Probabilistic Caching and Dynamic Delivery Policies for Categorized Contents and Consecutive User Demands

Minseok Choi, *Member, IEEE*, Andreas F. Molisch, *Fellow, IEEE*, Dong-Jun Han, *Graduate Student Member, IEEE*, Dongjae Kim, *Member, IEEE*, Joongheon Kim, *Senior Member, IEEE*, and Jaekyun Moon, *Fellow, IEEE*

***Abstract*— Wireless caching networks have been extensively researched as a promising technique for supporting the massive data traffic of multimedia services. Many of the existing studies on real-data traffic have shown that users of a multimedia service consecutively request multiple contents and this sequence is strongly dependent on the related list of the first content and/or the top referrer in the category. This paper thus introduces the notion of "temporary preference", characterizing the behavior of users who are highly likely to request the next content from a certain target category (i.e., related content list). Based on this observation, this paper proposes both probabilistic caching and dynamic delivery policies for categorized contents and consecutive user demands. The proposed caching scheme maximizes the minimum of the cache hit rates for all users. In the delivery phase, a dynamic helper association policy for receiving multiple contents in a row is designed to reduce the delivery latency. By comparing with the content placement optimized for one-shot requests, numerical results verify the effects of categorized contents and consecutive user demands on the proposed caching and delivery policies.**

***Index Terms*— Wireless caching, content delivery, delay-sensitive communications, consecutive user demands, and user preference.**

Manuscript received November 30, 2019; revised April 27, 2020 and September 5, 2020; accepted December 1, 2020. Date of publication December 18, 2020; date of current version April 9, 2021. This work was supported in part by the Institute for Information and Communications Technology Promotion Grant funded by the Korean Government (MSIT) under Grant 2018-0-00170, in part by the Virtual Presence in Moving Objects through 5G, by the National Research Foundation of Korea under Grant NRF-2020R1G1A1101164, and in part by the National Science Foundation under Project CNS-1816699. This article was presented in part at the 2019 IEEE International Conference on Communications. The associate editor coordinating the review of this article and approving it for publication was K. R. Chowdhury. *(Corresponding author: Joongheon Kim.)*

Minseok Choi is with the Department of Telecommunication Engineering, Jeju National University, Jeju 63243, South Korea (e-mail: ejaqmf@jejunu.ac.kr).

Andreas F. Molisch is with the Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90007 USA (e-mail: molisch@usc.edu).

Dong-Jun Han and Jaekyun Moon are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: djhan93@kaist.ac.kr; jmoon@kaist.edu).

Dongjae Kim is with the Major of Electrical and Electronics Engineering, Korea Maritime and Ocean University, Busan 49112, South Korea (e-mail: codong@kaist.ac.kr).

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

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2020.3044076.

Digital Object Identifier 10.1109/TWC.2020.3044076

I. Introduction

At present, tens of exabytes of global data traffic are being handled on a daily basis [1]. In many mobile services, a relatively small part of popular content is requested very frequently, e.g., on-demand streaming services [2]. To deal with these overlapped and repeated user demands, wireless caching technologies have been studied, wherein the core network via the base station (BS) pushes popular contents during off-peak hours to cache-enabled helpers so that these helpers can provide popular contents directly to nearby mobile users [3], [4]. Also, the user devices can be used as helpers, leading to a device-to-device (D2D)-assisted caching network [5]–[7].

In a wireless caching network, there are mainly two issues: 1) *content placement problem* - which contents to be cached at the caching nodes (e.g., caching helpers or cache-enabled devices), and 2) *node association problem* - which caching node is the optimal one to deliver the requested file to the user.

The goal of the content placement is to find the optimal caching policies according to the popularity distribution of contents and network topology. In practice, caching nodes have finite storage sizes due to cost issues. Therefore, the system should determine which content is better to be stored in which caching nodes. There have been several research efforts on probabilistic caching methods for various optimization goals, e.g., maximization of cache hit probability [11], cache-aided throughput [5], average success probability of content delivery [12], throughput under the outage constraint [6], and average successfully enjoyable content quality [13]. However, these previous research results on the content placement problem do not consider consecutive user demands for categorized contents.

In general, content caching is performed before user demands are made; therefore, it is important to determine the appropriate delivery decisions depending on the current caching distribution. Given the cache states of caching helpers, the critical challenge is to identify which caching helper is suitable for transmitting the content; this is called the helper association problem. In most of the existing studies that considered one-shot requests of contents having the same size, it is reasonable for the user to receive the desired

1536-1276 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

content from the caching helper whose channel condition is the strongest [12]. Meanwhile, for situations in which the same content is stored in different helpers with different qualities (and hence different file sizes), [13], [14] proposed dynamic helper associations for the differentiated quality requirements of users. In particular, the proposed delivery policy in [14] adaptively determines the quality level as well as the number of received chunks. Furthermore, BS or helper associations for content delivery in caching networks considering the interference caused by multiple transmitters were presented in [16], [17]. However, the delivery policies in the above studies are optimized for one-shot requests only, assuming that all different content requests are independent of one another.

Recently, the caching and delivery of categorized contents based on user preference have been extensively researched. Ref. [18] supposed that disjoint user groups have different content preferences and optimized the caching probabilities of all user groups. On the other hand, content popularity and user preference are separately defined in [20] and [21], and the authors characterized the relationship between content popularity and user preference. However, the user preference considered in all of the above studies is defined in a global manner, which is denoted as *global preference* in this paper. The *global preference* considered in [18], [20], [21] can be obtained by averaging many content requests for all users during a given period; it is assumed that each request is independent of all the others. In this regard, some learning-based caching methods in [23], [24] estimate the time-varying global popularity profile or global user preference based on multiple request events within the fixed time interval; however, their request events are intermittent and independent of each other.

In multimedia services, e.g., online video services, a user can consecutively request multiple contents and typically has the purpose of consuming contents in a specific category. In this case, the sequence of consecutively requested contents could be highly correlated. This phenomenon has been called as *temporal locality* [25] that characterizes the short-term content popularity different from long-term popularity. In addition, the service platform provides the user the related content list obtained by the recommendation system, and this related list strongly affects the user's next requests [26]. According to the studies on user behaviors based on real data traffic in YouTube [27]–[29], it is highly probable that the user requests the next video from the related list. The authors of [27] showed that the first item contributes to the views of the subsequent videos. In [28], the view rates of each video and the top referrer video whose popularity is the highest in the related list are shown to have a very strong correlation, and the request rates of the videos in the related list is modeled by a Zipf distribution. Here, according to [30], almost 80% of the users consume the limited number of sessions or videos in a specific category. Also, ref. [31] showed that the category of a given content is more influential than its individual properties when the selection is made as a recommended video in YouTube. Accordingly, this paper considers the related content list based on the video category, and supposes that the probability of requesting the content in the given related list is much larger than that of requesting the content outside the related list after the user watches the first video.

In this sense, we define *temporary preference* as the user preference shown in a short sequence of consecutively requested contents. Unlike global preference, temporary preference has the following distinct characteristics: 1) consecutive content requests in a short sequence are not independent, 2) these contents are likely to be in the identical target category or related list, and 3) temporary preference is applied to a given short sequence of contents and disappears after the user stops consuming the content. Consequently, elaborate designs for the caching and delivery of consecutively requested contents need to take into account the temporary preference.

The content popularity for an one-shot request with the assumption that individual requests are independent is commonly modeled as the Zipf distribution [5]. Meanwhile, if a user requests a short sequence of multiple contents continuously, this sequence could have temporary preference. For example, when a user begins to watch a video, the popularity of the first content can be random, e.g., Zipf distribution; however, the popularity of the next video largely depends on its related list, and generally follows a different distribution.

Based on this observation, we presented a probabilistic caching policy for consecutive user demands in wireless caching helper networks in our conference paper [32]; compared to [32], the current paper improves the popularity model for consecutively requested contents. In addition, with the assumption that the user does not know the exact channel state information, the caching problem in this manuscript maximizes the cache hit rate for consecutive content requests, rather than maximization of the successive delivery rate considered in [32]. Another important difference to [32] is that here we provide a dynamic helper association scheme for the delivery phase. Considering the delivery of a sequence of contents, the user sometimes needs to switch the caching helper for receiving the next content; however, a new helper association step consisting of checking the cache state, scheduling request, and request acceptance generally requires a relatively large delay time. In this sense, our helper association policy has the potential to reduce the entire delivery latency when a user consumes multiple contents in a row.

The main contributions of this paper are as follows:

- Unlike most papers on the content placement problem, in which only the one-shot content request and global preference are considered, consecutive content requests and *temporary preference* are considered in this paper. In practice, heavy users continuously consume a sequence of multiple contents that are likely to be in an identical target category. To the best of our knowledge, there has been no research on consecutive user demands and *temporary preference* in wireless caching networks.
- An iterative algorithm for finding the optimal probabilistic caching policy for categorized contents and consecutive user demands is proposed. The proposed caching scheme maximizes the minimum of the successful delivery probabilities of all users. The proposed iterative algorithm can ensure convergence.

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

- A dynamic helper association scheme for content delivery is presented. Considering the additional delay time caused by the switching of the caching helper associations, we propose a decision method to find an appropriate caching helper depending on the caching distribution and channel conditions when receiving multiple contents in a row. This association scheme can be used for any caching scheme.
- The numerical results show the impacts of the temporary preference on caching and helper association policies. The performance gains of the proposed scheme increase as the probability of consecutively requesting contents in the same category increases and the number of consecutive requested contents grows.

The rest of this paper is organized as follows. The system model is described in Section II. The probabilistic caching and the helper association schemes are proposed in Sections III and IV, respectively. Numerical results are shown in Section V, and Section VI concludes the paper.

II. System Model

This section describes the wireless caching network and the content popularity model. We consider a scenario in which users consume multiple contents in sequence, and the contents are grouped into several categories.

A. Wireless Caching Network

This paper considers a cellular model consisting of multiple caching helpers at fixed locations, and the users request a particular cached content from a library $\mathcal{F}$. Suppose that the library $\mathcal{F}$ consists of F contents, i.e., $c_i \in \mathcal{F}$ for all $i \in \{1, \cdots, F\}$, and all the contents have normalized unit sizes. For contents with different sizes, each content can be partitioned into small chunks of the same size and each chunk can be considered as an individual content. Suppose that there are given K categories (which can be interpreted as lists of related contents) in $\mathcal{F}$, denoted by $\mathcal{G}_k$ for $k \in \{1, \cdots, K\}$, where $\bigcup_{k=1}^{K} \mathcal{G}_k = \mathcal{F}$ holds. Here, we note that it is not necessary for categories to be disjoint; therefore, denote the index set of the categories that c_i belongs to by $\mathcal{K}(c_i)$. Also, $\mathcal{G}(c_i) = \bigcup_{k \in \mathcal{I}(c_i)} \mathcal{G}_k$ is the set of the contents in one of the categories that c_i belongs to. The number of contents in $\mathcal{G}_k$ is F_k, i.e., $|\mathcal{G}_k| = F_k$. Caching helpers have a finite storage size of M, which means that only M contents can be cached in each helper. In practice, $F > M$; therefore, the caching helpers cannot store all the contents in $\mathcal{F}$. The set of contents cached in helper α is denoted by $\mathcal{C}_\alpha$.

This paper considers the situation where active users consume multiple contents continuously, and each user can consume different numbers of videos. Therefore, this paper considers L types of users, where a type-l user requests l contents in a row from nearby helpers. We denote the l_0-th requested content of the type-l user by $c^{(l_0)} \in \mathcal{F}$ for any $l_0 \in \{1, \cdots, l\}$. The spatial distributions of the caching helpers and users follow homogeneous Poisson point processes (PPP) with intensities λ and λ_u, respectively. In this study, we utilize the probabilistic caching method [5] for the caching helpers to store file i with probability p_i.

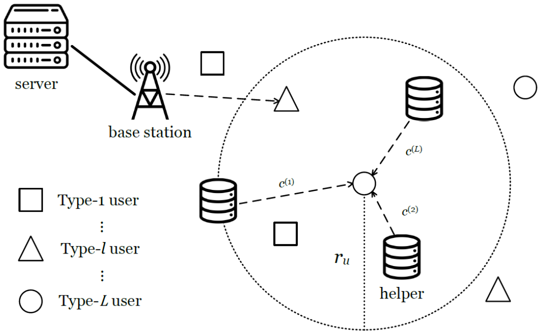


Fig. 1. Wireless caching network model.

TABLE I
System Description Parameters

F	Number of contents
K	Number of categories
$\mathcal{G}_k$	Category k
F_k	Number of contents in category k
M	Cache size
λ	Intensity of Poisson point process of device distribution
$\mathcal{C}_\alpha$	Content set cached in helper α
L	User type (Number of requested contents)
c_i	Content i
$c^{(l)}$	l-th requested content
f_i	Global popularity of content i
$f_{j\|i}$	Temporary popularity of content j when $c^{(1)} = c_i$
ν	Temporary preference probability
p_i	Caching probability of content i
τ_N	Delay time for a switch of helper association
τ_B	Delay time for a backhaul communication

A Rayleigh fading channel is assumed for wireless links from users to caching helpers. The channel is denoted by $h_\alpha = \sqrt{D_\alpha} g$, where $D_\alpha \propto 1/d_\alpha^\beta$ denotes the path gain (the inverse of the path loss) between the user and helper α, d_α and β are the distance between the user and helper α and the path loss exponent, respectively, and g represents a fast fading component with a circularly symmetric complex Gaussian distribution $g \sim CN(0, 1)$. Here, the shadowing effects are ignored for mathematical convenience.

B. Distance-Based Interference Management

In this paper, distance-based interference management is used. Activation of a new delivery link 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 set the pair $\{\rho, \epsilon\}$ to allow or to ban the new link activation, where ρ is the data rate threshold and ϵ is the minimum outage probability. In other words, a new link can be established only when the outage probabilities of all links are smaller than ϵ, where the link outage occurs if the data rate is smaller than ρ. Here, we assume that ρ is sufficiently large so that the content delivery can be successfully completed. Based on this

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

criterion with the pair $\{\rho, \epsilon\}$, we define r_u and r_a as the safety distances for guaranteeing the performances of the new user and the existing active users, respectively.

First, r_u is defined as the maximum distance between the new user and helper α supposed to deliver the content to the new user. The outage probability of the new link is given by

$$\Pr\Big\{\mathcal{B}\log_2\Big(1+\frac{|g|^2}{d_\alpha^\beta\sigma^2(\Upsilon_u+1)}\Big)\le\rho\Big\}, \tag{1}$$

where $\mathcal{B}$ is the bandwidth and Υ_u is the interference-to-noise ratio (INR), assuming a unit transmit power and a normalized noise variance of σ^2. Since $|g|^2$ follows the chi-squared distribution and the outage probability should be smaller than ϵ, we can obtain the following inequality,

$$d_\alpha \le \Big(\frac{2}{\sigma^2(\Upsilon_u+1)(2^{\rho/\mathcal{B}}-1)}\ln\frac{1}{1-\epsilon}\Big)^{\frac{1}{\beta}} = r_u, \tag{2}$$

and we define r_u as the righthand side of (2).

In addition, r_a is defined as the minimum distance between the helper that establishes the new delivery link and any existing active user. With the sufficiently large value of r_a, we can suppose that the interference from outside of the radius r_a could be ignored. In order to determine r_a appropriately, the newly activated delivery link has to inform their locations and interfering power to nearby potential users and helpers. Based on this information, the new user can measure its INR Υ_u in (1). Then, we can give a guideline for activation of a new delivery link by using r_u and r_a. When a user requests the content, it has to find the helper within the radius r_u, and that helper should be at least r_a away from all the existing active users. If these conditions are satisfied, the new delivery link can be established, and all links satisfy the criterion $\{\rho, \epsilon\}$.

Especially, if the density of content-requesting users is not very large compared to that of potential helpers, we can consider the constraint on r_u only. Therefore, we suppose that the content-requesting users are not very densely distributed, in other words, the active links are separated sufficiently not to significantly interfere with each other. A representative example scenario is the D2D-assisted caching network in which all the inactive devices act as the caching helpers and the number of inactive devices is much larger than that of active users [34]. In this scenario, the constraint on r_a is not very hard; accordingly, this paper considers the constraint on r_u only. In addition, if the density of active user distributions is sufficiently smaller than that of potential helper devices, the interference power becomes week and Υ_u could be also ignored in (1). Here, if there is no caching helper that can satisfy the condition of r_u, the new user can request the content from the core network via the BS. In the general content delivery network (CDN), the BS has access to one or more servers in the core network through backhaul connections and the servers have the whole file library.

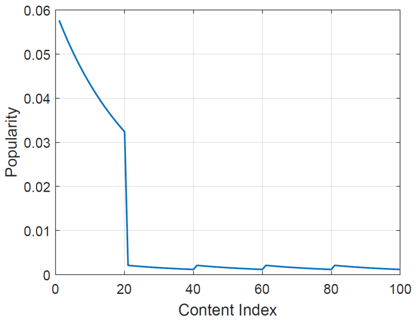


Fig. 2. Popularity of contents given the target category.

C. Content Popularity Model

Unlike most of the existing works on wireless caching, this paper allows each user to consecutively request multiple contents. For example, on-demand streaming users usually have the intention of watching videos in a specific category [30], and the short sequence of the requested contents is strongly correlated. We will call this tendency *temporary preference* in order to distinguish it from *global preference* in [18], [20], [21]. Given the target category (i.e., related list) for a video, we can expect that the probability of requesting any content in the target category is much larger than that of requesting content outside the target category. According to [30], the number of sessions or videos that the user consumes continuously is not very long; therefore, we suppose that the related list is unchanged during consuming multiple contents.

1) First Content Request: When the user begins to request the content, the global content popularity distribution is employed. Each content $c_i \in \mathcal{F}$ has a popularity probability f_i, which follows the M-Zipf distribution [35]: $f_i = (i+q)^{-\gamma}/\sum_{j=1}^{F}(j+q)^{-\gamma}$, where γ and q denote the popularity distribution skewness and the plateau factor, respectively.

2) Consecutive Content Requests: After consuming the first content, temporary preference is applied to the consecutive content requests. Here, since $c^{(1)}$ could belong to multiple categories, the target category of $c^{(1)}$ becomes $\mathcal{G}^{(1)} = \mathcal{G}(c^{(1)})$. The popularity of the following content c_j is denoted by $f_{j|i^{(1)}}$ and $f_{j|i^{(1)}} \neq f_j$, where $i^{(1)}$ is the file index of $c^{(1)}$. In order to characterize the temporary preference, we simply use constant probabilities ν_k which is the probability of requesting the next content in $\mathcal{G}_k$ for all $k \in \mathcal{K}(c^{(1)})$. It satisfies $\sum_{k\in\mathcal{K}(c^{(1)})}\nu_k = \nu$, and the user can request the next content from outside of $\mathcal{G}^{(1)}$ with probability $1-\nu$.

After the user determines the category to request the next content, the category-based conditional popularity distribution of a chosen target category $\mathcal{G}_k$ is modeled as an M-Zipf distribution [35], i.e., $f_{k,i}^{in} = (i+q_k^{in})^{-\gamma_k^{in}}/\sum_{j=1}^{K}(j+q_k^{in})^{-\gamma_k^{in}}$, which represents the popularity of the i-th popular content in $\mathcal{G}_k$ for all $i \in \{1, \cdots, F_k\}$. If ν is large, i.e., $\nu \approx 1$, popularities of contents not belonging to $\mathcal{G}^{(1)}$ are much smaller than that of any content in $\mathcal{G}^{(1)}$. Fig. 2 shows the popularity distribution of 100 contents for a given target category, grouped into 5 exclusive categories consisting of 20 contents. This figure is obtained with $\nu = 0.9$, $\gamma_k^{in} = 2.4$, and $q_k^{in} = 69$. In Fig. 2, contents 1–20 belong to the target category, and their popularity is relatively much larger than that of the others. Therefore, if ν is sufficiently large, we can approximate the popularity of contents outside the target category as a uniform

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

distribution. Consequently, the popularity model of $f_{j|i^{(1)}}$ is assumed as follows:

$$f_{j|i^{(1)}} = \begin{cases} \nu_k \cdot \dfrac{(j+q_k^{in})^{-\gamma_k^{in}}}{\sum_{n=1}^{F_k}(n+q_k^{in})^{-\gamma_k^{in}}} & c_j \in \mathcal{G}_k,\ \forall k \in \mathcal{K}(c^{(1)}) \\ \dfrac{1-\nu}{F-\sum_{k\in\mathcal{K}(c^{(1)})} F_k} & c_j \notin \mathcal{G}_k. \end{cases} \tag{3}$$

It can be seen from (3) that the popularity model for each content depends only on whether the requested content is in $\mathcal{G}^{(1)}$ or not. Therefore, the caching and delivery methods proposed in this paper are not affected by the categorization of the contents.

Remark: Based on the real data traffic and user behaviors in online video services, it is worthy to derive the appropriate values of ν_k in (3). For example, a live performance video belongs to both music and entertainment categories. By investigating user behaviors, relevance of this video to each related category can be measured, e.g., 90% correlation with music and 60% correlation with entertainment. Then, we can notice that ν_k of this video for the music category is larger than that for entertainment. However, investigations and analysis of real data traffic and the exact modeling of the popularity for consecutively requested contents are out of scope of this paper.

III. The Probabilistic Caching Policy Maximizing the Minimum Cache Hit Rate

This section introduces the cache hit rate for consecutive user demands and finds the probabilistic caching policy that maximizes the minimum cache hit rate of all users.

A. The Average Cache Hit Rate of Type-l User

Note that a type-l user requests l contents in a row, and the sufficiently large ρ guarantees the successful content delivery from any caching helper which is at some distance smaller than r_u away from the user. Therefore, a cache hit event occurs for the type-l user if each of the l requested contents can be delivered from any helper within the radius r_u of the user. Based on Slivnyak's theorem, we consider a typical user located at the origin, and the statistics of the typical user represent those of any other user generated by a PPP with the same intensity.

Denote the caching probability of c_i by p_i. Then, the probability that there is no helper caching c_i within the radius of r_u from the user becomes $e^{-\lambda p_i \pi r_u^2}$. Note that λp_i is the intensity of the PPP for the helpers caching c_i. the average cache hit rate of the type-l user is given by

$$P_l^h = \sum_{i^{(1)}} \cdots \sum_{i^{(l)}} f_{i^{(1)}} f_{i^{(2)}|i^{(1)}} \cdots f_{i^{(l)}|i^{(1)}} \cdot \prod_{i=i^{(1)}}^{i^{(l)}} (1-\Pr\{N_i(r_u)=0\}) \tag{4}$$

$$= \sum_{i^{(1)}} \cdots \sum_{i^{(l)}} f_{i^{(1)}} f_{i^{(2)}|i^{(1)}} \cdots f_{i^{(l)}|i^{(1)}} \cdot \prod_{i=i^{(1)}}^{i^{(l)}} (1-e^{-\lambda p_i \pi r_u^2}), \tag{5}$$

where $i^{(l)}$ represents the index of the l-th requested content. The summation $\sum_{i^{(l)}}$ averages the l-th requested content by using its popularity $f_{i^{(l)}|i^{(1)}}$. Devices usually store recently consumed contents in their local cache memory and repeated requests are directly provided from its local cache memory; therefore, we suppose that $i^{(l)} \neq i^{(1)}, \cdots, i^{(l-1)}$ for all $l = 2, \cdots, L$ in (5).

In practical scenarios, users consume different numbers of contents, and the number of contents that the user requested could vary also. Therefore, to improve the cache hit rates of all types of the users, we maximize the minimum cache hit rate of the users of all types $1, \cdots, L$ as follows:

$$\mathbf{p}^* = \underset{p_i, i\in\{1,\cdots,F\}}{\arg\max} \Big[\min\{P_1^h, \cdots, P_L^h\}\Big] \tag{6}$$

$$\text{s.t.} \quad \sum_{i=1}^{F} p_i \le M \tag{7}$$

$$0 \le p_i \le 1, \quad \forall i \in \{1, \cdots, F\}, \tag{8}$$

where (7) results from the finite memory size of caching helpers in [11].

The main reason why we maximize the minimum cache hit rates among all types of users is to weigh the importance of the heavy users. For example, one user watching ten videos on average is better than ten users consuming only one content. Users watching one video are not very loyal to this service or application, and we cannot guarantee that they will be back to this service for consuming more contents. For this reason, we would like to support the heavy users in the content providers' view by maximizing the minimum cache hit rate because Lemma 1 shows that the heaviest user's cache hit rate is the smallest.

B. Key Lemmas and Problem Re-Organization

The optimization problem of (6)–(7) can be reorganized via the following lemmas. By using Lemma 1, the max-min problem of (6)–(7) can be transformed into a convex optimization problem. In addition, Lemma 2 turns the inequality constraint (7) into an equality constraint.

Lemma 1: $P_l^h > P_m^h$ for any $l, m \in \{1, \cdots, L\}$ and $l < m$.

Proof: By showing $P_l^h > P_{l+k}^h$ for a positive integer k, this can be proved.

$$\begin{aligned} &P_l^h - P_{l+1}^h \\ &= \sum_{i^{(1)}} \cdots \sum_{i^{(l)}} f_{i^{(1)}} f_{i^{(2)}|i^{(1)}} \cdots f_{i^{(l)}|i^{(1)}} \cdot \Bigg[\prod_{i=i^{(1)}}^{i^{(l)}} (1-e^{-Cp_i}) \\ &\quad - \sum_{i^{(l+1)}} f_{i^{(l+1)}|i^{(1)}} \Bigg\{ \prod_{i=i^{(1)}}^{i^{(l)}} (1-e^{-Cp_i})(1-e^{-Cp_{i^{(l+1)}}}) \Bigg\} \Bigg] \\ &= \sum_{i^{(1)}} \cdots \sum_{i^{(l)}} f_{i^{(1)}} f_{i^{(2)}|i^{(1)}} \cdots f_{i^{(l)}|i^{(1)}} \cdot \\ &\quad \Bigg[\prod_{i=i^{(1)}}^{i^{(l)}} (1-e^{-Cp_i}) \Big(1 - \sum_{i^{(l+1)}} f_{i^{(l+1)}|i^{(1)}} (1-e^{-Cp_{i^{(l+1)}}})\Big) \Bigg] \\ &> 0. \end{aligned}$$

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

Since $\sum_{i^{(l+1)}} f_{i^{(l+1)}|i^{(l)}} = 1$, the second equality and the last inequality are satisfied. Thus, $P_l^h > P_{l+1}^h > \cdots > P_k^h$, and this lemma is proved. ■

Lemma 2: The optimum vector $\mathbf{p}^* = (p_1^*, \cdots, p_F^*)^T$ satisfies

$$\sum_{i=1}^{F} p_i^* = M. \tag{9}$$

Proof: Assume $\sum_{i=1}^{F} p_i^* < M$; then, $\exists \epsilon > 0$ such that $\sum_{i=1}^{F} p_i^* + \epsilon \leq M$ and $p_k^* + \epsilon \leq 1$ for a certain $k \in \{1, \cdots, F\}$. Let $\mathbf{p}' \triangleq (p_1^*, \cdots, p_k^* + \epsilon, \cdots, p_F^*)^T$; then, since P_L^h is an increasing function of any p_i for $i = \{1, \cdots, L\}$, $P_L^h(\mathbf{p}') < P_L^h(\mathbf{p}^*)$. Here, according to Lemma 1, the problem of (6)–(8) is converted into a maximization problem of P_L^h. Thus, it obviously leads to a contradiction. ■

According to Lemmas 1 and 2, the max-min optimization problem of (6)–(7) can be transformed into the following convex maximization problem:

$$\mathbf{p}^* = \underset{p_i, i \in \{1, \cdots, F\}}{\arg\max}\ P_L^h \tag{10}$$

$$\text{s.t.}\ \sum_{i=1}^{F} p_i = M \text{ and } 0 \leq p_i \leq 1, \quad \forall i \in \{1, \cdots, F\}. \tag{11}$$

C. Subproblem for Optimization of Two Contents and Iterative Algorithm

Since P_L^h is a multivariable function consisting of many exponential terms, an iterative algorithm is used to maximize P_L^h. A subproblem with respect to two decision variables is formulated by considering the other variables as constants. Let p_m and p_n be the caching probabilities to be optimized and let the other probabilities $\{p_i\}_{i \neq m,n}$ be fixed. In terms of p_m and p_n, P_L^h of (5) can be divided into four different parts as follows:

$$\begin{aligned} P_L^h &= a_{m,n}(1 - e^{-Cp_m})(1 - e^{-Cp_n}) \\ &\quad + b_{m,n}(1 - e^{-Cp_m}) + d_{m,n}(1 - e^{-Cp_n}) + e_{m,n}, \end{aligned} \tag{12}$$

where $a_{m,n}$, $b_{m,n}$, $d_{m,n}$, and $e_{m,n}$ are coefficients consisting of the system parameters, e.g., f_i, $f_{j|i^{(1)}}$, e^{-Cp_k} for any $i, j, k \in \{1, \cdots, F\}$ and $k \neq m, n$. The first term of $(1 - e^{-Cp_m})(1 - e^{-Cp_n})$ represents the event where the user requests both c_m and c_n. Similarly, the second and third parts of $(1 - e^{-Cp_m})$ and $(1 - e^{-Cp_n})$ correspond to the events where the user requests c_m but not c_n, and c_n but not c_m, respectively. The last constant term $e_{m,n}$ is obtained when the user does not request either c_m and c_n. For example, $b_{m,n}$ is given by

$$\begin{aligned} b_{m,n} &= f_m \sum_{i^{(2)}} \cdots \sum_{i^{(L)}} f_{i^{(2)}|m} \cdots f_{i^{(L)}|i^{(L-1)}} \\ &\quad + \sum_{i^{(1)}} \sum_{i^{(3)}} \cdots \sum_{i^{(L)}} f_{i^{(1)}} f_{m|i^{(1)}} \cdots f_{i^{(L)}|i^{(L-1)}} \\ &\quad + \cdots + \sum_{i^{(1)}} \cdots \sum_{i^{(L-1)}} f_{i^{(1)}} \cdots f_{i^{(L-1)}|i^{(L-2)}} f_{m|i^{(L-1)}}. \end{aligned} \tag{13}$$

In addition, $a_{m,n}$, $d_{m,n}$, and $e_{m,n}$ can be obtained in a similar way by using the procedure used to obtain $b_{m,n}$. Then, the subproblem for finding the optimal p_m and p_n is formulated as follows:

$$\{p_m^*, p_n^*\} = \underset{p_m, p_n}{\arg\min}\ \mathcal{M}_{(p_m, p_n)} \tag{14}$$

$$\text{s.t.}\ p_m + p_n = z_{m,n} = M - \sum_{i=1, i \neq m,n}^{F} p_i \tag{15}$$

$$0 \leq p_m, p_n \leq 1, \tag{16}$$

where $\mathcal{M}_{(p_m,p_n)} = (a_{m,n} + b_{m,n}) e^{-C \cdot p_m} + (a_{m,n} + d_{m,n}) e^{-C \cdot p_n}$. $\mathcal{M}_{(p_m,p_n)}$ is obtained from (12) by removing the constant terms and reversing the sign. Since $\{p_i\}_{i \neq m,n}$ are fixed, $p_m + p_n$ also becomes a constant. The following proposition provides the solution for the above subproblem.

Proposition 1: The optimal solution of the problem (14)–(16) is as follows:

$$\{p_m^*, p_n^*\} = \begin{cases} \{\tilde{p}_m, \tilde{p}_n\} & \text{if } 0 \leq \tilde{p}_m, \tilde{p}_n \leq 1 \\ \underset{\{p_m, p_n\}}{\arg\min} \{\mathcal{M}_{(0, z_{m,n})}, \mathcal{M}_{(z_{m,n}, 0)}\} & \text{else if } z_{m,n} < 1 \\ \underset{\{p_m, p_n\}}{\arg\min} \{\mathcal{M}_{(1, z_{m,n}-1)}, \mathcal{M}_{(z_{m,n}-1, 1)}\} & \text{else if } z_{m,n} \geq 1, \end{cases} \tag{17}$$

where

$$\tilde{p}_m = \frac{1}{2C} \log \frac{b_{m,n}}{d_{m,n}} + \frac{1}{2} z_{m,n}$$
$$\tilde{p}_n = \frac{1}{2C} \log \frac{d_{m,n}}{b_{m,n}} + \frac{1}{2} z_{m,n}. \tag{18}$$

Proof: According to the arithmetic-geometric mean inequality, the lower bound on $\mathcal{M}_{(p_m,p_n)}$ is given by $b_{m,n} e^{-C \cdot \tilde{p}_m} + d_{m,n} e^{-C \cdot \tilde{p}_n} \geq 2\sqrt{b_{m,n} d_{m,n} e^{-C \cdot z_{m,n}}}$. The equality holds if and only if $b_{m,n} e^{-C \cdot \tilde{p}_m} = d_{m,n} e^{-C \cdot \tilde{p}_n}$. Since $\tilde{p}_m = z_{m,n} - \tilde{p}_n$, $\tilde{p}_m$ and $\tilde{p}_n$ are obtained using (18). For (16), $\tilde{p}_m$ and $\tilde{p}_n$ become the optimal solution only when $0 \leq \tilde{p}_m, \tilde{p}_n \leq 1$. Otherwise, the four boundary conditions are compared as follows: 1) $p_m = 0$ and $p_n = z_{m,n}$, 2) $p_m = z_{m,n}$ and $p_n = 0$, 3) $p_m = 1$ and $p_n = z_{m,n} - 1$, and 4) $p_m = z_{m,n}$ and $p_n = z_{m,n} - 1$. Thus, the optimal solution (17) is obtained. ■

Note that $b_{m,n}$ represents the probability that c_m is in the sequence of L content requests but c_n is not, and $d_{m,n}$ is the probability that c_n is in the sequence of L content request but c_m is not. $b_{m,n}$ and $d_{m,n}$ would be large if c_m and c_n are in $\mathcal{G}^{(1)}$ respectively, and caching probabilities are determined depending on their values as shown in (18).

If the convergence is guaranteed, a multivariable function P_L^h can be optimized by iteratively optimizing the subset of variables. To find $\mathbf{p}^* = [p_1^*, \cdots, p_F^*]$, the subproblem of (14)–(15) can be iteratively optimized for all combinations of m and n, where $m, n \in \{1, \cdots, F\}$ and $m \neq n$. The details are given in Algorithm 1. We find the minimum of the dual-variable problem of (14)–(15) in each iteration, and the sequence of updated objective values $\mathcal{M}_{(p_m,p_n)}$ is non-increasing. Therefore, the corresponding sequence of updated

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Iterative Algorithm for the Optimization Problem of (10)-(11)

```
Precondition:
    • $M$: memory size
    • $F$: the number of contents
$p_i^* = \frac{M}{F}$ for all $i \in \{1, \cdots, F\}$
for $\forall (m, n) \in \{1, \cdots, F\} \times \{1, \cdots, F\}$ and $m \neq n$ do
    $z_{m,n} = M - p_m^* - p_n^*$
    Find $\tilde{p}_m$ and $\tilde{p}_n$ according to (18).
    if $0 \leq \tilde{p}_m, \tilde{p}_n \leq 1$ then
        $p_m^* \leftarrow \tilde{p}_m$ and $p_n^* \leftarrow \tilde{p}_n^*$
    else if $z_{m,n} < 1$ then
        $\{p_m^*, p_n^*\} \leftarrow \underset{\{p_m, p_n\}}{\arg\min} \{\mathcal{M}_{(0, z_{m,n})}, \mathcal{M}_{(z_{m,n}, 0)}\}$
    else if $z_{m,n} \geq 1$ then
        $\{p_m^*, p_n^*\} \leftarrow \underset{\{p_m, p_n\}}{\arg\min} \{\mathcal{M}_{(1, z_{m,n}-1)}, \mathcal{M}_{(z_{m,n}-1, 1)}\}$
    end if
end for
```

P_L^h from (10) is nondecreasing. Since P_L^h has a trivial upper bound of 1, i.e., $P_L^h \leq 1$, the convergence of Algorithm 1 is guaranteed. If we randomly pick two different variables m and n from $\{1, \cdots, F\}$ and sufficient iterations are performed, the global optimal solution of the problem (10)–(11) can be obtained.

Algorithm 1 requires F^2 iterations, and its computational complexity can be a serious problem when F is large. However, we can skip some iterations required for solving the subproblem of two contents having very small popularity both. Note that if randomly picked two contents have very small popularity, we do not have to accurately derive their caching probabilities because they have very little influence on the cache hit rate. We generate a subset of the content indices, i.e., $\Psi \subset \{1, \cdots, F\}$, whose contents have popularity smaller than a predefined threshold ζ. This method leads Algorithm 1 to iterate the dual-variable optimization process for all $(m, n) \in \{1, \cdots, F\} \times \{1, \cdots, F\} \setminus \Psi$. The iteration number of this method is still linearly increasing with F; however, it can significantly reduce the complexity because a large portion of the contents are in Ψ. In addition, large L may significantly increase the complexity of Algorithm 1 for computing the coefficients as given by (13). However, according to [30], the number of contents that a user consumes is limited; therefore, its required complexity is also not very large.

IV. Dynamic Content Delivery for Consecutive User Demands

In this section, we propose a strategy to determine an appropriate caching helper for satisfying the consecutive user demands. Based on the Markov decision process (MDP), we formulate the helper association problem. Here, we assume that an additional delay occurs when the user changes the caching helper for receiving the next content. Then, dynamic programming (DP) can provide a dynamic content delivery strategy for consecutive user demands.

A. Caching Helper Associations for Consecutive User Demands

Let the user request $c_i \in \mathcal{F}$ first and helper α is assumed to deliver c_i to this user. If this user subsequently requests another content $c_j \neq c_i$ and α does not cache c_j, then the user has to find another helper. In order to switch the caching helper for receiving c_j, the user should send the request message to some helpers for link activation, and helpers choose whether to accept or reject the request. These message exchanges cause additional time delay; therefore, we suppose that an additional delay τ_N is generated when the user switches the caching helper. The user thus prefers to find the helper that can continuously provide as many contents as possible to reduce the overall delay time while consuming subsequent contents.

If the content delivery fails due to the bad channel condition, we call this event as link outage and the user requires retransmission. Therefore, considering discretized time slots $t \in \{1, 2, \cdots\}$, the delay, which is equal to the unit time duration t_o, is caused by an outage event. Assuming that the size of a content is S, the content delivery can be successful if the data rate is larger than S/t_o. Therefore, the threshold of data rate to ensure successful delivery in each slot can be determined as $\rho = \frac{S}{t_o}$. According to (1), the successful delivery probability can be obtained as

$$z_\alpha^s = 1 - \exp\Big\{ -\frac{1}{2}(d_\alpha^\beta \sigma^2 (\Upsilon_u + 1))(2^{\rho/\mathcal{B}} - 1)\Big\}. \qquad (19)$$

The expected delay time caused by the outage event when the user receives the content from helper α is denoted by $\tau_{o,\alpha}$, which can be obtained as

$$\tau_{o,\alpha} = \sum_{k=1}^{\infty} z_\alpha^s \cdot (1 - z_\alpha^s)^k \cdot k \cdot t_o = \frac{1 - z_\alpha^s}{z_\alpha^s} t_o. \qquad (20)$$

If a content consists of V chunks, the expected delay time caused by outage events until V chunks are successfully delivered to the user is obtained as follows:

$$\begin{aligned}\tau_{o,\alpha} &= \sum_{k=1}^{\infty} \binom{V+k-1}{k} (z_\alpha^s)^V (1 - z_\alpha^s)^k \cdot k \cdot t_o \\ &= \frac{V(1 - z_\alpha^s)}{z_\alpha^s} t_o. \qquad (21)\end{aligned}$$

Each term in (20) and (21) represents the delay time incurred by k outage events for all $k \in \{1, 2, \cdots\}$. If the user cannot find an appropriate helper caching the requested content within radius r_u, then the user needs to receive the content from a server (which has the whole file library) via backhaul communications. Suppose that a backhaul delay τ_B is generated when the user has to receive the content from the server. Since backhaul communications generate greater latency than that when employing the caching network, we suppose that $t_o < \tau_N < \tau_B$.

For example, consider two helpers around the content-requesting user as shown in Fig. 3. Let $c_{k,i}$ denote the i-th content in $\mathcal{G}_k$. Suppose that the user requests $c_{1,1}$ at first. Then, the user is highly likely to request another content in $\mathcal{G}_1$ next. Assume that $\mathcal{C}_x = \{c_{1,1}, c_{2,1}, c_{3,1}\}$ and

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

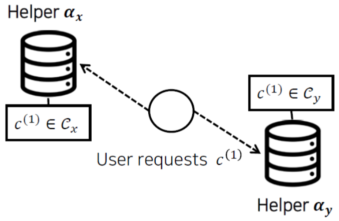


Fig. 3. Helper association example.

$\mathcal{C}_y = \{c_{1,1}, c_{1,2}, c_{1,3}\}$. Then, some delay time due to the switching of helper associations can be saved when the user is initially associated with α_y, which is more probable to provide multiple contents in a row compared to α_x. However, if α_y has a worse channel condition than α_x, outage events could occur more frequently at the link with α_y than at the link with α_x. In this case, there exists a tradeoff between link outage events and the switching of helper associations in terms of delay time. Definitely, the helper that caches many contents of the target category and has a good channel condition that is the best choice; otherwise, the tradeoff between link outage events and switching of helper associations should be taken into account at the user side.

B. Dynamic Delivery for Consecutive User Demands

The goal of this subsection is to find an appropriate caching helper $\alpha^{(l)}$ to receive $c^{(l)}$ for all $l \in \{1, \cdots, L\}$. The problem that minimizes the expected delay can be formulated as follows:

$$\boldsymbol{\alpha} = \underset{\alpha^{(l)} \in \mathcal{A}(l)}{\arg\min} \sum_{l=1}^{L} \tau^{(l)}, \tag{22}$$

where $\boldsymbol{\alpha} = [\alpha^{(1)}, \cdots, \alpha^{(L)}]$, $\mathcal{A}(l)$ is the helper candidate set storing $c^{(l)}$, $\tau^{(l)}$ is the expected delay for receiving $c^{(l)}$, and L is the type of the user. The problem in (22) is a stochastic shortest path problem based on MDP. $c^{(1)}$ and $\mathcal{A}(1)$ are given before making decisions on $\boldsymbol{\alpha}$ in the future.

When requesting the l-th content, the previously associated helper $\alpha^{(l-1)}$ represents the state, and $c^{(l)}$ is a random event. Then, the state $\alpha^{(l-1)}$ can have a Markov property because the decision on whether maintaining or switching the helper depends on both $\alpha^{(l-1)}$ and $c^{(l)}$. The action becomes a choice of $\alpha^{(l)}$, and how valuable the action is can be assessed by its expected delay time. The action at the l-th request is denoted by $\Theta(l)$. Although the choice of $c^{(l)}$ is random and does not depend on the previous node associations, the user obviously knows its next desired content and $c^{(l)}$ determines $\mathcal{A}(l)$ which is the action space at the l-th request. Therefore, $c^{(l)}$ can be an element of the state set, and we can define the state at the l-th request as $S(l) = \{\alpha^{(l-1)}, c^{(l)}\}$. Also, define $\mathcal{S} \triangleq \mathcal{A} \times \mathcal{F}$ and $\Xi \triangleq \mathcal{A}$ as the state space and action space, respectively, where $\mathcal{A}$ is the set of helper candidates.

The cost of MDP is the expected delay time caused by action $\Theta(l) = \alpha^{(l)}$ when the agent is at state $S(l) = \{\alpha^{(l-1)}, c^{(l)}\}$, as given by

$$\begin{aligned} &g(S(l), \Theta(l)) \\ &= \begin{cases} \dfrac{1 - z^s_{\alpha^{(l)}}}{z^s_{\alpha^{(l)}}} \tau_{o,\alpha^{(l)}} & \text{if } c^{(l)} \in \mathcal{C}_{\alpha^{(l)}} \text{ and } \alpha^{(l)} = \alpha^{(l-1)} \\ \dfrac{1 - z^s_{\alpha^{(l)}}}{z^s_{\alpha^{(l)}}} \tau_{o,\alpha^{(l)}} + \tau_N & \text{if } c^{(l)} \in \mathcal{C}_{\alpha^{(l)}} \text{ and } \alpha^{(l)} \neq \alpha^{(l-1)} \\ \tau_B & \text{if } c^{(l)} \notin \mathcal{C}_{\alpha^{(l)}}. \end{cases} \end{aligned} \tag{23}$$

Since $\alpha^{(l)}$ is determined by any action $\Theta(l) \in \Xi$, the transition probability from $S(l)$ to $S(l+1)$ follows the temporary popularity in (3), and can be defined for all states s and s' as

$$\begin{aligned} P_{s's}(\theta) &= \Pr\{S(l+1) = s' | S(l) = s, \Theta(l) = \theta\} \\ &= \mathcal{I}\{\theta = s'\} \cdot f_{\psi(c^{(l)})|\psi(c^{(1)})}, \end{aligned} \tag{24}$$

where $s, s' \in \mathcal{S}$, $\theta \in \Xi$, $\mathcal{I}(.)$ is the indicator function and $\psi(c)$ is the index of content c in library $\mathcal{F}$. The term $\mathcal{I}\{\theta = s'\}$ forces the agent to choose the helper that the action indicates.

The minimum incurred cost at $S(l_0) = s_0$ conditioned on $\Theta(l_0) \in \mathcal{A}(l_0)$ is given by

$$G(s_0) = \min_{\Theta} \mathbb{E}\Big[\sum_{l=l_0}^{L} g(S(l), \Theta(l)) \Big| S(l_0) = s_0\Big], \tag{25}$$

According to Bellman optimality equation, the DP minimizes the cost as follows:

$$\begin{aligned} &G(s_0) \\ &= \min_{\theta} \mathbb{E}\Big[g(s_0, \theta) + G\big(S(l_0+1)\big) \Big| S(l_0) = s_0, \Theta(l_0) = \theta\Big] \\ &= \min_{\theta} \mathbb{E}\Big[g(s_0, \theta) + \sum_{y \in \mathcal{S}} P_{y,s_0}(\theta) G(y) \\ &\qquad\qquad \Big| S(l_0) = s_0, \Theta(l_0) = \theta\Big] \end{aligned} \tag{26}$$

where the expectation of (26) is with respect to $\{c^{(l)} : l \in \{l_0+1, \cdots, L\}\}$. The minimum cost is obtained by greedily testing all possible actions (i.e., helper associations) $\Theta(l_0) \in \mathcal{A}(l_0)$.

The end costs of $\mathbf{G}(S(L+1))$ are required to find the optimal costs $G^*(S(l))$ for all $l \in \{1, \cdots, L\}$ and $S(l) \in \mathcal{S}$. Note that at the end of the path, the user stops requesting the content; therefore, the state space at the end is $\mathcal{A}$, rather than $\mathcal{A} \times \mathcal{F}$. Even if the helper that does not cache the requested content is chosen in MDP, the user is able to receive the content from the server with the cost τ_B. It means that any path from $l = 1$ to $l = L$ ensures that L consecutive user demands are provided to the user; therefore, we do not have to assign different end costs; therefore, suppose that $G(S(L+1)) = 0$ for all $S(L+1) \in \mathcal{A}$. Then, $G^*(S(l))$ for all $l \in \{1, \cdots, L\}$ and $S(l) \in \mathcal{S}$ can be obtained by using the DP equation (26). After finding the cost sums at the first step for all $\alpha \in \mathcal{A}(1)$, the user determines the initial caching helper by comparing cost values, as described in (22). In addition, the helper association plan for every content can be designed by storing the expected costs at every step of the MDP.

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

Remark: Note that the proposed delivery scheme is adaptive to the time-varying channel and random user demands; therefore, this delivery phase is not jointly optimized with the caching part. The caching method proposed in Section III is optimized in a probabilistic manner to observe the effects of consecutively requested contents whose popularity depends on the first content. On the other hand, in the delivery phase, we aimed at demonstrating the effects of switching of helper associations while requesting multiple contents continuously. However, it is very difficult to derive the probabilistic node association method with consideration of the delivery latency because it has to average out helper distribution as well as contents randomly cached in all nearby caching helpers. Thus, we consider the deterministic scenario in the delivery phase where locations and cache states of helpers are fixed and the user knows them. Nevertheless, the main goal of both caching and delivery proposed in this paper is the same, which makes consecutive user demands be provided from the identical caching helper as much as possible.

As future research directions, modeling of the popularity profile and joint optimization of caching and delivery for consecutive user demands are worthy to consider. Since the content delivery depends on the user's random requests, the relationship between the caching distribution and the popularity profile of consecutively requested contents would be a key for joint optimization of caching and delivery. Therefore, after the popularity capturing the temporary preference is closely modeled, caching and delivery for consecutive user demands can be jointly optimized.

C. Dynamic Delivery Analysis in a Two-Helper Scenario

This subsection provides the guideline to the user for helper association to minimize the delivery latency. We first begin with the simple situation in which there are two helper options and the user requests two contents in a row. Later, it will be generalized to multiple helper options and L consecutive content requests. Consider the situation shown in Fig. 3 where two helpers are denoted as α_x and α_y, and their content sets are denoted by $\mathcal{C}_x$ and $\mathcal{C}_y$, respectively. In this scenario, if any content not cached in both helpers is requested, the delay τ_B is caused regardless of which helper is associated with the user; therefore, this case is not considered here.

The following lemma gives the condition that determines which helper is better for the user to be associated in terms of latency minimization, especially when there are only two helper options and $L = 2$.

Lemma 3: When $\tau_{o,\alpha_x} \leq \tau_{o,\alpha_y}$ and a user consecutively requests $L = 2$ contents, if (27) is satisfied, it is better for the user to be initially associated with α_x than α_y in terms of the expected delivery delay.

$$\tau_N\Big(\sum_{c_j\in\mathcal{C}_x\setminus\mathcal{C}_y} f_{j|i^{(1)}} - \sum_{c_u\in\mathcal{C}_y\setminus\mathcal{C}_x} f_{u|i^{(1)}}\Big) + (\tau_{o,\alpha_y} - \tau_{o,\alpha_x})\Big(\sum_{c_j\in\mathcal{C}_x\cap\mathcal{C}_y} f_{j|i^{(1)}} + 1\Big) \geq 0. \quad (27)$$

Proof: Let $D_\alpha(l)$ denote the expected delay sum while receiving l contents in a row when the initial helper association is α. If the user initially chooses α_x, the expected delay for delivery of two contents is obtained as

$$D_{\alpha_x}(2) = \tau_{o,\alpha_x}\sum_{c_j\in\mathcal{C}_x} f_{j|i^{(1)}} + (\tau_{o,\alpha_y} + \tau_N)\sum_{c_u\in\mathcal{C}_y\setminus\mathcal{C}_x} f_{u|i^{(1)}} + \tau_{o,\alpha_x}, \quad (28)$$

where the last term τ_{o,α_x} is the latency for the first content delivery and the other terms represent the expected delivery delay of the second content. Similarly, if α_y is initially associated with the user, the expected latency is given by

$$D_{\alpha_y}(2) = (\tau_{o,\alpha_x} + \tau_N)\sum_{c_j\in\mathcal{C}_x\setminus\mathcal{C}_y} f_{j|i^{(1)}} + \tau_{o,\alpha_y}\sum_{c_u\in\mathcal{C}_y} f_{u|i^{(1)}} + \tau_{o,\alpha_y}. \quad (29)$$

Therefore, if (27) is satisfied, $D_{\alpha_y}(2) \geq D_{\alpha_x}(2)$. ■

In Lemma 3, we can easily notice that the first term in (27) gives an importance to the helper that caches more contents in the user's target category, and the helper which is closer to the user benefits from the second term in (27). Conversely, if the condition in (27) is not satisfied, initial association with α_y is recommended to the user. Here, in order for the user to choose the initial helper based on (27) in practical scenarios, the user has to know the popularity of the contents stored in both helpers and the distances from helpers in advance of helper association.

The following lemmas and proposition are extended from Lemma 3 and provide the guideline for the general helper association with multiple helper options and L consecutive content requests.

Lemma 4: If (27) is satisfied and $D_{\alpha_i}(L) \leq D_{\alpha_j}(L)$ for any $i, j \in \{x, y\}$ and $i \neq j$, then $D_{\alpha_i}(L+1) \leq D_{\alpha_j}(L+1)$ is satisfied for all $L \geq 2$.

Proof: Suppose that $D_{\alpha_y}(L) \geq D_{\alpha_x}(L)$ for any $L \geq 2$. The sequence vector of consecutively requested contents is denoted by $\mathbf{c}^{(L)} = \{c^{(1)}, c^{(2)}, \cdots, c^{(L)}\}$. The delivery phase for L contents can be divided into two different situations, in which $\alpha^{(L)} = \alpha_x$ and $\alpha^{(L)} = \alpha_y$. Suppose that the user continuously requests the $(L+1)$-th content $c^{(L+1)}$. Then, $c^{(L+1)}$ can be in one of following exclusive sets: $\mathcal{C}_x \setminus \mathcal{C}_y$, $\mathcal{C}_y \setminus \mathcal{C}_x$, $\mathcal{C}_x \cap \mathcal{C}_y$, and $(\mathcal{C}_x \cup \mathcal{C}_y)^c$. The expected latency during delivery of L contents when the initial helper is α and $\alpha^{(L)} = \alpha_i$ is given by $D_\alpha(L|\alpha^{(L)} = \alpha_i)$ for any $i \in \{x, y\}$. Then,

$$\begin{aligned} &D_\alpha(L+1|\alpha^{(L)} = \alpha_x) \\ &\quad= D_\alpha(L|\alpha^{(L)} = \alpha_x) + \Big(\Pr\{c^{(L+1)} \in \mathcal{C}_x\}\cdot\tau_{o,\alpha_x} \\ &\qquad+ \Pr\{c^{(L+1)} \in \mathcal{C}_y \setminus \mathcal{C}_x\}\cdot(\tau_N + \tau_{o,\alpha_y}) \\ &\qquad+ \Pr\{c^{(L+1)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\cdot\tau_B\Big) \end{aligned} \quad (30)$$

and

$$\begin{aligned} &D_\alpha(L+1|\alpha^{(L)} = \alpha_y) \\ &\quad= D_\alpha(L|\alpha^{(L)} = \alpha_y) + \Big(\Pr\{c^{(L+1)} \in \mathcal{C}_y\}\cdot\tau_{o,\alpha_y} \\ &\qquad+ \Pr\{c^{(L+1)} \in \mathcal{C}_x \setminus \mathcal{C}_y\}\cdot(\tau_N + \tau_{o,\alpha_x}) \\ &\qquad+ \Pr\{c^{(L+1)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\cdot\tau_B\Big). \end{aligned} \quad (31)$$

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

Let $\Delta_{\alpha^{(1)}}(L+1)$ denote the expected additional delay while receiving the $(L+1)$-th content when the initial associated helper is $\alpha^{(1)}$; then, $D_\alpha(L+1) = D_\alpha(L) + \Delta_\alpha(L+1)$. Here, $\Delta_{\alpha^{(1)}}(L+1)$ can be obtained as

$$\begin{aligned}\Delta_{\alpha^{(1)}}(L+1) = \Pr\{\alpha^{(L)} = \alpha_x|\alpha^{(1)}\} \cdot \delta(\alpha_x)\\ +\Pr\{\alpha^{(L)} = \alpha_y|\alpha^{(1)}\} \cdot \delta(\alpha_y),\end{aligned} \quad (32)$$

where $\delta(\alpha)$ represents the latency during delivery of a content when the previously associated helper is α, according to (30) and (31), as given by

$$\begin{aligned}\delta(\alpha_x) = &\Pr\{c^{(L+1)} \in \mathcal{C}_y \setminus \mathcal{C}_x\} \cdot (\tau_N + \tau_{o,\alpha_y})\\ &+\Pr\{c^{(L+1)} \in \mathcal{C}_x\} \cdot \tau_{o,\alpha_x}\\ &+\Pr\{c^{(L+1)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\} \cdot \tau_B\end{aligned} \quad (33)$$

and

$$\begin{aligned}\delta(\alpha_y) = &\Pr\{c^{(L+1)} \in \mathcal{C}_x \setminus \mathcal{C}_y\} \cdot (\tau_N + \tau_{o,\alpha_x})\\ &+\Pr\{c^{(L+1)} \in \mathcal{C}_y\} \cdot \tau_{o,\alpha_y}\\ &+\Pr\{c^{(L+1)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\} \cdot \tau_B.\end{aligned} \quad (34)$$

Since we assume that (27) is satisfied, $\delta(\alpha_x) \leq \delta(\alpha_y)$ can be proven by

$$\begin{aligned}&\delta(\alpha_y) - \delta(\alpha_x)\\ &= \Pr\{c^{(L+1)} \in \mathcal{C}_y\}\tau_{o,\alpha_y} - \Pr\{c^{(L+1)} \in \mathcal{C}_x\}\tau_{o,\alpha_x}\\ &\quad + \Pr\{c^{(L+1)} \in \mathcal{C}_x \setminus \mathcal{C}_y\}(\tau_N + \tau_{o,\alpha_x})\\ &\quad - \Pr\{c^{(L+1)} \in \mathcal{C}_y \setminus \mathcal{C}_x\}(\tau_N + \tau_{o,\alpha_y})\\ &= \Big(\Pr\{c^{(L+1)} \in \mathcal{C}_x \setminus \mathcal{C}_y\} - \Pr\{c^{(L+1)} \in \mathcal{C}_y \setminus \mathcal{C}_x\}\Big) \cdot \tau_N\\ &\quad + \Pr\{c^{(L+1)} \in \mathcal{C}_x \cap \mathcal{C}_y\} \cdot (\tau_{o,\alpha_y} - \tau_{o,\alpha_x})\\ &\geq 0.\end{aligned} \quad (35)$$

In (32), $\Pr\{\alpha^{(L)} = \alpha_x|\alpha^{(1)}\}$ and $\Pr\{\alpha^{(L)} = \alpha_y|\alpha^{(1)}\}$ can be obtained in the following manner:

$$\begin{aligned}&\Pr\{\alpha^{(L)} = \alpha_x|\alpha^{(1)} = \alpha_x\} = \Pr\{\mathbf{c}^{(2,L-1)}\}\Pr\{c^{(L)} \in \mathcal{C}_x \setminus \mathcal{C}_y\}\\ &\quad + \Pr\{\mathbf{c}^{(2,L-1)}\} \cdot \Pr\{c^{(L-1)} \in \mathcal{C}_x \setminus \mathcal{C}_y\}\\ &\qquad \cdot \Big(\Pr\{c^{(L)} \in \mathcal{C}_x \cap \mathcal{C}_y\} + \Pr\{c^L \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\Big)\\ &\quad + \cdots + \Pr\{c^{(2)}\} \cdot \Pr\{c^{(3)} \in \mathcal{C}_x \setminus \mathcal{C}_y\}\\ &\qquad \cdot \prod_{l=4}^{L}\Big(\Pr\{c^{(l)} \in \mathcal{C}_x \cap \mathcal{C}_y\} + \Pr\{c^{(l)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\Big)\\ &\quad + \Pr\{c^{(2)} \in \mathcal{C}_x\} \cdot \prod_{l=3}^{L}\Big(\Pr\{c^{(l)} \in \mathcal{C}_x \cap \mathcal{C}_y\}\\ &\qquad + \Pr\{c^{(l)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\Big)\end{aligned} \quad (36)$$

$$\begin{aligned}&\Pr\{\alpha^{(L)} = \alpha_y|\alpha^{(1)} = \alpha_x\} = \Pr\{\mathbf{c}^{(2,L-1)}\}\Pr\{c^{(L)} \in \mathcal{C}_x \setminus \mathcal{C}_y\}\\ &\quad + \Pr\{\mathbf{c}^{(2,L-1)}\} \cdot \Pr\{c^{(L-1)} \in \mathcal{C}_x \setminus \mathcal{C}_y\}\\ &\qquad \cdot \Big(\Pr\{c^{(L)} \in \mathcal{C}_x \cap \mathcal{C}_y\} + \Pr\{c^{(L)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\Big)\\ &\quad + \cdots + \Pr\{c^{(2)}\} \cdot \Pr\{c^{(3)} \in \mathcal{C}_x \setminus \mathcal{C}_y\} \cdot\\ &\qquad \prod_{l=4}^{L}\Big(\Pr\{c^{(l)} \in \mathcal{C}_x \cap \mathcal{C}_y\} \cdot +\Pr\{c^{(l)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\Big)\\ &\quad + \Pr\{c^{(2)} \in \mathcal{C}_x \setminus \mathcal{C}_y\} \cdot \prod_{l=3}^{L}\Big(\Pr\{c^{(l)} \in \mathcal{C}_x \cap \mathcal{C}_y\}\\ &\qquad + \Pr\{c^{(l)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\Big),\end{aligned} \quad (37)$$

where $\mathbf{c}^{(m,n)}$ is the sequence vector from the m-th content to the n-th content for $m < n$. Note that every term in (36) and (37) is the same except for the last term. The last term in (36) includes $\Pr\{c^{(2)} \in \mathcal{C}_x\}$, but that in (37) includes $\Pr\{c^{(2)} \in \mathcal{C}_x \setminus \mathcal{C}_y\}$; therefore, $\Pr\{\alpha^{(L)} = \alpha_x|\alpha^{(1)} = \alpha_x\} > \Pr\{\alpha^{(L)} = \alpha_y|\alpha^{(1)} = \alpha_x\}$. Similarly, $\Pr\{\alpha^{(L)} = \alpha_x|\alpha^{(1)} = \alpha_y\} < \Pr\{\alpha^{(L)} = \alpha_y|\alpha^{(1)} = \alpha_y\}$ can be obtained.

Therefore, $\Delta_{\alpha_y}(L+1) > \Delta_{\alpha_x}(L+1)$ can be proven by

$$\begin{aligned}&\Delta_{\alpha_y}(L+1) - \Delta_{\alpha_x}(L+1)\\ &\quad = \delta(\alpha_x) \cdot \Pr\{\alpha^{(L)} = \alpha_x|\alpha^{(1)} = \alpha_y\}\\ &\qquad + \delta(\alpha_2) \cdot \Pr\{\alpha^{(L)} = \alpha_y|\alpha^{(1)} = \alpha_y\}\\ &\qquad - \delta(\alpha_1) \cdot \Pr\{\alpha^{(L)} = \alpha_x|\alpha^{(1)} = \alpha_x\}\\ &\qquad - \delta(\alpha_2) \cdot \Pr\{\alpha^{(L)} = \alpha_y|\alpha^{(1)} = \alpha_x\}\\ &\quad = \Pr\{c^{(2)} \in \mathcal{C}_x \cap \mathcal{C}_y\} \cdot \prod_{l=3}^{L}\Big(\Pr\{c^{(l)} \in \mathcal{C}_x \cap \mathcal{C}_y\}\\ &\qquad + \Pr\{c^{(l)} \in (\mathcal{C}_x \cup \mathcal{C}_y)^c\}\Big) \cdot (\delta(\alpha_y) - \delta(\alpha_x)).\end{aligned} \quad (38)$$

We have already proved $\delta(\alpha_y) \geq \delta(\alpha_x)$; therefore, $\Delta_{\alpha_y}(L+1) \geq \Delta_{\alpha_x}(L+1)$ and finally $D_{\alpha_y}(L+1) \geq D_{\alpha_x}(L+1)$ if $D_{\alpha_y}(L) \geq D_{\alpha_x}(L)$ is satisfied for any $L \geq 2$. ■

Proposition 2: When the user requests $L > 1$ contents consecutively, if $\tau_{o,\alpha_x} \leq \tau_{o,\alpha_y}$ (27) is satisfied, it is better for the user to be initially associated with α_x than α_y in terms of the expected delivery delay.

Proof: This proposition can be proved by mathematical induction using Lemmas 3 and 4. ■

Proposition 2 provides a guideline on which of the two caching helpers would be better for a content-requesting user in order to reduce the delivery latency. Even if there are K caching helpers that can deliver the desired content around the user, Proposition 2 can greedily find the best initial caching helper association. The worst case requires $K-1$ comparison steps, and K would not be a very large number in practical scenarios; therefore, this greedy search is a reasonable method. Note that Proposition 2 is applied to an individual content request taking into account near-future requests; however, the DP-based delivery proposed in Section IV-B provides even near-future helper associations by estimating consecutive user demands. This enables the user to prepare the next association while consuming the current content, which is more efficient for reducing delays incurred by the switching of helper associations.

D. Load Balancing at Caching Helpers

In the previous subsections, we consider only the scenario where caching helper candidates do not support any other user before. Since we suppose that the density of active users is smaller than that of potential helpers, this scenario is consistent with this assumption; however, still the proposed

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

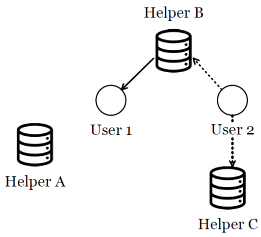


Fig. 4. Example scenario for load balancing at helper side.

node association method can be extended to deal with the load balancing issue.

When multiple users attempt to request their desired contents from the identical caching helper, appropriate resource allocation at the helper is required to support multiple delivery links. The example scenario is illustrated in Fig. 4, in which user 1 is receiving its desired content from helper B, and user 2 newly requests the content. Helpers A and B cache user 1's desired content, and helpers B and C can provide the content request of user 2. When helper B receives the association request from user 2, helper B lets user 2 know its available resources (e.g., frequency bands) while still supporting the reliable delivery for user 1. Then, user 2 can estimate the expected latency for receiving contents from helper B. In this way, user 2 can compare the expected delivery delays from helpers B and C, and finally decide which helper is better in terms of latency minimization. The basic concept of comparing the expected delivery delays from nearby potential helpers is still consistent with the core of the node association method proposed in Section IV-B.

Here, the point is how to allocate fractions of the resources of helper B to user 1 (existing user) and user 2 (new user). Since user 1 already chose helper B rather than helper A, suppose that helper B should provide the sufficient data rate not to lose the advantage of being associated with helper B. Therefore, even though helper B allocates fractions of frequency bands to user 2, the expected latency of the link between helper B and user 1 has to be smaller than or equal to at least that of the link between helper A and user 1. Accordingly, when the user is associated with certain helper, this user should remember its minimum data rate guarantee to preserve the advantage of the helper choice. With the knowledge of the available resources of helper B while guaranteeing the minimum data rate for user 1, user 2 can estimate the expected latency by using the proposed MDP-based method. However, depending on channel conditions, it would be better that user 1 changes its association to helper A, and helper B uses all of its resources for user 2. In this case, other existing delivery links can be reconstructed serially and the centralized link scheduling is required, but this is out of scope in this manuscript.

V. Numerical Results

In this section, we numerically show the impacts of categorized contents and consecutive user demands on the proposed caching and delivery in Sections III and IV, respectively. The simulation settings are as follows: $F = 25$ contents are grouped into $K = 5$ categories and each consists of $F_k = 5$ contents for all k. Suppose that a lower content index indicates a more popular content, i.e., $f_i > f_j$ for $i < j$, and the content lists of K categories are as follows: $\mathcal{G}_1 = \{1, 6, 11, 16, 21\}$, $\mathcal{G}_2 = \{2, 7, 12, 17, 22\}$, $\mathcal{G}_3 = \{3, 8, 13, 18, 23\}$, $\mathcal{G}_4 = \{4, 9, 14, 19, 24\}$, and $\mathcal{G}_5 = \{5, 10, 15, 20, 25\}$. Here, all categories are assumed to be disjoint for simplicity and lack of the exact modeling of the popularity profile for consecutively requested contents. Assume that $M = 7$, $\nu = 0.9$, $\mathcal{B} = 1MHz$, $\rho = 1Mbps$, $L = 4$, $\lambda = 0.03$, and $\beta = 4$, unless otherwise noted. In addition, following M-Zipf parameters are used: $\gamma = 1.2$, $q = -0.65$, $\gamma_k^{in} = 2.5$, and $q_k^{in} = 3$ for all k. For comparison purposes, the probabilistic caching policy optimized for one-shot requests only, i.e., $L = 1$, in [12] is used.

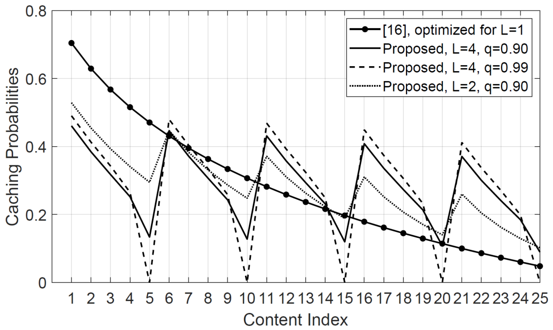


Fig. 5. Caching probabilities for each content.

A. Probabilistic Caching for Consecutive User Demands

Fig. 5 shows the caching probabilities of all contents in $\mathcal{F}$. In Fig. 5, the caching probabilities obtained by the comparison scheme in [12] depend on individual content popularities, i.e., Zipf distribution. On the other hand, the caching probabilities of the proposed scheme are largely influenced by the popularity model for consecutive content requests characterized by ν. The caching probabilities of the contents in $\mathcal{G}_1$ are the highest among all categories, and the contents in $\mathcal{G}_5$ have the smallest caching probabilities. For example, in the proposed scheme, even though f_5 is much larger than f_{11}, p_5 is smaller than p_{11} because c_{11} belongs to the category of $\mathcal{G}_1$. This phenomenon is consistent with the observation in [28] which is that the view rate of each content is very similar to that of its top referrer one; therefore, in this example, p_{11} is relatively as large as p_1 which is the top referrer in $\mathcal{G}_1$. Especially when $L = 4$, contents in the same category have more similar caching probabilities; in other words, *temporary preference* becomes stronger than *global preference*. Meanwhile, when $L = 2$, contents 1–5 have evidently larger caching probabilities than most of the other contents in the same category; in this case, the temporary preference is weakened compared to the case with $L = 4$. Definitely, as ν grows, the temporary preference becomes stronger; therefore, the popularity of each content becomes more similar to that of its top referrer content.

Fig. 6 shows the plots of cache miss rates versus density of caching helpers (i.e., λ). As λ grows, the user can easily

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

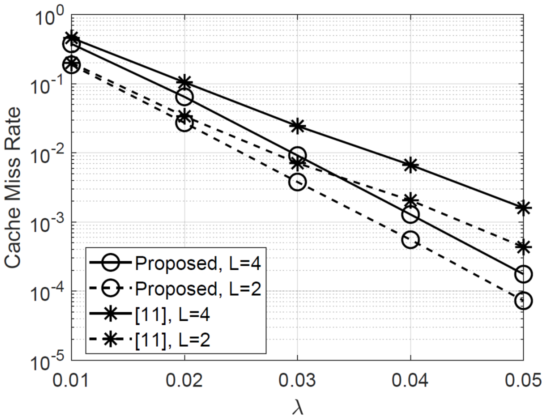


Fig. 6. The cache hit rate vs. λ.

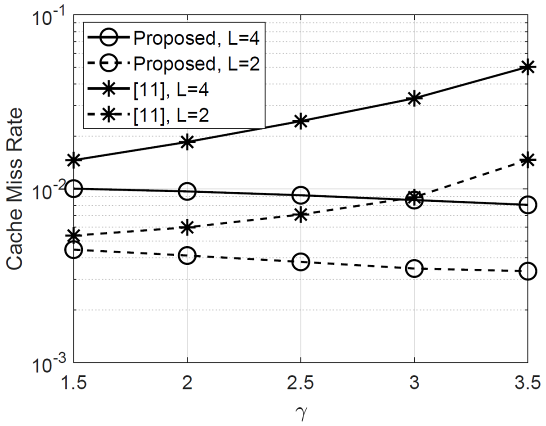


Fig. 7. The cache miss rate vs. γ.

find the caching helpers that store the requested contents; therefore, the cache miss rates decrease. Overall, the proposed caching policy outperforms the comparison scheme; however, the performance gain increases as λ decreases. The reason is that when λ is large, there are sufficiently many helpers that can provide the requested contents without the caching scheme being optimized for consecutive user demands. Additionally, the performance gain of the proposed caching compared to the comparison scheme increases as L grows. This is consistent with the results in Fig. 5, which show that the caching probabilities of the proposed scheme and the comparison scheme have more differences with larger L.

The impacts of the skew factor of the M-Zipf distribution for the first content request (i.e., γ) and the cache size (i.e., M) are shown in Figs. 7 and 8, respectively. Interestingly, the cache miss rate of the proposed scheme decreases, but that of the comparison method increases as γ grows in Fig. 7. A large γ significantly increases the difference in global popularity among all contents, and it makes the global popularity profile deviate considerably from the temporary content popularity. For example, as γ increases, f_5 grows but f_{16} decreases. In this case, the comparison scheme is more likely to cache c_5 rather than c_{16}. However, when the user consecutively requests multiple contents, the view rate of c_{16} would be also as large as that of c_1, resulting in decline of the cache hit rate of the comparison scheme. Meanwhile, the proposed scheme captures the effect of the temporary preference so that it is still more likely to cache c_{16} than c_5; therefore, its performance still increases slightly with γ. In Fig. 8, performance gaps between the proposed and comparison techniques increase as M decreases. The reason is that if M is not large, contents with a small global popularity are not likely to be stored in the caching helpers for the comparison scheme in [12], even though their popularity can be boosted by the temporary preference.

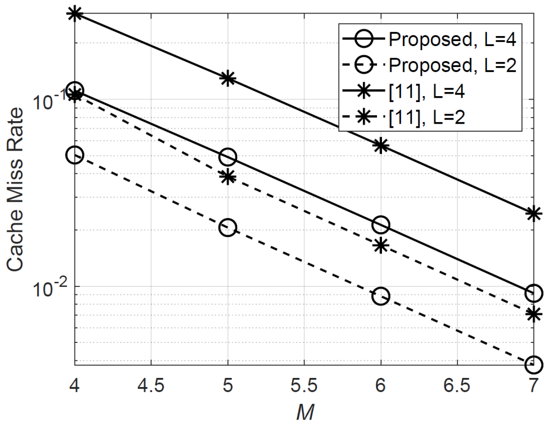


Fig. 8. The cache miss rate vs. M.

B. Dynamic Delivery for Consecutive User Demands

In this subsection, the delivery latency parameters are assumed as follows: $t_o = 0.1$, $\tau_N = 1.0$, and $\tau_B = 2.0$. To observe the effects of switches of caching helpers, we consider the situation where there are multiple helper candidates for delivering consecutively requested contents. Also, outage events due to channel fading are considered; therefore, a larger $\lambda = 0.05$ is used in this section unless otherwise noted. In order to show the advantages of the proposed caching and delivery schemes separately, simulations are performed using the following approaches:

- ‘Proposed, Proposed’: The proposed caching and dynamic helper association methods are applied; these are described in Sections III and IV, respectively.
- ‘Proposed, Nearest’: The proposed caching method is applied; however, the nearest caching helper among the helpers having the desired content is chosen for delivering the content to the user, which is a traditional method for helper association.
- ‘ [12], Proposed’: Caching is optimized for a one-shot request as in [12]; however, the proposed dynamic helper association is applied.
- ‘ [12], Nearest’: Caching is optimized for a one-shot request, and the nearest caching helper among helpers having the desired content is chosen for delivering the content.

Figs. 9 and 10 show the expected delay performances versus λ when $\nu = 0.90$ and $\nu = 0.99$, respectively. Overall, the proposed technique outperforms other comparison approaches,

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

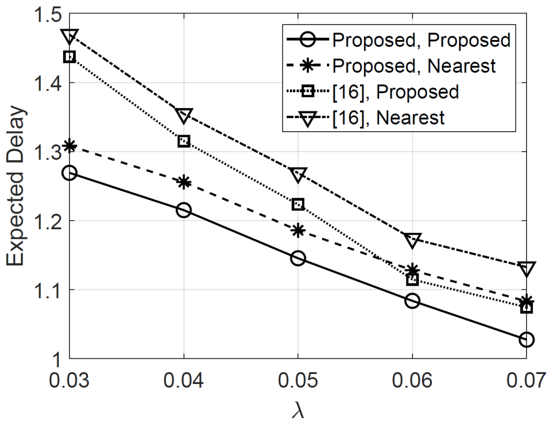


Fig. 9. Expected delays vs. λ when $\nu = 0.90$.

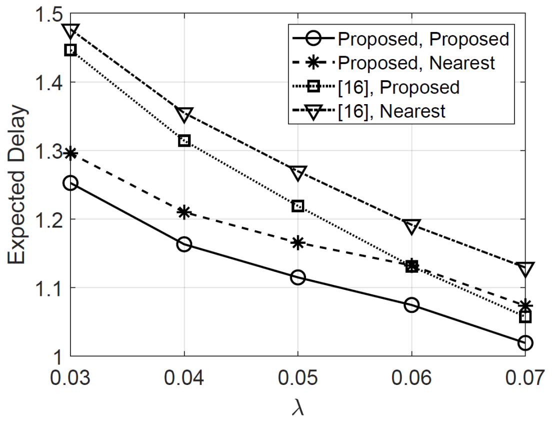


Fig. 10. Expected delays vs. λ when $\nu = 0.99$.

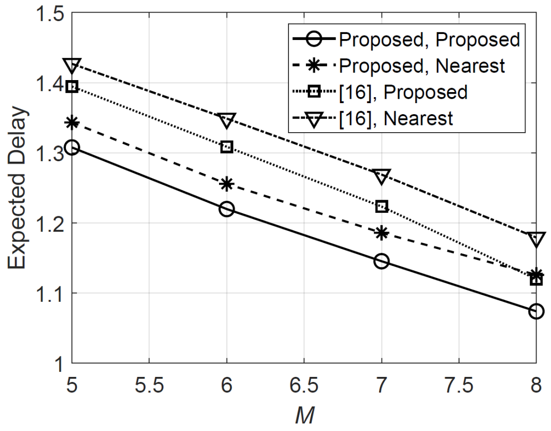


Fig. 11. Expected delays vs. M.

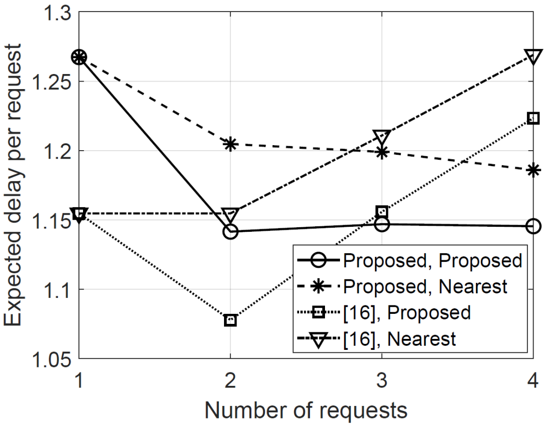


Fig. 12. Expected delays vs. L.

and the expected delays decrease as λ increases. Comparing the proposed scheme with ‘Proposed, Nearest’, it can be seen that the performance gains of the proposed scheme slightly increase as λ grows. Since there are many caching helpers around the user with a large λ, it is highly probable for the user to find a helper having multiple requested contents; therefore, the delays caused by switches of helper associations can be reduced further when λ is large. When there is a small number of caching helpers around the user, the comparison method optimized for only one-shot requests makes it difficult for the user to find the helper that can provide multiple contents in the request sequence; therefore, the delay performance gains over ‘ [12], Proposed’ increase as λ decreases. In addition, if ν increases, i.e., the temporary user preference becomes stronger, bigger performance gains of the proposed caching method are achieved.

Fig. 11 shows the impacts of M on the expected delays; the delay performance gains of the proposed scheme increase as M grows, compared to other methods. When M is large, a helper is likely to store contents in the same category in the proposed caching method; therefore, it can reduce the latency required for switches of caching helpers. On the other hand, a very small M requires frequent updates of helper associations and finally even the proposed scheme generates comparable delivery latency to comparison methods with a small M.

Fig. 12 shows the plots of the expected delay per content request versus L, when the caching method is optimized for $L = 4$. Since users can request any number of contents in $\{1, \cdots, L\}$, although the caching method is optimized for L consecutive content requests, the usefulness of the proposed scheme can be verified by investigating its performance in the situation in which the user requests l contents in a row, where $l < L$. Since the caching method in [12] is optimized for $L = 1$, the expected delay of [12] is smaller than that of the proposed caching when $L = 1$; however, the proposed caching outperforms the comparison scheme with $L \geq 3$. Meanwhile, the proposed delivery policy is not optimized at a specific value L. Therefore, the performance gain obtained by the proposed delivery scheme can be observed immediately when the user requests more than one content. Note that Fig. 12 shows the expected delay per request so that total delay of the type-L user incurred during the sequence of consumed contents becomes L times the result in Fig. 12. It means that the total delay generated while consuming $l \geq 3$ contents can be reduced further by using the proposed caching and delivery methods. Thus, we can conclude that the proposed scheme is useful to support heavy users consuming many contents at once.

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

VI. Concluding Remarks

This paper proposes a probabilistic caching policy and dynamic helper associations for content delivery when users request different numbers of categorized content. Unlike the existing studies that consider global user preference, this paper introduces temporary user preference. Temporary user preference is an essential characteristic of multimedia services where users continuously consume multiple contents in an identical related content list in a row, and the contents in this short sequence of consumption are strongly correlated. Given categorized contents (i.e., related content lists), the probabilistic caching is optimized to maximize the minimum cache hit rates of all users. In addition, in the delivery phase, a DP-based helper association method is used to minimize the expected delivery latency. Since multiple content consumption is considered, delay incurred by switching the caching helper for receiving the next content is also captured.

References

[1] *Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2018–2023*, Cisco, San Jose, CA, USA, Mar. 2020.

[2] X. Cheng, J. Liu, and C. Dale, "Understanding the characteristics of Internet short video sharing: A YouTube-based measurement study," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1184–1194, Aug. 2013.

[3] 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*, Orlando, FL, USA, Mar. 2012, pp. 1–15.

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

[5] 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 Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[6] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.

[7] M. Ji, G. Caire, and A. F. Molisch, "Wireless Device-to-Device caching networks: Basic principles and system performance," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.

[8] J. Kim, G. Caire, and A. F. Molisch, "Quality-aware streaming and scheduling for device-to-device video delivery," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2319–2331, Aug. 2016.

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

[10] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, and L. Hanzo, "Distributed caching for data dissemination in the downlink of heterogeneous networks," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3553–3568, Oct. 2015.

[11] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 3358–3363.

[12] S. H. Chae and W. Choi, "Caching placement in stochastic wireless caching helper networks: Channel selection diversity via caching," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6626–6637, Oct. 2016.

[13] M. Choi, J. Kim, and J. Moon, "Wireless video caching and dynamic streaming under differentiated quality requirements," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1245–1257, Jun. 2018.

[14] M. Choi, A. No, M. Ji, and J. Kim, "Markov decision policies for dynamic video delivery in wireless caching networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 12, pp. 5705–5718, Dec. 2019.

[15] W. Jing, X. Wen, Z. Lu, and H. Zhang, "User-centric delay-aware joint caching and user association optimization in cache-enabled wireless networks," *IEEE Access*, vol. 7, pp. 74961–74972, 2019.

[16] 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," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5506–5519, Aug. 2019.

[17] M. Choi, A. F. Molisch, and J. Kim, "Joint distributed link scheduling and power allocation for content delivery in wireless caching networks," 2019, *arXiv:1911.13010*. [Online]. Available: http://arxiv.org/abs/1911.13010

[18] Y. Pan, C. Pan, H. Zhu, Q. Zeeshan Ahmed, M. Chen, and J. Wang, "On consideration of content preference and sharing willingness in D2D assisted offloading," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 4, pp. 978–993, Apr. 2017.

[19] Y. Guo, L. Duan, and R. Zhang, "Cooperative local caching under heterogeneous file preferences," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 444–457, Jan. 2017.

[20] M.-C. Lee and A. F. Molisch, "Individual preference aware caching policy design for energy-efficient wireless D2D communications," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–7.

[21] B. Chen and C. Yang, "Caching policy for cache-enabled D2D communications by learning user preference," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6586–6601, Dec. 2018.

[22] W. Hoiles, O. N. Gharehshiran, V. Krishnamurthy, N.-D. Dao, and H. Zhang, "Adaptive caching in the YouTube content distribution network: A revealed preference game-theoretic learning approach," *IEEE Trans. Cognit. Commun. Netw.*, vol. 1, no. 1, pp. 71–85, Mar. 2015.

[23] B. N. Bharath, K. G. Nagananda, and H. V. Poor, "A learning-based approach to caching in heterogenous small cell networks," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1674–1686, Apr. 2016.

[24] B. N. Bharath, K. G. Nagananda, D. Gunduz, and H. V. Poor, "Caching with time-varying popularity profiles: A learning-theoretic perspective," *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 3837–3847, Sep. 2018.

[25] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Temporal locality in today's content caching: Why it matters and how to model it," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 5, pp. 5–12, Nov. 2013.

[26] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, Oct. 2009.

[27] R. Zhou, S. Khemmarat, L. Gao, and H. Wang, "Boosting video popularity through recommendation systems," in *Databases Social Netw.*, New York, NY, USA, 2011, pp. 13–18.

[28] R. Zhou, S. Khemmarat, and L. Gao, "The impact of YouTube recommendation system on video views," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, New York, NY, USA, 2011, pp. 404–410.

[29] D. K. Krishnappa, M. Zink, C. Griwodz, and P. Halvorsen, "Cache-centric video recommendation: An approach to improve the efficiency of youtube caches," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 4, Jun. 2015, Art. no. 48.

[30] C. Li, J. Liu, and S. Ouyang, "Large-scale user behavior characterization of online video service in cellular network," *IEEE Access*, vol. 4, pp. 3675–3687, 2016.

[31] B. McClanahan and S. S. Gokhale, "Interplay between video recommendations, categories, and popularity on YouTube," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput.*, San Francisco, CA, USA, Aug. 2017, pp. 1–7.

[32] M. Choi, D. Kim, D.-J. Han, J. Kim, and J. Moon, "Probabilistic caching policy for categorized contents and consecutive user demands," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[33] M. Choi, A. F. Molisch, D.-J. Han, J. Kim, and J. Moon, "Cache allocations for consecutive requests of categorized contents: Service Provider's perspective," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.

[34] D. Liu and C. Yang, "Caching at base stations with heterogeneous user demands and spatial locality," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1554–1569, Feb. 2019.

[35] M.-C. Lee, A. F. Molisch, N. Sastry, and A. Raman, "Individual preference probability modeling and parameterization for video content in wireless caching networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 676–690, Apr. 2019.

[36] M. Ji, G. Caire, and A. F. Molisch, "The throughput-outage tradeoff of wireless one-hop caching networks," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6833–6859, Dec. 2015.

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.

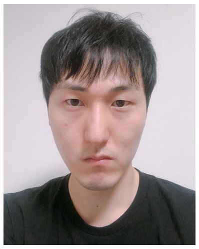
Minseok Choi (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2011, 2013, and 2018, respectively. He was a Visiting Post-Doctoral Researcher in electrical and computer engineering with the University of Southern California (USC), Los Angeles, CA, USA, and a Research Professor in electrical engineering with Korea University, Seoul, South Korea. He has been an Assistant Professor with Jeju National University, Jeju, South Korea, since 2020. His research interests include wireless caching networks, stochastic network optimization, and machine learning in wireless networks.

Andreas F. Molisch (Fellow, IEEE) received the Dipl.Ing., Ph.D., and Habilitation degrees from the Technical University Vienna, Austria, in 1990, 1994, and 1999, respectively.

He spent the ten years in industry, at FTW, AT&T (Bell) Laboratories, and Mitsubishi Electric Research Labs (where he rose to Chief Wireless Standards Architect). In 2009, he joined the University of Southern California (USC), Los Angeles, CA, USA, as a Professor, and founded the Wireless Devices and Systems (WiDeS) Group. In 2017, he was appointed to the Solomon Golomb-Andrew and Erna Viterbi Chair. His research interests include revolve around wireless propagation channels, wireless systems design, and their interaction. Recently, his main interests have been wireless channel measurement and modeling for 5G and beyond 5G systems, joint communication-caching-computation, hybrid beamforming, UWB/TOA-based localization, and novel modulation/multiple access methods. Overall, he has published four books (among them the textbook *Wireless Communications*, currently in its second edition), 21 book chapters, 260 journal articles, and 360 conference papers. He is also the inventor of 60 granted (and more than 20 pending) patents, and coauthor of some 70 standards contributions.

Dr. Molisch is a fellow of the National Academy of Inventors, a fellow of the AAAS, a fellow of the IET, an IEEE Distinguished Lecturer, and a member of the Austrian Academy of Sciences. He has received numerous awards, among them the IET Achievement Medal, the Technical Achievement awards of IEEE Vehicular Technology Society (the Evans Avant-Garde Award) and the IEEE Communications Society (the Edwin Howard Armstrong Award), and the Technical Field Award of the IEEE for Communications, the Eric Sumner Award. He has been an editor of a number of journals and special issues, the general chair, the technical program committee chair, or the symposium chair of multiple international conferences, as well as the chairman of various international standardization groups.

Dong-Jun Han (Graduate Student Member, IEEE) received the B.S. degrees in mathematics and electrical engineering, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree. His research interests include distributed machine learning and information theory.



Dongjae Kim (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2011, 2013, and 2020, respectively. His research interests include transceiver design, and 5G communications.

Joongheon Kim (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science and engineering from Korea University, Seoul, South Korea, in 2004 and 2006, respectively, and the Ph.D. degree in computer science from the University of Southern California (USC), Los Angeles, CA, USA, in 2014.

He was with LG Electronics, Seoul, from 2006 to 2009, InterDigital, San Diego, CA, USA, in 2012, Intel Corporation, Santa Clara, Silicon Valley, CA, USA, from 2013 to 2016, and Chung-Ang University, Seoul, from 2016 to 2019. He has been with the School of Electrical Engineering, Korea University, since 2019, where he is currently an Assistant Professor. He internationally published more than 80 journals, 110 conference papers, and six book chapters. He also holds more than 50 granted patents. He was a recipient of Annenberg Graduate Fellowship with his Ph.D. admission from USC in 2009, the Intel Corporation Next Generation and Standards (NGS) Division Recognition Award in 2015, the Haedong Young Scholar Award by the Korea Institute of Communication and Information Sciences (KICS) in 2018, the IEEE Vehicular Technology Society (VTS) Seoul Chapter Award in 2019, the Outstanding Contribution Award by KICS in 2019, the Gold Paper Award from IEEE Seoul Section Student Paper Contest in 2019, and IEEE SYSTEMS JOURNAL Best Paper Award in 2020. He serves as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

Jaekyun Moon (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA. From 1990 to 2009, he was a Faculty Member with the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, MN, USA. He consulted as a Chief Scientist with DSPG, Inc., from 2004 to 2007. He was also a Chief Technology Officer with Link-A-Media Devices Corporation. He is currently Professor with the School of Electrical Engineering, KAIST. His research interests include distributed and decentralized storage, communication, and machine intelligence. He was a recipient of IBM Faculty Development awards and IBM Partnership awards. He also received the National Storage Industry Consortium Technical Achievement Award for the invention of the maximum transition run code, a widely used error-control/modulation code in commercial storage systems. He was awarded the McKnight Land-Grant Professorship from the University of Minnesota. He served as the Program Chair for the 1997 IEEE Magnetic Recording Conference. He was also the Chair of the Signal Processing for Storage Technical Committee of the IEEE Communications Society. He served as a Guest Editor for the 2001 IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Issue on Signal Processing for High Density Recording. He also served as an Editor for the IEEE TRANSACTIONS ON MAGNETICS in the area of signal processing and coding from 2001 to 2006.

Authorized licensed use limited to: Korea University. Downloaded on April 09,2021 at 00:56:50 UTC from IEEE Xplore. Restrictions apply.