\}$, as follows:

$$\bigcup_{n=1}^{\mathcal{N}_B} \mathcal{B}_n = \mathcal{B} \tag{39}$$

$$\mathcal{B}_n \cap \mathcal{B}_m = \phi, \ \forall n \neq m, \ n, m \in \{1, \cdots, \mathcal{N}_B\} \tag{40}$$

$$\Theta_k^*(t, s, b_1) = \Theta_k^*(t, s, b_2), \ \forall b_1, b_2 \in \mathcal{B}_n. \tag{41}$$

Thus, the user needs to store $T \cdot \tilde{Q}L \cdot \mathcal{N}_B$ actions. The whole steps for video delivery decisions on caching node, video quality, and receiving chunk amounts are presented in Algorithm 1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

Algorithm 1 Dynamic video delivery decisions on $\boldsymbol{\alpha}$, $\boldsymbol{q}$, and $\boldsymbol{M}$ in different timescales

Precondition:

- V: parameter for streaming quality-delay trade-offs
- $\tilde{Q}$: threshold for queue backlog size
- K: the number of caching node decisions
- T: the time interval of updating caching node decision

$t = 0$ // $KT - 1$: number of discrete-time operations
while $k \leq K$ **do**
  $t_k = (k-1)T$: time for the k-th caching node decision
  Observe $S(t_k)$ and find $\mathcal{A}(t_k)$
  Compute $\mathcal{D}_k(\alpha(t_k), Q(t_k))$ by using dynamic programming equation (35) and store $\Theta_k^*(t, s, b)$ for every $\alpha(t_k) \in \mathcal{A}(t_k)$, $s \in \mathcal{S}$ and $b \in \mathcal{B}$.
  Make a decision of $\alpha^*(t_k)$ by using (22)
  for $t = t_k : t_k + T - 1$ **do**
    Observe $S(t)$ and $B_k(t)$
    Make a decision of $\Theta_k^*(t, S(t), B_k(t))$
  end for
end while

E. Computational complexity of dynamic programming

To determine the optimal policy at each time slot, it seems that at least $\tilde{Q}LB_{\max}$ computations are required, but some of channel realizations can perform the same computation as seen in Section V-D. Since all realizations $b \in \mathcal{B}_n$ not only give the same $\Theta_k^*(t, S(t), b)$ but also make the same computations of $g_k(S(t) = s_0, \Theta(t)) + J_k(t+1, S(t+1))$ for all possible combinations of $q(t)$ and $M(t)$, $\tilde{Q}L\mathcal{N}_B$ computations are required at least.

However, in most of random network events, more computations are required to take the minimum function in (35). As shown in the example of $L = 2$ quality levels and $q \in \{1, 2\}$ corresponding to the file size of $N \in \{10, 20\}$ in Kbits, there are four combinations of decisions of $q(t)$ and $M(t)$ when $B = b \in [20{:}30)$ Kbits and $q(t-1) = 1$. Generally speaking, there are $\sum_{q=1}^{L} \lfloor \frac{b}{N_q} \rfloor + 1$ combination of possible decisions of $q(t)$ and $M(t)$ for any $b \in \mathcal{B}_n$, when $\mathcal{B}_n$ is given. Denote $\mathcal{N}_\theta(\mathcal{B}_n)$ by the number of decision combinations of $q(t)$ and $M(t)$ when $B_k(t) = b \in \mathcal{B}_n$; then, $\mathcal{N}_\theta(\mathcal{B}_n)$ becomes the size of feasible action sets. Let the average number of $\mathcal{N}_\theta(\mathcal{B}_n)$ for all $n \in \{1, \cdots, \mathcal{N}_B\}$ be $\bar{\mathcal{N}}_\theta$, then total $\tilde{Q}L\mathcal{N}_B\bar{\mathcal{N}}_\theta$ computations are required at each time slot in dynamic programming.

Here, $\mathcal{N}_B$ and $\bar{\mathcal{N}}_\theta$ obviously depend on $B_{\max}$, L and N_q for $q \in \mathcal{L}_L$. There are not many versions of the identical video of different quality levels, i.e. L is small in general, and N_q is not controllable unless the video encoding scheme is changed. On the other hand, $B_{\max}$ increases as transmit SNR grows; therefore, large SNR could result in huge computational complexity as well as large number of registers to store the optimal costs for decisions of quality and chunk amounts. However, the streaming user can receive a large number of high-quality chunks enough to avoid queue emptiness in the sufficiently large transmit SNR region. Considering that the proposed video delivery scheme targets the streaming user who is worrying about playback delays as well as video quality degradation, however, huge complexity burden for large transmit SNR is out of scope in targeting scenarios. Thus, $\mathcal{N}_B$ and $\bar{\mathcal{N}}_\theta$ are expected not much large in our targeting scenarios where adjustments of the tradeoff between playback delay and video quality are necessary; therefore, computational complexity for dynamic programming can be somewhat limited. Nevertheless, the deep reinforcement learning-based approach could relax the curse of dimensionality for the system having massive number of states and actions or no prior channel information. This paper remains the learning-based method for dynamic video delivery as a future work.

TABLE I
SYSTEM PARAMETERS

No. of quality levels (L)	3
Default PPP intensity (λ)	0.4
Time interval of caching node decisions (T)	5
Caching probabilities ($\mathbf{p} = [p_1, \cdots, p_L]$)	$[\frac{4}{7}, \frac{2}{7}, \frac{1}{7}]$
Transmit SNR (Ψ)	25 dB
INR (Υ)	5 dB
Path loss exponent (β)	2
Shadowing variance (σ_w^2)	4 dB
Minimum probability of finding caching node ($\eta_{\min}$)	0.99
Queue departure (c)	1
User radius (R)	50 m
Bandwidth ($\mathcal{W}$)	1 MHz
Coherence time (t_c)	5 ms
End cost coefficient (A)	10^4
End cost coefficient (μ)	1
$\tilde{Q}$	100
$B_{\max}$	52 kbits
V	0.015

VI. SIMULATION RESULTS

In this section, we show that the proposed algorithm for dynamic video delivery works well with video files of different quality levels in wireless caching network with user mobility. Simulation parameters are listed in Table I, and these are used unless otherwise noted. The proposed technique can be applied to any distribution model for caching nodes, but we suppose that caching nodes which store the desired content are modeled as an independent PPP with the intensity of λ, which is generally assumed for researches of wireless caching networks [7], [8]. Then, the PPP intensity of type-l caching nodes becomes λp_l, where p_l denote the caching probability of the video which can be encoded into any quality in $\mathcal{L}_l$. Therefore, larger p_l, more caching nodes of type-l around the streaming user. Based on the network model described in Fig. 1, the user is slowly moving towards certain direction. In practice, the channel condition between the user and the caching node delivering the desired video could be varying due to Doppler shift as the user is moving, but this effect is not captured in this paper. Peak-signal-to-noise ratio (PSNR) is considered as a video quality measure, and quality measures and file sizes depending on quality levels are supposed as $\mathcal{P}(q) = [34,\ 36.64,\ 39.11]$ dB and $N(q) = [2621,\ 5073,\ 10658]$ Kbits which are obtained from real-world video traces [35]. Since we assume that $\eta_{\min} = 0.99$ and $t_0\mathcal{W}\log_2(1 + \gamma_{\min}) = N(1)$, the minimum intensity of PPP distributions to satisfy the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

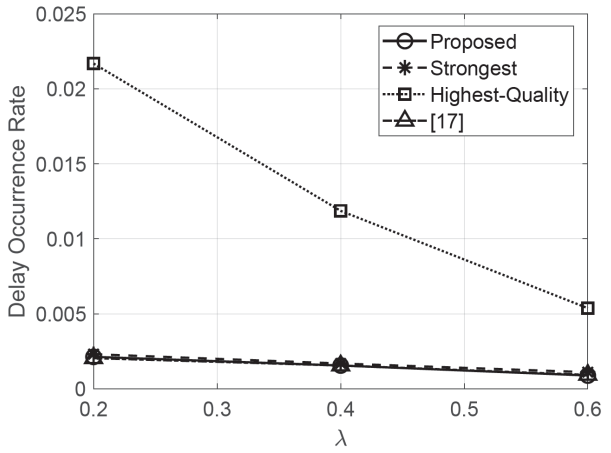

Fig. 4. Delay occurrence rates over λ

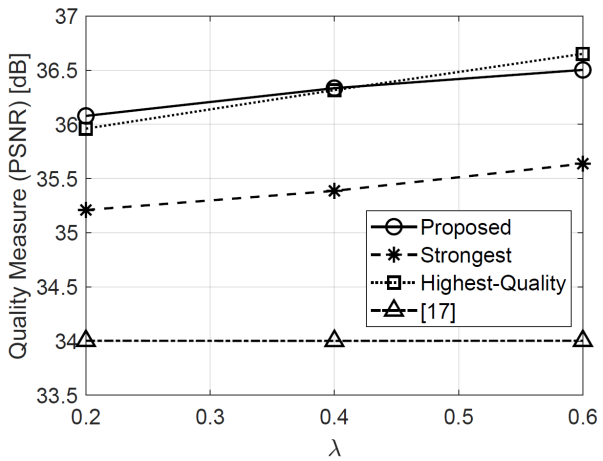

Fig. 5. Video quality measures over λ

performance criterion of $\{\gamma_{\min}, \eta_{\min}\}$ becomes $\lambda_{\min} = 0.0352$. Therefore, even for the least distributed chunks of the highest quality, i.e., $q = 3$, the intensity of caching node distribution is sufficiently large, i.e., $\lambda p_3 = 0.0571 > \lambda_{\min}$.

To verify the advantages of the proposed algorithm, this paper compares the proposed one with three other schemes:

- 'Strongest': The user receives the desired video file from the caching node whose channel condition is the strongest among $\mathcal{A}(t_k)$ at time slots of t_k, for $k \in \{1, 2, \cdots\}$. Decisions of $q(t)$ and $M(t)$ are made based on dynamic programming results.
- 'Highest-Quality': The user receives the desired video file from the caching node which can provide the highest-quality file among $\mathcal{A}(t_k)$ at time slots of t_k, for $k \in \{1, 2, \cdots\}$. Decisions of $q(t)$ and $M(t)$ are made based on dynamic programming results.
- The work of [16]: The user associates with one of nearby caching nodes for content delivery by maximizing the average of quality levels chosen at every time slot under the constraint of playback delay. This work considered caching of video files having a specific quality level only; therefore, the node association is equivalent to the quality decision. Even though two different timescales are investigated for video delivery in this paper, we let the user with this scheme change the caching node to receive video chunks in short-term scale. Note that the performance metric of [16] is the average of quality levels chosen at every time slot without adjustments of receiving chunk amounts, and this scheme lets the user to receive as many chunks as the channel capacity allows.

In summary, performance comparisons with 'Strongest' and 'Highest-Quality' can show the effects of caching node decision based on Lyapunov optimization. Also, comparison with [16] can show that the proposed scheme is appropriate for the practical performance metric (i.e., average video quality per played chunk) and specify the advantage of using MDP and dynamic programming for decisions on video quality and the amounts of receiving chunks.

A. Caching node distribution

At first, impacts of the PPP intensity, i.e. how many caching nodes are distributed around the streaming user, are shown in Figs. 4 and 5, which give the plots of playback delay occurrence rates and average video quality measures per received chunk versus λ, respectively. As λ grows, there becomes an increase chance that the user can find the caching node around itself that can deliver many high-quality chunks; therefore, both delay and quality performances of all schemes become improved.

Since the caching node whose channel condition is the strongest can deliver the largest number of chunks to the user, 'Strongest' provides much lower delay occurrence rates than 'Highest-Quality' in Fig. 4. However, 'Strongest' does not consider the video quality when associating with the caching node; therefore, its quality measure is worse than 'Highest-Quality'. On the other hand, 'Highest-Quality' can provide the higher quality measure to the user compared to 'Strongest', however, the user can suffer from frequent playback delays especially when λ is small.

Meanwhile, the proposed technique determines to associate with the caching node by balancing the video quality and channel condition; therefore, the proposed one can provide quite high quality as 'Highest-Quality' while limiting playback delays as 'Strongest' and the scheme in [16], as shown in Figs. 4 and 5, respectively. Thus, the proposed scheme can be said to smooth out the tradeoff between quality and playback delay and to fairly achieve both goals. Also, note that 'Strongest' and 'Highest-Quality' determines the caching node depending on their purposes, i.e., the channel condition and quality of cached contents, without consideration of user mobility. Even though the channel condition of the caching node chosen by 'Strongest' is the strongest at t_k, after that it could not be the strongest due to time-varying channels and user mobility. Therefore, 'Strongest' is not the best choice for reducing playback delays, and the proposed one and the work of [16] can show almost the same delay performance as 'Strongest'. Similarly, with the knowledge of user mobility and estimated the future decisions on quality and receiving chunk amounts,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

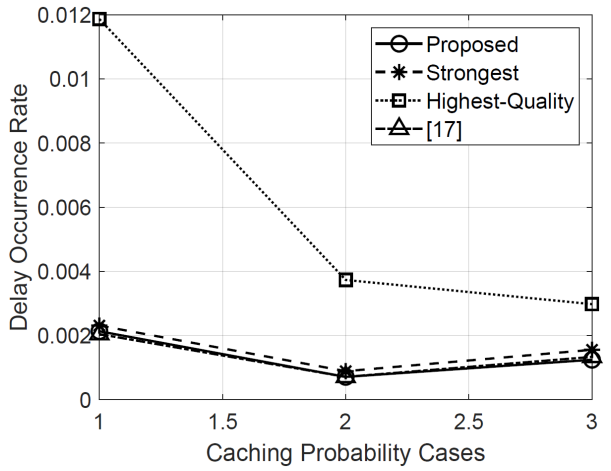

Fig. 6. Delay occurrence rates over caching policies

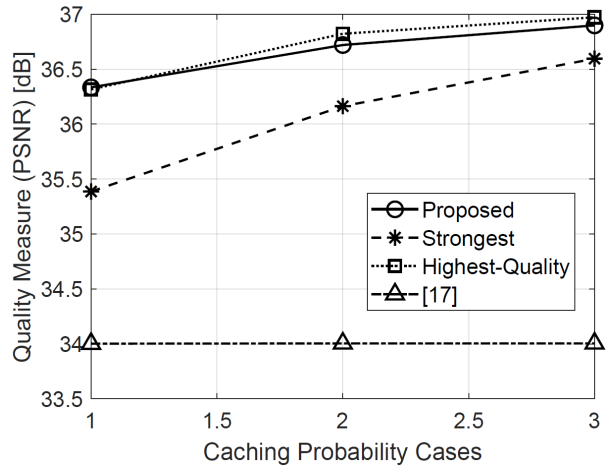

Fig. 7. Video quality measures over caching policies

the proposed scheme provides as high quality measure as 'Highest-Quality'.

The scheme proposed in [16] aims at limiting the playback delay using the same constraint of queue stability in (15) based on Lyapunov optimization; therefore, its delay performance is as good as 'Strongest' and the proposed one. However, the average quality measure per received chunk maintains the lowest value. Since maximization of the average of quality levels chosen at every slot in [16] treats receiving one chunk and multiple chunks of the same quality as the identical case, the actual playback quality could be significantly degraded when the user receives very many low-quality chunks to avoid the playback delay if the channel capacity is sufficiently large. In practice, the average video quality per playing chunk is more appropriate for a performance metric representing the users' satisfactions; therefore, we can see from the results in Fig. 5 that adjustments of receiving chunk amounts are very important.

B. Uniform and nonuniform caching probabilities

We set three cases of caching probabilities for the video file with different quality levels, as follows:

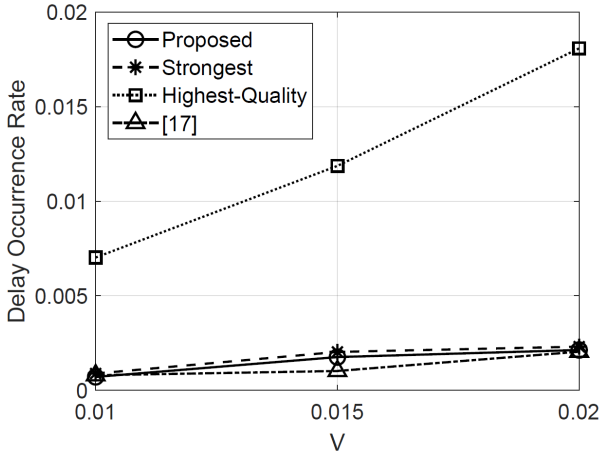

Fig. 8. Delay occurrence rates over V

- Case 1: $p_1 = 4/7,\ p_2 = 2/7,\ p_3 = 1/7$
- Case 2: $p_1 = 1/3,\ p_2 = 1/3,\ p_3 = 1/3$
- Case 3: $p_1 = 1/7,\ p_2 = 2/7,\ p_3 = 4/7$

Note that Case 2 corresponds to the uniform caching probability case and Case 1 and Case 3 are nonuniform. In Case 3, the streaming user is more likely to receive high-quality video than other cases, on the other hand, Case 1 represents an environment where there are not many caching nodes which can provide the high-quality video around the user. The performances of playback delay and quality measure depending on those cases of caching probabilities are shown in Figs. 6 and 7, respectively.

In Fig. 6, delay incidence of 'Highest-Quality' definitely decreases as p_1 decreases and p_3 grows, because the caching nodes storing the high-quality video are likely to be near to the streaming user. However, since the distribution density of all caching nodes does not change, i.e., $\lambda \sum_{l=1}^{L} p_l = \lambda$, the delay performances of all the other schemes are not influenced much by different caching policies. For the 'Strongest' scheme, any caching probability case can deliver as many low-quality chunks of small size as possible when there are too few chunks in queue so the playback delay is about to occur. In this sense, the proposed technique and [16] show almost the same delay performances as 'Strongest', because both schemes strongly limit the playback delay compared to quality improvement.

The average quality measures of all schemes increase as p_1 decreases and p_3 grows as shown in Fig. 7. Even though 'Highest-Quality' pursues the video quality, its average quality measure per received chunk does not differ much from that of the proposed one for any caching probability case similar to Fig. 5. Especially in Case 3, caching nodes storing the highest-quality video are distributed more than nodes of other types, therefore the caching node whose channel condition is the strongest would be highly probable to be type 3. Thus, the difference between quality performances of 'Strongest' and 'Highest-Quality' or the proposed scheme is not large.

The performance rankings in Figs. 6 and 7 among comparison techniques are consistent with the results of Figs. 4 and 5. Compared to those comparison schemes, the proposed technique provides quite high average video quality, while

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

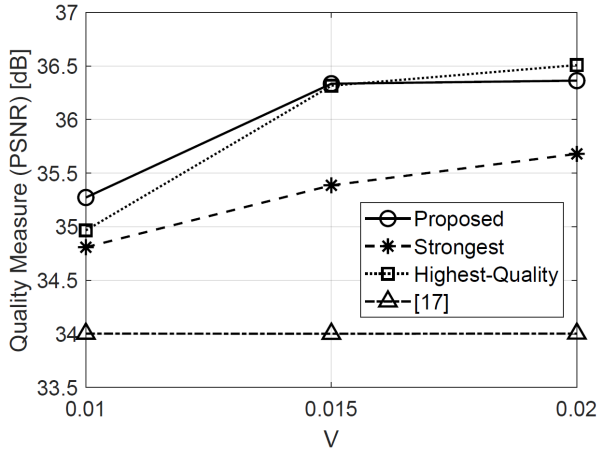

Fig. 9. Video quality measures over V

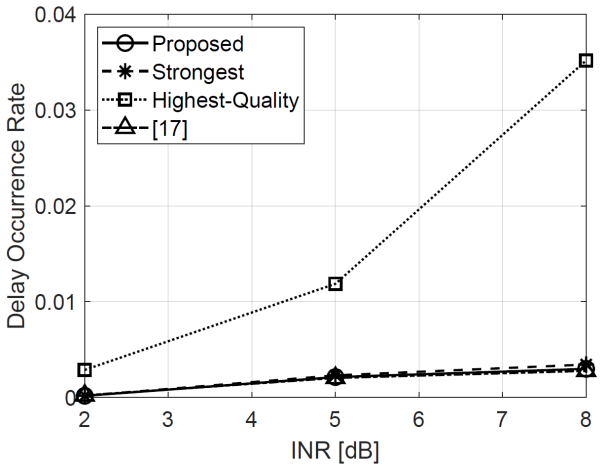

Fig. 10. Delay occurrence rates over Υ

limiting delay occurrence rate as low as 'Strongest' and [16]. In addition, [16] still provides as low delay occurrence rates as the proposed one but its average quality per chunk is too poor to achieve user satisfaction.

C. System parameter V

Since V has a role to weigh quality maximization compared to averting playback delay in Lyapunov optimization problem, delay occurrence rates increase and the expected quality measures of all techniques become improved, as V grows, as shown in Figs. 8 and 9, respectively. Therefore, we can control the tradeoff between video quality and playback latency by adjusting the system parameter V. Among comparison techniques, the proposed scheme improves the quality performance sufficiently while minimizing the increase in delay incidence by taking large V. As we've seen in Sections VI-A and VI-B, the proposed technique provides higher average video quality than 'Strongest' and [16], and much better delay performance than 'Highest-Quality'.

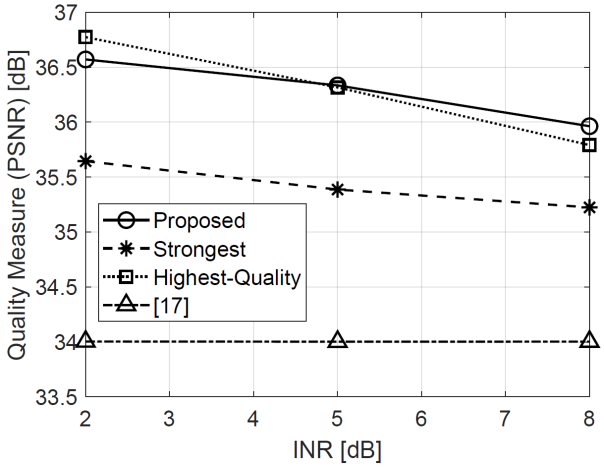

Fig. 11. Video quality measures over Υ

D. SINR level

The delay and quality performances over INR levels are shown in Figs. 10 and 11, respectively. It is easily expected that quality performances decrease and delay occurrence rates increase as INR grows for all comparison techniques. Almost all of the performance rankings among comparison techniques remain as seen former subsections, but the performance of 'Highest-Quality' is influenced by INR levels more than the proposed one and 'Strongest'. We can expect that 'Highest-Quality' becomes more difficult to accumulate video chunks in the queue as the INR grows, therefore the quality level chosen by the user becomes increasingly degraded. Rather, 'Strongest' is not significantly affected by INR changes compared to 'Highest-Quality', because the channel condition of its caching node is much stronger than that of the node chosen by 'Highest-Quality'. The proposed scheme still achieves the improved video quality while guaranteeing very low delay occurrence rate.

VII. Conclusion

This paper studies the dynamic delivery policy of video files of various quality levels in the wireless caching network. When the caching node distribution around the streaming user is varying, e.g. the user is moving, the streaming user makes decisions on caching node to receive the desired file, video quality, and the number of receiving chunks. The different timescales are considered for the caching node association and decisions on quality and the number of receiving chunks. The optimization framework of those video delivery decisions conducted on different timescales is constructed based on Lyapunov optimization theory and Markov decision process. By using dynamic programming and the frame-based drift-plus-penalty algorithm, the dynamic video delivery policy is proposed to maximize average streaming quality while limiting playback delay quite low. Further, the proposed technique can adjust the tradeoff between performances of video quality and playback delay by controlling the system parameter of V.

REFERENCES

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021 White Paper", Cisco. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[2] X. Cheng, J. Liu, and C. Dale, "Understanding the characteristics of Internet short video sharing: A YouTube-based measurement study," *IEEE Trans. Multimedia*, 15(5):1184–1194, Aug. 2013.

[3] J. Koo, J. Yi, J. Kim, M. A. Hoque, and S. Choi, "REQUEST: Seamless dynamic adaptive streaming over HTTP for multi-homed smartphone under resource constraints," in *Proc. of ACM Multimedia*, Mountain View, CA, 2017.

[4] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," in *Proc. of IEEE INFOCOM*, Orlando, FL, USA, 2012.

[5] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, 52(8):82–89, Aug. 2014.

[6] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, 52(2):131–139, Feb. 2014.

[7] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *Proc. of IEEE Int'l Conf. Communi. (ICC)*, London, 2015, pp. 3358–3363.

[8] S. H. Chae and W. Choi, "Caching placement in stochastic wireless caching helper networks: Channel selection diversity via caching," *IEEE Trans. Wireless Communi.*, 15(10):6626–6637, Oct. 2016.

[9] M. Choi, D. Kim, D.-J. Han, J. Kim, and J. Moon, "Probabilistic Caching Policy for Categorized Contents and Consecutive User Demands," *IEEE International Conference on Communications (ICC)*, 2019.

[10] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Video delivery over heterogeneous cellular networks: Optimizing cost and performance," in *Proc. of IEEE INFOCOM*, Toronto, ON, 2014.

[11] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos and L. Tassiulas, "Caching and operator cooperation policies for layered video content delivery," *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, 2016, pp. 1-9.

[12] C. Yang, Y. Yao, Z. Chen and B. Xia, "Analysis on Cache-Enabled Wireless Heterogeneous Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 131-145, Jan. 2016.

[13] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Trans. Comm.*, 62(10):3665–3677, Oct. 2014.

[14] L. Zhang, M. Xiao, G. Wu, and S. Li, "Efficient scheduling and power allocation for D2D-assisted wireless caching networks," *IEEE Trans. Communi.*, 64(6):2438–2452, Jun. 2016.

[15] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, 16(5):1382–1393, May 2017.

[16] M. Choi, J. Kim and J. Moon, "Wireless Video Caching and Dynamic Streaming Under Differentiated Quality Requirements," *IEEE Journal on Selected Areas in Communications,* vol. 36, no. 6, pp. 1245-1257, June 2018.

[17] X. Wang, M. Chen, T. T. Kwon, L. Yang, and V. C. M. Leung, "AMESCloud: A framework of adaptive mobile video streaming and efficient social video sharing in the clouds", *IEEE Trans. Multimedia*, 15(4):811–820, Jun. 2013.

[18] D. Bethanabhotla, G. Caire, and M. J. Neely, "Adaptive video streaming for wireless networks with multiple users and helpers," *IEEE Trans. Comm.*, 63(1):268–285, Jan. 2015.

[19] J. Kim, G. Caire, and A. F. Molisch, "Quality-aware streaming and scheduling for device-to-device video delivery," *IEEE/ACM Trans. Netw*, 24(4):2319–2331, Aug. 2016.

[20] J. Yang, P. Si, Z. Wang, X. Jiang and L. Hanzo, "Dynamic Resource Allocation and Layer Selection for Scalable Video Streaming in Femtocell Networks: A Twin-Time-Scale Approach," *IEEE Trans. on Commun.*, vol. 66, no. 8, pp. 3455–3470, Aug. 2018.

[21] T. Stockhammer, "Dynamic adaptive streaming over HTTP - standards and design principles," *Proc. ACM MMSys2011*, California, Feb. 2011.

[22] J. Qiao, Y. He and X. S. Shen, "Proactive Caching for Mobile Video Streaming in Millimeter Wave 5G Networks," *IEEE Trans. on Wireless Commun.*, vol. 15, no. 10, pp. 7187-7198, Oct. 2016.

[23] R. Wang, J. Zhang, S. H. Song and K. B. Letaief, "Mobility-aware caching in D2D networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5001-5015, Aug. 2017.

[24] J. Song and W. Choi, "Mobility-Aware Content Placement for Device-to-Device Caching Systems," Early Access, *IEEE Trans. on Wireless Commun.*, May 2019.

[25] J. Kwak, L. B. Le, H. Kim and X. Wang, "Two Time-Scale Edge Caching and BS Association for Power-Delay Tradeoff in Multi-Cell Networks," Early Access, *IEEE Trans. on Commun.*, Apr. 2019.

[26] M. J. Neely and S. Supittayapornpong, "Dynamic Markov decision policies for delay constrained wireless scheduling," *IEEE Trans. Automatic Control*, 58(8):1948–1961, Aug. 2013.

[27] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 4th Ed., Athena Scientific, 2017

[28] Z. Chen, J. Lee, T. Q. S. Quek and M. Kountouris, "Cooperative Caching and Transmission Design in Cluster-Centric Small Cell Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3401-3415, May 2017.

[29] Z. Qin, X. Gan, L. Fu, X. Di, J. Tian and X. Wang, "Content Delivery in Cache-Enabled Wireless Evolving Social Networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6749-6761, Oct. 2018.

[30] R. Etkin, A. Parekh and D. Tse, "Spectrum sharing for unlicensed bands," *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.,* Baltimore, MD, USA, 2005, pp. 251-258.

[31] J. Blomer and N. Jindal, "Transmission Capacity of Wireless Ad Hoc Networks: Successive Interference Cancellation vs. Joint Detection," *2009 IEEE International Conference on Communications*, Dresden, 2009, pp. 1-5.

[32] D. Bertsekas and R. Gallager, *Data Networks*, 2nd Ed., Prentice, 1992.

[33] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan & Claypool, 2010.

[34] M. Choi, J. Kim and J. Moon, "Adaptive Detector Selection for Queue-Stable Word Error Rate Minimization in Connected Vehicle Receiver Design," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3635-3639, April 2018.

[35] J. Kim and E. S. Ryu, "Feasibility study of stochastic streaming with 4K UHD video traces," in *Proc. IEEE Int'l Conf. Information and Communi. Technology Convergence (ICTC)*, 2015, pp. 1350-1355.

Minseok Choi is a postdoctoral researcher in electrical and computer engineering at the University of Southern California (USC), Los Angeles, CA, USA, and a postdoctoral associate in school of software at the Chung-Ang University, Seoul, Korea. He received the B.S., M.S. and Ph.D. degrees in the School of Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2011, 2013, and 2018, respectively. His research interests include wireless caching networks, stochastic network optimization, non-orthogonal multiple access, and 5G networks.

Albert No received a B.Sc. in both Electrical Engineering and Mathematics from Seoul National University, in 2009, and a M.Sc. and Ph.D. in Electrical Engineering from Stanford University in 2012 and 2015 respectively. From 2015 to 2017, he was a Data Scientist at Roche. In 2017, he joined Hongik University, Seoul, Korea, where he is an Assistant Professor of Electronic and Electrical Engineering. His research interests include the relation between information and estimation theory, lossy compression, and bioinformatics.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2938755, IEEE Transactions on Wireless Communications

Mingyue Ji (S'09–M'15) received the B.E. in Communication Engineering from Beijing University of Posts and Telecommunications (China), in 2006, the M.Sc. degrees in Electrical Engineering from Royal Institute of Technology (Sweden) and from University of California, Santa Cruz, in 2008 and 2010, respectively, and the PhD from the Ming Hsieh Department of Electrical Engineering at University of Southern California in 2015. He subsequently was a Staff II System Design Scientist with Broadcom Corporation (Broadcom Limited), where he made a number of contributions for the standardization of the next generation of WLAN systems (802.11ax), in 2015-2016. He is now an Assistant Professor of Electrical and Computer Engineering Department at University of Utah. He received the best paper award in IEEE International Conference on Communications (ICC) 2015, the best student paper award in IEEE European Wireless Conference 2010 and USC Annenberg Fellowship from 2010 to 2014. His main interests are in the field of communications and networking, information and coding theory, and statistics with particular focus on caching networks, next generation wireless communications, distributed storage and computing systems, and mobile ad hoc and sensor networks.

Joongheon Kim (M'06–SM'18) has been an assistant professor with Korea University, Seoul, Korea, since 2019. He received his B.S. (2004) and M.S. (2006) in computer science and engineering from Korea University, Seoul, Korea; and his Ph.D. (2014) in computer science from the University of Southern California (USC), Los Angeles, CA, USA. Before joing Korea University as a faculty member, he was with Chung-Ang University (Seoul, Korea, 2016–2019), Intel Corporation (Santa Clara, CA, USA, 2013-–2016), InterDigital (San Diego, CA, USA, 2012), and LG Electronics (Seoul, Korea, 2006–2009). He is a senior member of the IEEE; and a member of IEEE Communications Society. He was awarded Annenberg Graduate Fellowship with his Ph.D. admission from USC (2009).