Mobile Device-to-Device (D2D) Content Delivery Networking: A Design and Optimization Framework

Hye Joong Kang and Chung Gu Kang

Abstract: **We consider a mobile content delivery network (mCDN) in which special mobile devices designated as caching servers (caching-server device: CSD) can provide mobile stations with popular contents on demand via device-to-device (D2D) communication links. On the assumption that mobile CSD's are randomly distributed by a Poisson point process (PPP), an optimization problem is formulated to determine the probability of storing the individual content in each server in a manner that minimizes the average caching failure rate. Further, we present a low-complexity search algorithm, optimum dual-solution searching algorithm (ODSA), for solving this optimization problem. We demonstrate that the proposed ODSA takes fewer iterations, on the order of $O(\log N)$ searches, for caching N contents in the system to find the optimal solution, as compared to the number of iterations in the conventional subgradient method, with an acceptable accuracy in practice. Furthermore, we identify the important characteristics of the optimal caching policies in the mobile environment that would serve as a useful aid in designing the mCDN.**

Index Terms: **Caching probability, caching server device, device-to-device (D2D) communication, mobile contents distributed network (mCDN), Poisson point process (PPP).**

I. INTRODUCTION

Ever-increasing demands for multimedia contents have had a critical impact on network capacity in serving content to end-users with high availability and performance. In particular, the pure client-server model is highly inefficient for content distribution, as it suffers from performance degradation owing to a bottleneck problem at the single server, while overloading the network to serve physically remote clients. To deal with the issues in the client-server model, a content delivery network (CDN) has been introduced as a distributed system of proxy servers deployed in multiple data centers across the network. In CDN, popular content on servers subject to frequent demand for delivery is stored in proxy servers, placed at multiple locations close to the end users, so as to offload the network and server.

Recently, a concept of CDN has been extended to mobile networks in which wireless access nodes, such as an access point and a base station, can be used as caching servers for mobile users at their end [1]. Furthermore, individual mobile devices themselves can be caching servers as well, since they are directly connected to each other by establishing device-to-device (D2D) communication links [2]. The main advantage of D2D communication in a cellular system is its spatial reuse capability gained by enabling multiple direct links between two near-by devices at the same time. In the current commercial system, however, D2D communication links would be rarely activated owing to the fact that most traffic is originated mainly by the client/server model-based content delivery architecture, i.e., overloading the access network. If mobile devices themselves serve as content caching servers, mobile CDN (mCDN) traffic becomes enormous within each cell, making D2D communication essential for an aggressive spatial reuse gain. Furthermore, a mobile device as a caching server, designated as a caching-server device (CSD), would reduce the traffic load of a backbone network as intended by CDN, without incurring an extra cost of deploying and maintaining proxy servers.

D2D communication-enabled CSD's with content-caching capability differ in several ways from conventional caching servers in the network. First of all, their service coverage can be limited by transmission power over a wireless link. Second, there would be a large number of CSD's, which are subject to mobility, implying that the availability of contents would be spatially broad yet random. Third, caching capability is limited by a physical nature of mobile devices, e.g., a limited storage capacity. Due to these characteristics, content-caching policies must differ from the conventional ones.

We assume that all CSD's are capable of storing the contents that are obtained upon its own request or overhearing what other devices or serving base station (BS) have transmitted. Whenever a specific content is requested by a device, it can be provided by its neighbor CSD with the same content if available. Unless a device cannot find a neighbor CSD with the specific content in demand, it is subject to caching failure, implying that the requested content must be provided over the cellular link. In order to reduce the caching failure rate, CSD's must cache the contents as much as possible within its own physical capability. As every CSD has a limited storage, however, it would cache only the relatively popular contents, while ignoring ones in less demand. For example, suppose that only a single content can be stored in each CSD due to its physical limit. Then, every CSD would cache the most popular one, which will be requested most frequently. Is it really necessary for every CSD to cache the same content all the time in the current scenario? In case that a large number of mobile CSD's are randomly distributed in the vicinity, some CSD's may be allowed to have another popular content rather than caching the same content in all CSD's. Since additional contents are spatially available among the near-by CSD's, the caching capability is further improved, thereby reducing a

Manuscript received December 28, 2013; approved for publication by Huang, Polly, Division III Editor, July 27, 2014.

This work was supported by the ICT R&D program of MSIP/IITP [14-000-04-001, Development of 5G Mobile Communication Technologies for Hyper-connected Smart Services].

H. J. Kang is with the Department of IT Convergence, Korea University in Seoul, Korea, Republic of, email: dreamftr@korea.ac.kr.

C. G. Kang is with the Department of Electrical Engineering, Korea University in Seoul, Korea, Republic of, email: ccgkang@korea.ac.kr.

Digital object identifier 10.1109/JCN.2014.000095

1229-2370/14/$10.00 © 2014 KICS

caching failure. In fact, all CSD's are coordinated to determine which contents are cached by the different CSD's rather than caching the most popular yet redundant contents in every CSD. Therefore, the physical storage can be effectively utilized to minimize the caching failure rate. However, as CSD's may be steadily on move, local CSD coordination is not effective. Instead, whether to cache or not must be determined for each content in an average sense by taking a distribution of CSD's and popularity of individual contents into account. Since it would require too much signaling overhead and processing complexity to determine whether to cache each content or not in each CSD, we consider a centralized control approach. It employs a central server in which information on CSD density and content popularity is used to determine the average caching probabilities of individual contents. Based up these content caching probabilities (known to all CSD's as broadcast from a base station), each CSD builds a list of the contents to be cached. Whenever any content in the list is available over the cellular link or D2D links, then it will be cached in the storage.

In this paper, our objective is to provide a design optimization framework that determines the average caching probabilities of individual contents in the central server. We first formulate an optimization problem to determine the optimum caching probabilities by minimizing the average caching failure probability. Then, we propose a low-complexity solution approach, the optimum dual-solution searching algorithm (ODSA) to solve our optimization problem. Because ODSA converts the continuous dual solution region into discrete solution regions, it allows a full search to determine the optimum caching probability. In fact, it takes fewer iterations, on the order of $O(\log N)$ searches, for caching N contents in the system to find the optimal solution than the number of iterations in the conventional subgradient method [4], with an acceptable accuracy in practice. The performance of the proposed policy with the optimal caching probability is compared with that of other caching policies, one with an equal caching probability (EP caching policy) and the other with high-priority-first selection (HPF caching policy). Based on conclusions from our optimization framework, we provide a practical design principle that can improve its performance when the demand statistics are known a priori.

This paper is organized as follows. In Section II, we present the system model and formulate the optimization problem for finding the caching probability for each content. In Section III, we propose ODSA. In Section IV, numerical results are presented to evaluate the complexity of the proposed algorithm, and the performance of the different caching policies is evaluated. Finally, conclusions are drawn in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

We assume that CSD's are randomly distributed within a disc area with radius d centered at a reference receiver that requests a specific content. In particular, we assume a distribution of L CSD's by a Poisson point process (PPP) with average intensity λ (arrivals/m^2) [3], i.e., the probability that l CSD's exist within a distance of d from the reference content-requesting device is given as

$$f_L(l;d) = \frac{(\pi d^2 \lambda)^l}{l!} e^{-\pi d^2 \lambda}. \quad (1)$$

Let $\mathbf{I}$ denote a set of N contents, one of which is requested by a device at an arbitrary location. Note that the caching server device is different from the device in general, as it is capable of storing the contents (up to M contents) that are obtained upon its own request or overhearing what other devices or serving BS have transmitted and providing these contents to any neighbor device via a direct link upon the request. In fact, both cellular and D2D communication links are used to share contents among the CSD's forming an mCDN. We assume that the CSD stores the content of index i with a probability of p_i, when the corresponding content was obtained upon its own request or overheard when other devices or serving BS have transmitted it. More specifically, we assume that the pre-specified contents transmitted by one CSD or BS can be overheard by all authorized CSD's. Furthermore, we assume that $\{p_i\}$ is determined in a central server with a periodic or event-driven manner, and broadcasted to all CSD's. Once $\{p_i\}$ is updated by the central server, each CSD determines which contents to store by the given probability of $\{p_i\}$. More specifically, based up these content caching probabilities, each CSD builds a list of the contents to be cached. A design of the specific algorithm for listing up the content to be cached subject to the given caching probabilities is beyond the scope of our current work. Whenever any content in the list is available over the cellular link or D2D links, then it will be cached in the storage.

Note that CSD's with content i are distributed by a PPP with average intensity λp_i. Let D be a random variable to denote the distance between a reference receiver that is requesting content i and the closest CSD with content i. In other words, as shown in Fig. 1, this corresponds to the situation in which there exists no CSD with content i within a disc of radius D centered at the reference receiver. Then, the cumulative distribution function (CDF) of D, $F_D^{(i)}(d)$, is given as

$$F_D^{(i)}(d) = 1 - \Pr(D > d) = 1 - f_L(0;d) = 1 - e^{-\pi d^2 \lambda p_i}. \quad (2)$$

Let $r_{\max}$ denote the maximum transmission range of the CSD, which is determined by the system parameters in physical and medium access control layers. A content caching failure is defined as an event in which the reference receiver cannot find any CSD with the requesting content within a fixed distance $r_{\max}$. The caching failure rate of each content i for a reference receiver that is R meters apart from the closest CSD with content i, denoted as $P_f^{(i)}(r_{\max})$, is given by the following probability

$$P_f^{(i)}(r_{\max}) = 1 - F_D^{(i)}(R) = e^{-\pi r_{\max}^2 \lambda p_i}. \quad (3)$$

Meanwhile, let g_i denote the probability that content i is requested by a device. We assume that the content with a smaller index has a larger probability of requesting the content, i.e., $g_i \geq g_j$ if $i < j$. Given the content caching probabilities $\mathbf{p} = \{p_1, p_2, \cdots, p_N\}$, the average content caching failure rate is given by the weighted sum of individual content caching failure probabilities for all contents, each weighted by the probability

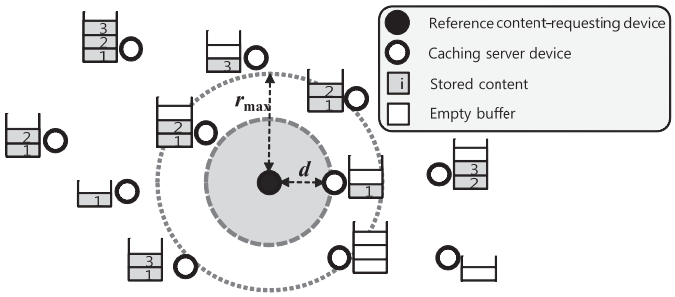


Fig. 1. Distribution of caching-server devices (CSD's): Illustration.

of requesting the corresponding content, as follows

$$\bar{f}(\mathbf{p}) = \sum_{i\in\mathbf{I}} \left\{1 - F_D^{(i)}(r_{\max})\right\} g_i = \sum_{i\in\mathbf{I}} g_i e^{-\pi\lambda r_{\max}^2 p_i}. \quad (4)$$

B. Problem Formulation

We intend to determine the caching probability for each content, $\mathbf{p} = \{p_1, p_2, \cdots, p_N\}$, that minimizes the average caching failure probability in (4) when each device caches content *i* with a probability of p_i. Note that the average number of contents cached in CSD is $\sum_{i\in\mathbf{I}} p_i$. Then, let us impose a constraint that the average number of contents cached in CSD cannot exceed *M*, i.e.,

$$\sum_{i\in\mathbf{I}} p_i \le M. \quad (5)$$

In other words, *M* corresponds to the maximum number of contents that can be cached in a CSD on average. The caching probability for each content is determined in a central server, which must steadily update the CSD density λ and probability of requesting the content, $\{g_i\}$. Now, our optimization problem can be formulated as

$$\min_{\mathbf{p}=\{p_1,p_2,\cdots,p_N\}} \sum_{i\in\mathbf{I}} g_i e^{-\pi\lambda r_{\max}^2 p_i} \quad (6)$$

subject to

$$\sum_{i\in\mathbf{I}} p_i - M \le 0, \quad (7)$$

$$p_i - 1 \le 0,\ i = 1, 2, \cdots, N, \quad (8)$$

$$-p_i \le 0,\ i = 1, 2, \cdots, N. \quad (9)$$

This is a constrained non-linear convex optimization problem, which can be solved by a conventional iterative approach, e.g., the subgradient method [4]. However, we note that the convergence rate of the subgradient method depends mainly on the step size in an iteration formula, typically requiring a large number of iterations to obtain the optimal solution within the given accuracy. In order to circumvent the complexity of an iterative approach, therefore, we propose a reduced complexity search algorithm, ODSA, in the next section.

Table 1. Notations.

Symbol	Description
D	A random variable to denote the distance between a reference content-requesting device and CSD
$r_{\max}$	The maximum transmission range
N	The number of contents
M	Storage capacity
$\mathbf{I}$	A set of N contents, one of which is requested by a device at an arbitrary location.
λ	Intensity of CSD
g_i	Request probability of content *i*
p_i	Caching probability of content *i*
$f_L(l; d, \lambda)$	The probability density function (PDF) of the number of CSD's, *L*, for the CSD intensity of λ and $D = d$.
$F_D^{(i)}(d)$	The cumulative distribution function (CDF) of *D* for content *i*
$P_f^{(i)}(r_{\max})$	The average caching failure probability of content *i* with the maximum transmission range of $r_{\max}$

III. OPTIMUM DUAL-SOLUTION SEARCHING ALGORITHM

Note that the solution to our optimization problem (6)–(9) is a vector $\mathbf{p} = \{p_1, p_2, \cdots, p_N\}$ with real elements. This type of problem can be solved using iterative schemes such as the subgradient method [4]. The Appendix presents the solution approach based on the subgradient method. For a convex or quasi-convex problem, these iterative schemes guarantee an optimal solution, but may involve enormous computational complexity. This is because the optimal step size for updating decision variables is difficult to determine, so the number of iterations needed to converge varies with other parameters. However, in our optimization problem (6)–(9), we find a non-iterative solution approach with a fixed but low computational complexity. We propose a method for converting this problem into one of searching for the optimal dual solution over a finite set. Toward this end, we consider the Karush-Kuhn-Tucker (KKT) conditions for our optimization problem (6)–(9) as follows.

The gradient of the Lagrangian with respect to p_i vanishes

$$\frac{\partial L}{\partial p_i} = -\pi\lambda r_{\max}^2 e^{-\pi\lambda r_{\max}^2 p_i} g_i + \mu + \gamma_i - \sigma_i = 0 \quad (10)$$

where

$$L(\cdot) = \sum_{i\in\mathbf{I}} g_i e^{-\pi\lambda r_{\max}^2 p_i} + \mu\left(\sum\nolimits_{i\in\mathbf{I}} p_i - M\right) + \sum_{i=1}^{N} \gamma_i (p_i - 1) - \sum_{i=1}^{N} \sigma_i p_i.$$

- Primal conditions:

$$\sum_{i\in\mathbf{I}} p_i - M \le 0, \quad (11)$$

$$p_i - 1 \le 0, \ -p_i \le 0, \ i = 1, 2, \cdots, N. \tag{12}$$

- Dual conditions:

$$\mu \ge 0, \tag{13}$$

$$\gamma_i \ge 0, \sigma_i \ge 0, \ i = 1, 2, \cdots, N. \tag{14}$$

- Complementary slackness:

$$\mu \left(\sum_{i \in \mathbf{I}} p_i - M \right) = 0, \tag{15}$$

$$\gamma_i \left(p_i - 1 \right) = 0, \ i = 1, 2, \cdots, N, \tag{16}$$

$$\sigma_i p_i = 0, \ i = 1, 2, \cdots, N. \tag{17}$$

Solving (10) for p_i,

$$\begin{aligned} p_i(\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma}) &= \frac{1}{\pi \lambda r_{\max}^2} \log \frac{\pi \lambda r_{\max}^2 g_i}{\mu + \gamma_i - \sigma_i} \\ &= \frac{1}{\pi \lambda r_{\max}^2} \log \frac{\pi \lambda r_{\max}^2 g_i}{\xi_i} \end{aligned} \tag{18}$$

where $\xi_i = \mu + \gamma_i - \sigma_i$, $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \cdots, \gamma_N)$, and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \cdots, \sigma_N)$. As $\sum_{i \in \mathbf{I}} p_i = M$ must be satisfied at the optimal solution, μ is not necessarily zero in (15). Furthermore, $\gamma_i = 0$ for $p_i \ne 1$ in (16) and $\sigma_i = 0$ for $p_i \ne 1$ in (17). Depending on the value of p_i, ξ_i can be given as

$$\xi_i = \begin{cases} \mu - \sigma_i & \text{if} \quad p_i = 0; \\ \mu & \text{if} \quad 0 < p_i < 1; \\ \mu + \gamma_i & \text{if} \quad p_i = 1. \end{cases} \tag{19}$$

Lemma 1: For an arbitrary value of μ, $\boldsymbol{\sigma}$, and $\boldsymbol{\gamma}$ are given as

$$\sigma_i = \left[\mu - \pi \lambda r_{\max}^2 g_i \right]^+, \ i = 1, 2, \cdots, N, \tag{20}$$

$$\gamma_i = \left[\pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2} - \mu \right]^+, \ i = 1, 2, \cdots, N \tag{21}$$

where $[x]^+ = \max(0, x)$.

Proof: It is immediately required from (18) that $\xi_i = \pi \lambda r_{\max}^2 g_i$ for some *i* such that $p_i = 0$. Since $\xi_i = \mu - \sigma_i$ for $p_i = 0$ from (19), therefore, $\sigma_i = \mu - \pi \lambda r_{\max}^2 g_i$ in this case. Meanwhile, as $\sigma_i = 0$ and $\mu < \pi \lambda r_{\max}^2 g_i$ for some *i* such that $p_i > 0$. Therefore σ_i can be expressed as $[\mu - \pi \lambda r_{\max}^2 g_i]^+$, proving (20). From (18), furthermore, $\xi_i = \pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2}$ for some i such that $p_i = 1$. As $\xi_i = \mu + \gamma_i$ for $p_i = 1$ from (19), $\gamma_i = \pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2} - \mu$ in this case. Meanwhile, as $\gamma_i = 0$ and $\mu > \pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2}$ for some *i* such that $p_i < 1$, γ_i can be expressed as $[\pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2} - \mu]^+$, proving (21). □

As $\boldsymbol{\sigma}$ and $\boldsymbol{\gamma}$ are given in terms of μ by Lemma 1, we can represent $p_i(\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma})$ in (18) as $p_i(\mu)$ in short. As μ is known for $p_i = 1$ or $p_i = 0$ in Lemma 1, we only need to determine μ for $0 < p_i < 1$. Let $\mathbf{I}_0$ denote a set of contents that is never stored in any device, i.e., $\mathbf{I}_0 = \{i | p_i = 0, \ i \in \mathbf{I}\}$. Similarly, define $\mathbf{I}_1$ to denote a set of contents that is always stored in all devices, i.e., $\mathbf{I}_1 = \{i | p_i = 1, \ i \in \mathbf{I}\}$. As $\mathbf{I}_0$ and $\mathbf{I}_1$ depend on the given dual solution μ, they can be re-defined as functions of μ by the fact that $\mu < \pi \lambda r_{\max}^2 g_i$ for $p_i > 0$ and $\mu > \pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2}$ for $p_i < 1$

$$\mathbf{I}_0(\mu) = \{i | \pi \lambda r_{\max}^2 g_i \le \mu, i \in [1, N]\}, \tag{22}$$

$$\mathbf{I}_1(\mu) = \{i | \pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2} \ge \mu, \ i \in [1, N]\}. \tag{23}$$

In Theorem 1, we will show that given $\mathbf{I}_0(\mu)$ and $\mathbf{I}_1(\mu)$ for an arbitrary μ, the optimal dual solution μ^* will be a function μ.

Theorem 1: If $\mathbf{I}_1(\mu) = \mathbf{I}_1(\mu^*)$ and $\mathbf{I}_0(\mu) = \mathbf{I}_0(\mu^*)$ for an arbitrary dual solution μ, then the optimal dual solution $\mu^*$is given as

$$\mu^* = \mu \exp \left\{ \frac{\pi \lambda r_{\max}^2}{\tilde{n}(\mu)} \left(\sum_{i \in \mathbf{I}} p_i(\mu) - M \right) \right\} \tag{24}$$

where $\tilde{n}(\mu) = N - |\mathbf{I}_1(\mu) \cup \mathbf{I}_0(\mu)|$.

Proof: Let μ and $p_i(\mu)$ denote the arbitrary dual and primal solutions to (6), respectively. Similarly, let μ^* and $p_i(\mu^*)$ denote the optimal dual and primal solutions to (6), respectively. Define $\Delta p_i = p_i(\mu) - p_i(\mu^*)$, given by (18) as follows

$$\Delta p_i = \frac{1}{\pi \lambda r_{\max}^2} \log \frac{\xi_i(\mu^*)}{\xi_i(\mu)}. \tag{25}$$

Adding Δp_i for all i,

$$\sum_{i \in \mathbf{I}} \Delta p_i = \frac{1}{\pi \lambda r_{\max}^2} \sum_{i \in \mathbf{I}} \log \frac{\xi_i(\mu^*)}{\xi_i(\mu)}. \tag{26}$$

Letting $\boldsymbol{\Phi}(\mu) = \mathbf{I}_1(\mu) \cup \mathbf{I}_0(\mu)$, the summation term in (26) can be represented as

$$\begin{aligned} \sum_{i \in \mathbf{I}} \log \frac{\xi_i(\mu^*)}{\xi_i(\mu)} &= \sum_{i \in (\boldsymbol{\Phi}(\mu) \bigcup \boldsymbol{\Phi}(\mu^*))^c} \log \frac{\xi_i(\mu^*)}{\xi_i(\mu)} \\ &\quad + \sum_{i \in \boldsymbol{\Phi}(\mu) \bigcup \boldsymbol{\Phi}(\mu^*)} \log \frac{\xi_i(\mu^*)}{\xi_i(\mu)} \\ &= n(\mu^*, \mu) \log \frac{\mu^*}{\mu} + \omega(\mu^*, \mu) \end{aligned} \tag{27}$$

where

$$n(\mu^*, \mu) = N - |\boldsymbol{\Phi}(\mu) \cup \boldsymbol{\Phi}(\mu^*)| \tag{28}$$

and

$$\begin{aligned} \omega(\mu^*, \mu) = &\sum_{i \in \mathbf{I}_0(\mu) - \boldsymbol{\Phi}(\mu^*)} \log \frac{\mu^*}{\pi \lambda r_{\max}^2 g_i} \\ &+ \sum_{i \in \mathbf{I}_1(\mu) - \boldsymbol{\Phi}(\mu^*)} \log \frac{\mu^*}{\pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2}} \\ &+ \sum_{i \in \mathbf{I}_0(\mu^*) - \boldsymbol{\Phi}(\mu)} \log \frac{\pi \lambda r_{\max}^2 g_i}{\mu} \\ &+ \sum_{i \in \mathbf{I}_1(\mu^*) - \boldsymbol{\Phi}(\mu)} \log \frac{\pi \lambda r_{\max}^2 g_i e^{-\pi \lambda r_{\max}^2}}{\mu} \\ &+ \sum_{i \in \mathbf{I}_0(\mu^*) \bigcap \mathbf{I}_1(\mu)} \pi \lambda r_{\max}^2 \\ &- \sum_{i \in \mathbf{I}_1(\mu^*) \bigcap \mathbf{I}_0(\mu)} \pi \lambda r_{\max}^2. \end{aligned} \tag{29}$$

Furthermore, p_i^* in (6) can be optimal when $\sum_{i \in \mathbf{I}} p_i^* = M$, and thus,

$$\begin{aligned} \sum_{i \in \mathbf{I}} \Delta p_i &= \sum_{i \in \mathbf{I}} p_i - M \\ &= \frac{1}{\pi \lambda r_{\max}^2} \left\{ n(\mu^*, \mu) \log \frac{\mu^*}{\mu} + \omega(\mu^*, \mu) \right\}. \end{aligned} \tag{30}$$

Solving (30) for μ^*,

$$\mu^* = \mu \exp\left\{\frac{\pi\lambda r_{\max}^2\left(\sum_{i\in\mathbf{I}} p_i(\mu) - M\right) - \omega(\mu^*,\mu)}{n(\mu^*,\mu)}\right\}. \tag{31}$$

If $\mathbf{I}_1(\mu) = \mathbf{I}_1(\mu^*)$ and $\mathbf{I}_0(\mu) = \mathbf{I}_0(\mu^*)$, then $n(\mu^*,\mu) = N - |\mathbf{\Phi}(\mu)|$ and $\omega(\mu^*,\mu) = 0$, which reduces (31) to (24), i.e.,

$$\mu^* = \mu \exp\left\{\frac{\pi\lambda r_{\max}^2}{\tilde{n}(\mu)}\left(\sum_{i\in\mathbf{I}} p_i(\mu) - M\right)\right\}. \tag{32}$$

□

In the subsequent theorem, we provide a necessary and sufficient condition for the optimal dual solution, which will serve as a stopping condition for our search algorithm.

Theorem 2: When $\mathbf{I}_1(\mu^*) \neq \mathbf{I}$, $\sum_{i\in\mathbf{I}} p_i(\mu^*) = M$, i.e., an arbitrary value of is the optimal dual- solution μ^* iff $\sum_{i\in\mathbf{I}} p_i(\mu) = M$.

Proof: When $N = M$, $p_i^* = 1$, $\forall i \in \mathbf{I}$, i.e., $\mathbf{I}_1(\mu^*) = \mathbf{I}$, which turns out to be a trivial solution. Therefore, we just consider the case of $N \neq M$, i.e., $\mathbf{I}_1(\mu^*) \neq \mathbf{I}$. First, $\sum_{i\in\mathbf{I}} p_i(\mu^*) = M$ from (15). For arbitrary μ such that $\mu < \mu^*$, $\sum_{i\in\mathbf{I}} p_i(\mu) > M$ since $p_i(\mu) \geq p_i(\mu^*)$, $\forall i \in \mathbf{I}$, and $p_j(\mu) > p_j(\mu^*)$, $\exists j \in \mathbf{I} - \mathbf{I}_1(\mu^*)$. Furthermore, for arbitrary μ such that $\mu > \mu^*$, $\sum_{i\in\mathbf{I}} p_i(\mu) < M$ since $p_i(\mu) \leq p_i(\mu^*)$, $\forall i \in \mathbf{I}$, and $p_j(\mu) < p_j(\mu^*)$, $\exists j \in \mathbf{I} - \mathbf{I_1}(\mu^*)$. Therefore, for $\mathbf{I}_1(\mu^*) \neq \mathbf{I}$, μ^* is a unique value of μ such that $\sum_{i\in\mathbf{I}} p_i(\mu) = M$. In other words, $\sum_{i\in\mathbf{I}} p_i(\mu) = M$ is a necessary and sufficient condition to be $\mu = \mu^*$. □

Theorem 1 indicates that the optimal solution μ^* can be derived by searching for an arbitrary value of μ such that $\mathbf{I}_1(\mu) = \mathbf{I}_1(\mu^*)$ and $\mathbf{I}_0(\mu) = \mathbf{I}_0(\mu^*)$. In order to facilitate the search for μ, a continuous real number, we first investigate the properties of μ^*. In fact, the following theorem identifies the range of the optimal solution μ^*.

Theorem 3: The optimal dual solution μ^* exists within the following ranges:

$$\pi\lambda r_{\max}^2 g_M e^{-\pi\lambda r_{\max}^2} \leq \mu^* \leq \pi\lambda r_{\max}^2 g_M. \tag{33}$$

Proof: If $\pi\lambda r_{\max}^2 g_M e^{-\pi\lambda r_{\max}^2} > \mu^*$, then $|\mathbf{I}_1(\mu^*)| > M$ and $\sum_{i\in\mathbf{I}} p_i(\mu^*) > M$. Furthermore, if $\mu^* > \pi\lambda r_{\max}^2 g_M$, then $|\mathbf{I}_0(\mu^*)| > N - M$ and $\sum_{i\in\mathbf{I}} p_i(\mu^*) < M$. As $\sum_{i\in\mathbf{I}} p_i(\mu^*) = M$ by Theorem 2, the optimal dual solution must be in the range of (33). □

According to the definitions in (22) and (23), the elements in $\mathbf{I}_0(\mu)$ and $\mathbf{I}_1(\mu)$ vary by g_i, which determines the boundary value. Now, let us define a set of boundary values for the dual solution within the range of μ^* given by Theorem 3, which is given as

$$\begin{aligned}\boldsymbol{\mu} &= \{\mu_1, \mu_2, \cdots, \mu_N\} \\ &= \left\{\pi\lambda r_{\max}^2 g_m e^{-\pi\lambda r_{\max}^2} | m \in [1, M]\right\} \\ &\cup \left\{\pi\lambda r_{\max}^2 g_m | g_m > g_M e^{-\pi\lambda r_{\max}^2}, m \in [M, N]\right\}\end{aligned} \tag{34}$$

where we have assumed that $\mu_i \leq \mu_j$ for $i > j$ without loss of generality. Note that there are a maximum of N elements in (34). To describe the proposed search algorithm, we define the following dual-solution update function

$$f(x) = x \exp\left\{\frac{\pi\lambda r_{\max}^2}{N - |\mathbf{\Phi}(x)|}\left(\sum_{i\in\mathbf{I}} p_i(x) - M\right)\right\}. \tag{35}$$

Now, the optimal dual solution can be found by searching for μ satisfying $\sum_{i\in\mathbf{I}} p_i(f(\mu)) = M$ as required by Theorem 2. Therefore, based on Theorems 2 and 3, the following optimal dual-solution searching algorithm can be constructed.

Algorithm 1 Optimal Dual-Solution Searching (ODSA).

```
//Set the boundary values
μ ← {μ_1, μ_2, ···, μ_N}
  = {πλr_max^2 g_m e^{-πλr_max^2} | m ∈ [1, M]}
  ∪ {πλr_max^2 g_m | m ∈ [M, N]}
where μ_i ≤ μ_j for i > j;

// Initialization of starting point, step size, and dual solution
n ← ⌈N/2⌉; t ← ⌈N/2⌉; μ ← μ_n;
while Σ_{i∈I} p_i(f(μ)) ≠ M do
    // Repeat until Σ_{i∈I} p_i(f(μ)) = M
    t ← ⌈t/2⌉;
    if Σ_{i∈I} p_i(μ) − M > 0 then
        n ← max(0, n − t); //Set the next searching point
    else if Σ_{i∈I} p_i(μ) − M < 0 then
        n ← min(N, n + t); //Set the next searching point
    end if;
    μ ← μ_n;
end while
μ* ← f(μ);
p* ← p(μ*);
```

Fig. 2 illustrates the searching process for Algorithm 1 when $|\boldsymbol{\mu}| = 14$. At the first step, n is set to $\lceil|\boldsymbol{\mu}|/2\rceil = 7$. Since $\sum_{i\in\mathbf{I}} p_i(f(\mu_7)) \neq M$ and $\sum_{i\in\mathbf{I}} p_i(\mu_7) > M$, n is reduced by one-half, i.e., $n = 7 - \lceil 7/2\rceil = 3$, for the second step. At the second and third steps, n is updated as $n = 3 + \lceil 4/2\rceil = 5$ and $n = 5 - \lceil 2/2\rceil = 4$, respectively. Since $\sum_{i\in\mathbf{I}} p_i(f(\mu_4)) = M$ at the fourth step, searching is stopped, and the optimum dual solution is given by $f(\mu_4)$. As each step reduces a search space range by one half, starting from the maximum range of N, Algorithm 1 involves a complexity of $O(\log_2 N)$ iterations to search for the optimal solution.

IV. NUMERICAL RESULTS

In this section, we first compare the performance of the proposed ODSA and the subgradient method in the Appendix from the viewpoints of computational complexity and accuracy for the given operational environment. Then, we compare the performance of the average caching failure rate obtained by using the different caching probabilities, including the optimal one found by ODSA.

In the current numerical analysis, we assume a maximum transmission radius of 100 m, i.e., $r_{\max} = 100$ m, and a PPP distribution of devices with average λ (devices/m^2). For $r_{\max} = 100$ m, an average of $10,000\lambda\pi$ devices are uniformly

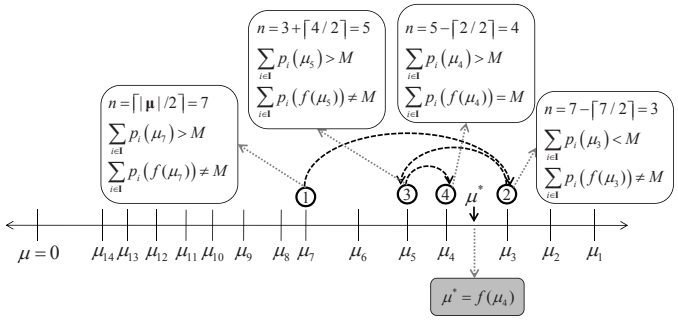


Fig. 2. Illustrative example of Algorithm 1: $|\mu| = \mathbf{14}$.

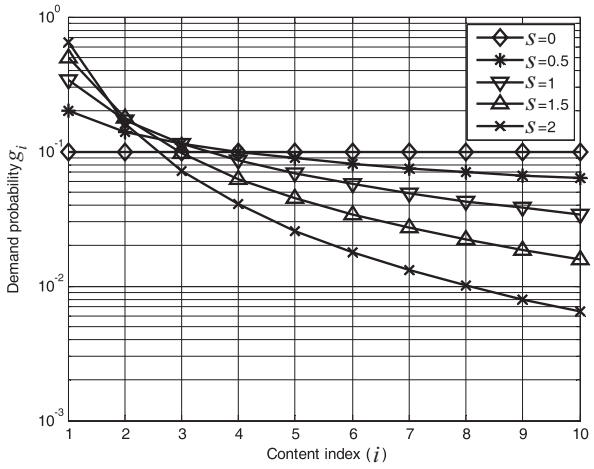


Fig. 3. Zipf distribution with varying demand dominance factor s: illustration.

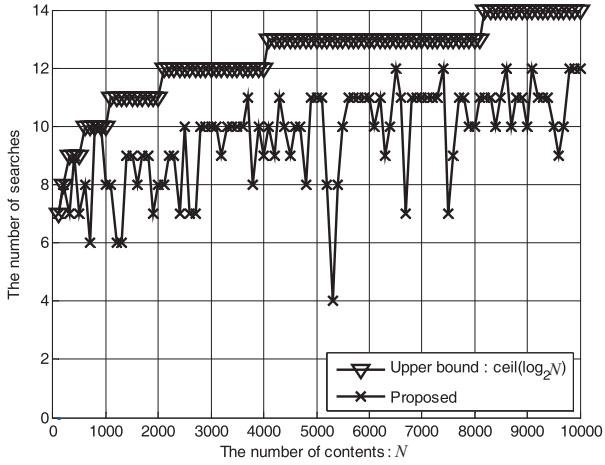
distributed within a disc area of radius 100 m, centered around an arbitrary receiver. We assume that each of the *N* representative contents has its own demand probability. For example, the content demand probability for content *i* can be modeled by the following Zipf distribution [8], [9]

$$g_i(s,N) = \frac{(1/i)^s}{\sum_{k=1}^{N}(1/k)^s} \tag{36}$$

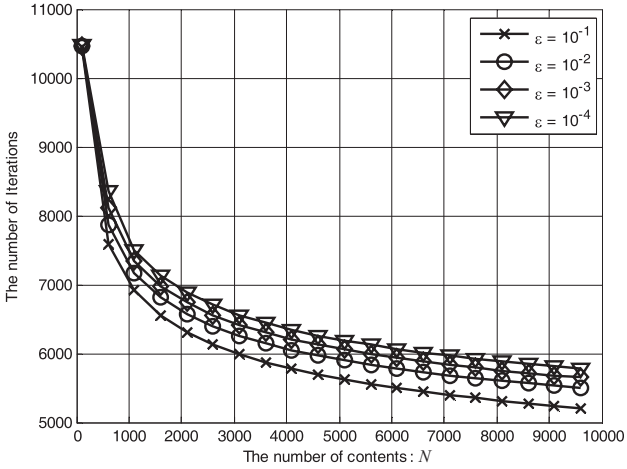
where *s* is a demand dominance factor. The demand dominance factor is a parameter that characterizes the distribution, i.e., the larger the dominance factor is, the more demanded the popular contents are. Fig. 3 illustrates the distribution in (36) with varying s. As shown in Fig. 3, more popular contents are more frequently demanded as the demand dominance factor increases, while $s = 0$ corresponds to a uniform distribution.

A. Complexity and Accuracy of Solution Approaches: Comparison

The convergence rate and accuracy with ODSA and the subgradient method are discussed in the Appendix, depending on the number of contents (N), the number of contents to be cached (M), the intensity of the caching server (λ), and the demand dominance factor (s) in (36). As ODSA guarantees the optimal solution, it will be a basis for evaluating the accuracy of the subgradient method, along with its relative complexity. For the subgradient method, an initial value of μ is set to



(a)



(b)

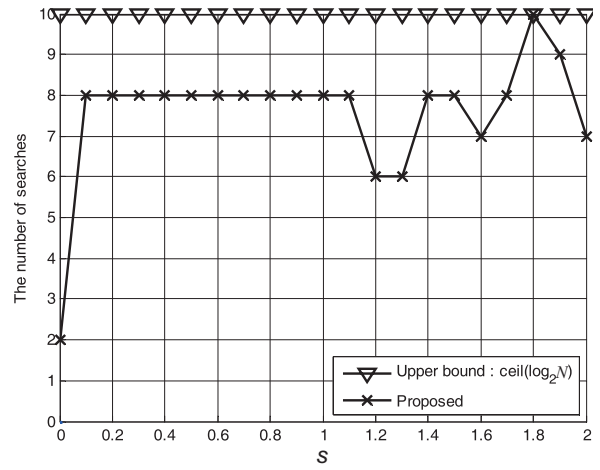
Fig. 4. Computational complexity as varying the number of contents, N : $M = 10$, $s = 1$, $\lambda = 0.01/\pi$, $r_{\max} = 100$ m, and $\Delta_\mu = 10^{-5}$: (a) ODSA and (b) subgradient method.

$\pi\lambda r_{\max}^2 g_M e^{-\pi\lambda r_{\max}^2}$. Let $\mathbf{p}_{A1}^*$ denote the optimal solution obtained by ODSA. Let $k^*(\varepsilon)$ be the minimum number of iterations required for the iterative solution to obtain its performance close to $\bar{f}(\mathbf{p}_{A1}^*)$ in (4), within an error range of ε, i.e.,

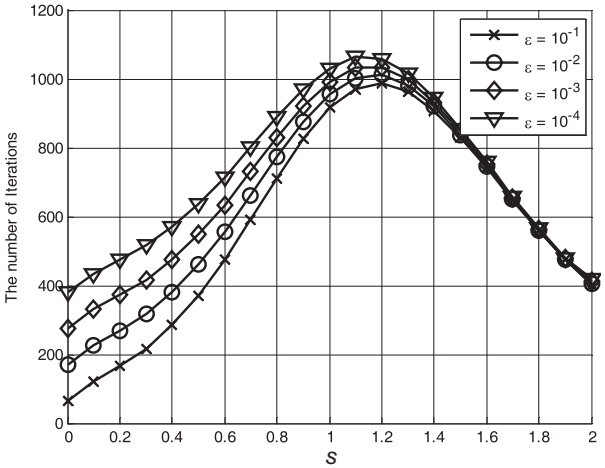
$$k^*(\varepsilon) = \min\left\{ k \,\middle|\, \frac{\left|\bar{f}\left(\mathbf{p}_{SG}^{(k)}\right) - \bar{f}\left(\mathbf{p}_{A1}^*\right)\right|}{\bar{f}\left(\mathbf{p}_{A1}^*\right)} < \varepsilon \right\} \tag{37}$$

where $\mathbf{p}_{SG}^{(k)}$ is the solution using the subgradient method with k iterations. The number of iterations in (37), $k^*(\varepsilon)$, indicates how much faster than the subgradient method ODSA converges to the optimal solution.

Given device density λ and caching capacity M, the demand dominance factor s and the number of contents N vary in practice. Therefore, let us investigate the number of iterations as N, s, and M vary. Fig. 4 shows the size of the search space for ODSA and the number of iterations, $k^*(\varepsilon)$, in the subgradient method varying N with $M = 1$, $s = 1$, and $\lambda = 0.01/\pi$. As shown in Fig. 4(a), the number of iterations for ODSA is bounded by $\lceil \log_2 N \rceil$, e.g., 14 for $N = 10,000$, while achiev-

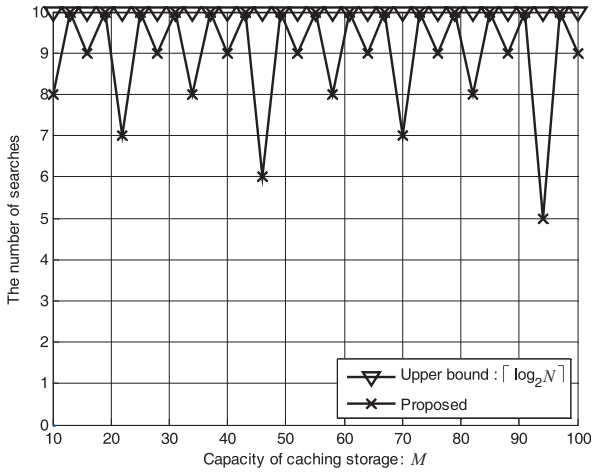


(a)

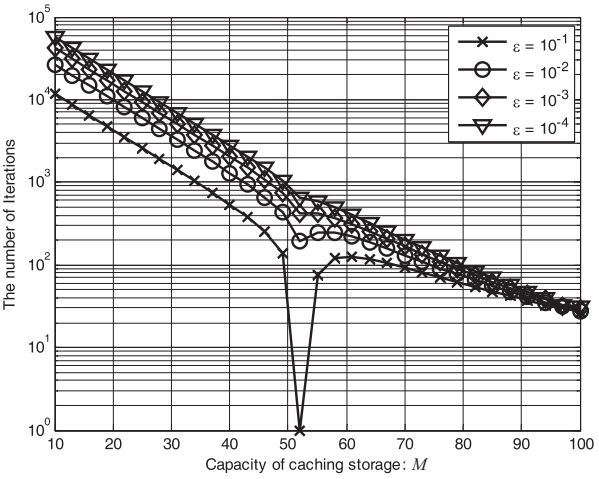


(b)

Fig. 5. Computational complexity as varying the demand dominance factor, s : $N = 1,000$, $M = 10$, $\lambda = 0.001/\pi$, $r_{\max} = 100$ m, and $\Delta_\mu = 8 \times 10^5$: (a) ODSA and (b) subgradient method.



(a)



(b)

Fig. 6. Computational complexity as varying the number of contents that can be store in the device, M :$N = 1,000$, $s = 1$,$\lambda = 0.001/\pi$, $r_{\max} = 100$ m, and $\Delta_\mu = 2 \times 10^7$: (a) ODSA and (b) subgradient method.

ing the actual number of iterations below the bound by breaking the rule in Theorem 2. As shown in Fig. 4(b), meanwhile, the subgradient method requires 5,000 to 10,000 iterations with an update step size $\Delta_\mu = 10^{-5}$, while requiring more iterations as ε is reduced. Note that $\Delta_\mu = 10^{-5}$ in this result is the best value found by trial and error. However, it must be reconfigured as M, s, and λ change. This implies that the actual number of iterations required in practice would be greater than in Fig. 4(b).

Fig. 5 also shows the search space size for ODSA and the number of iterations in the subgradient method with varying s for $N = 1,000$, $M = 10$, and $\lambda = 0.001/\pi$. In Fig. 5(a), we find that ODSA requires an average of eight searches with an upper bound of ten searches. We note that the upper bound on the number of searches does not change with s, since N is fixed. Meanwhile, the subgradient method takes roughly 100 to 1,000 iterations, depending on ε and s, with $\Delta_\mu = 8 \times 10^5$, as shown in Fig. 5(b). The convergence rate of the subgradient method depends on Δ_μ and the initial value of μ. Furthermore, their optimal values depend on s. This explains why the number of iterations required to meet the target ε varies with s. As general forms for the optimal values of Δ_μ and initial values of μ are not known for the various problems, it is impossible to set the appropriate values for each problem in practice.

On the other hand, Fig. 6 shows the same results for varying M with $N = 1,000$, $s = 1$, and $\lambda = 0.001/\pi$. As in Fig. 5(a), Fig. 6(a) shows that ODSA finds the optimal solution with the number of searches bounded by $\lceil \log_2 N \rceil$. Meanwhile, Fig. 6(b) shows that the subgradient method with $\Delta_\mu = 2 \times 10^7$ involves a large number of iterations, e.g., more than 10,000 iterations for a rather small M. It can be observed in Fig. 6(b) that only one iteration is required for the subgradient method to meet $\varepsilon = 10^{-1}$ when $M = 51$. We conjecture that $\Delta_\mu = 2 \times 10^7$ and the initial value of μ happens to be the best choice at $M = 51$ with a rather loose requirement for accuracy, e.g., $\varepsilon = 10^{-1}$. We note that a large number of iterations are still required for $\varepsilon < 10^{-1}$ even when $M = 51$. As it is not possible to set appropriate values to individual problems with different parameters, we may have to resort to some fixed values of $\Delta_\mu = 2 \times 10^7$ and an arbitrary initial value of μ. In practice, then, a sufficiently large number of iterations will be required for the subgradient method to con-

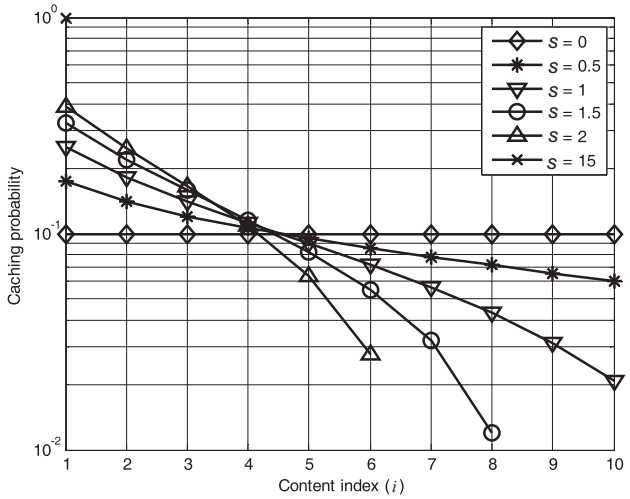


Fig. 7. Caching probability distribution as varying the demand dominance factor, s : $N = 10$, $M = 1$, $\lambda = 0.001/\pi$, and $r_{\max} = 100$ m.

verge within the given accuracy.

Taking all the results from Figs. 4–6 into account, we conclude that the proposed ODSA can significantly reduce complexity as compared to the subgradient method, while guaranteeing a finite number of searches even when the various system parameters are changed.

B. Performance Analysis of Caching Policies: Comparison

The performance of the proposed policy with the optimal caching probability is compared with that of other caching policies: one with an EP caching policy and the other with a HPF caching policy. The EP policy is to cache all contents with equal probability, e.g., with the following probability

$$p_i = M/N, \forall i \in \mathbf{I}. \tag{38}$$

The HPF policy is to cache the M most frequently requested contents, corresponding to the following caching probabilities

$$p_i = \begin{cases} 1 \text{ if } i \leq M; \\ 0 \text{ if } i > M. \end{cases} \tag{39}$$

In the numerical results in this subsection, we assume that $\lambda = 0.001/\pi$.

Fig. 7 presents the optimum caching probability of individual content for for $N = 10$, $M = 1$, $\lambda = 0.001/\pi$, and $r_{\max} = 100$ m, that is computed by the proposed algorithm, as varying the demand dominance factor s. Note that the content index without a value of caching probability in Fig. 7 corresponds to $p_i = 0$. The optimum caching probabilities turn out be slightly different from the demand probabilities by Zipf distribution. It indicates that the most popular content does not need to be stored by all CSD's. Furthermore, it is found from Fig. 7 that some contents with very low demand do not need to be stored in any CSD, i.e., $p_i = 0$, due to the limited storage capacity, especially when s is large, e.g., $s > 1.5$. Furthermore, it show that only the most popular content must be stored by all CSD's, i.e., $p_1 = 1$ and $p_2 = p_3 = \cdots = p_{10} = 0$, when $s = 15$.

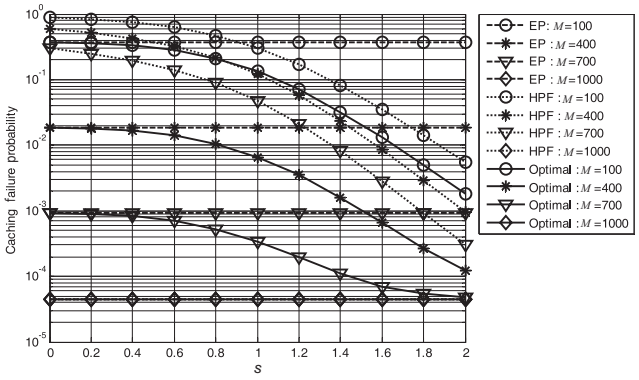


Fig. 8. The average caching failure probability as varying the demand dominance factor s: $N = 1,000$, $r_{\max} = 100$ m, and $\lambda = 0.001/\pi$.

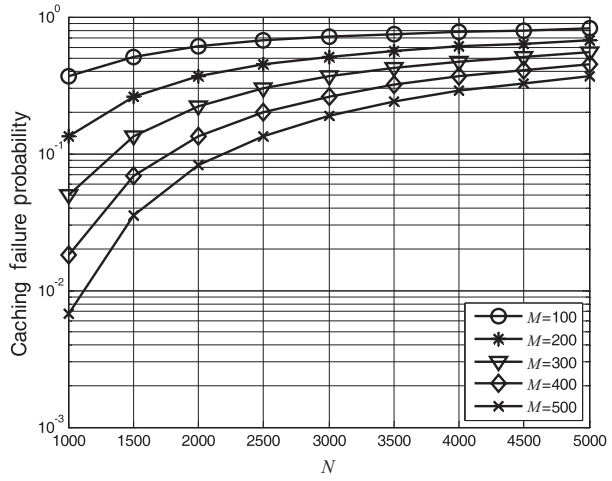
Fig. 8 shows the average caching failure performance for the EP, HPF, and optimal policies with different numbers of caching storage capacity M as a dominance factor s varies when $N = 1,000$. First, the caching failure probability decreases as the demand dominance factor s increases. This is attributed to the fact that only a limited number of contents will be requested with a large demand dominance factor, which tends to store some contents with a high probability, reducing the caching failure rate. In fact, the performance of the HPF policy is close to the optimal performance for a large demand dominance factor. It is the other way around for the EP policy. Meanwhile, when $M = 1,000$, all schemes have the same caching failure probability (showing only a single line in Fig. 8). It is attributed to the fact that all contents can be stored with $M = 1,000$ for $N = 1,000$ contents, i.e., with the caching probability of 1 for each content.

Fig. 9 shows the average caching failure performance of EP, HPF, and optimal policies for different numbers of caching storage capacity M as the number of contents N varies with $s = 1$. Due to limited storage capacity, the average caching failure probability increases with N. We observe that the EP and HPF policies perform better for small and large numbers of contents, respectively. Furthermore, the crossing point for the performance of the EP and HPF policies moves toward the larger N as M increases. However, the average caching failure performance for the EP or HPF scheme deviates from the optimum when M is sufficiently larger than N.

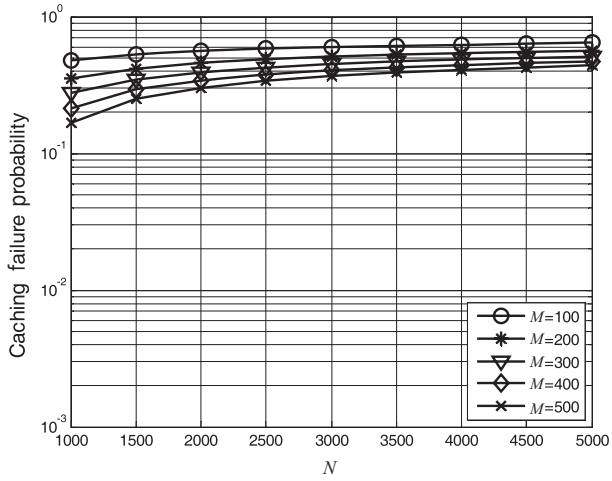
Taking the results from Figs. 8 and 9 into account, we note that the optimal performance can be achieved by employing the EP or HPF policy selectively, depending on N and s. As N tends to be much larger than M, and most traffic will be governed by some popular contents in general, a further study on such a selective scheme may be meaningful.

V. CONCLUSION

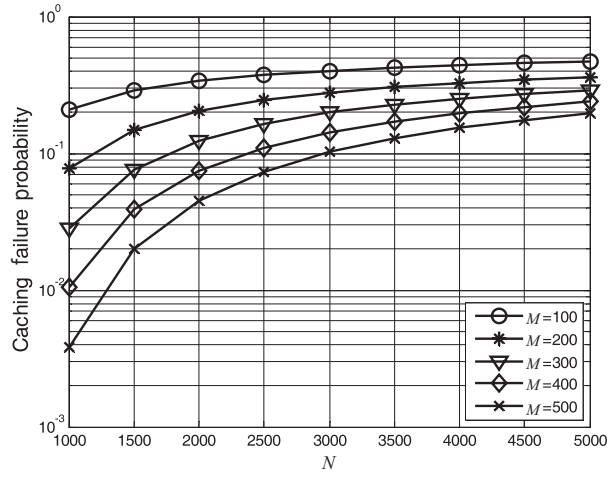
In this paper, we considered a design and optimization framework for an mCDN in which, by direct D2D communication, mobile devices in proximity can share the contents that are cached in the individual devices, i.e., a mobile device consumes the contents while storing them as a CSD. As the connectivity of mobile D2D links are highly dynamic and, furthermore,



(a)



(b)



(c)

Fig. 9. The average caching failure probability as varying the number of caching-service devices, N: $s = 1$, $r_{\max} = 100$ m, and $\lambda = 0.001/\pi$: (a) EP, (b) HPF, and (c) Optimum.

since storage size is strictly limited in the mCDN, we attempted to determine which contents must be stored in CSD's, given the popularity of individual contents in terms of their demand probabilities. In this paper, we presented a low-complexity search algorithm to solve an optimization problem that minimizes the average caching failure probability. Based on our optimization framework, we found that less popular contents must still be cached with some given probabilities while caching more popular contents with a higher probability. On the other hand, it was found that when the demand statistics are not known a priori, performance could be improved significantly by alternately employing the policy of caching all contents with the same probability and that of caching some of the highly popular contents only.

As the small-cell approach becomes an essential means of coping with mobile traffic explosion, the practicality of mobile caching devices becomes more acceptable with an mCDN. In fact, the ultimate form of the small cell would be a portable base station that can be carried by an individual user with a wireless backhaul, eventually allowing for more base stations than mobile devices in some situations. The portable base stations are dynamically inter-connected to form reconfigurable backhaul links. As the popular contents can be cached in portable base stations, the reconfigurable backhaul infrastructure will serve as an mCDN, reducing wireless data transmissions, especially without communication with the server in the core network. Then, caching policies become essential for dealing with the mobility and limited storage of portable base stations. Therefore, our proposed optimization framework and the solution approach therein can be useful for implementing mCDN with portable base stations, which would be the ultimate form of small cells in the next generation mobile information system.

ACKNOWLEDGMENTS

We also would like to express our appreciation to Mr. Kumin Cho and Mr. Kwon Y. Park for their valuable and constructive suggestions during the the planning and development of this research work.

APPENDIX

A. Subgradient-based Iterative Method

The optimization problem (6)–(9) can be solved using the projected subgradient method for the dual solution [4], [5]. It first defines the following Lagrangian function

$$L(\mathbf{p}, \mu, \boldsymbol{\gamma}, \boldsymbol{\sigma}) = \sum_{i=1}^{N} g_i e^{-\pi\lambda r_{\max}^2 p_i} - \mu\left(M - \sum_{i=1}^{N} p_i\right) - \sum_{i=1}^{N} \gamma_i (1 - p_i) + \sum_{i=1}^{N} \sigma_i p_i. \tag{V-A.40}$$

Letting $g(\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma}) = \inf_{\mathbf{p}} L(\boldsymbol{p}, \mu, \boldsymbol{\gamma}, \boldsymbol{\sigma})$, a dual problem of (6)-(9) can be stated as

$$\max_{\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma}} g(\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma}) \tag{V-A.41}$$

subject to

$$\mu \geq 0, \tag{V-A.42}$$

$$\gamma_i \geq 0,\ \sigma_i \geq 0,\ \forall i = 1, \cdots, N. \tag{V-A.43}$$

As $L(\mathbf{p}, \mu, \boldsymbol{\gamma}, \boldsymbol{\sigma})$ in (V-A.40) is jointly convex with respect to $\mathbf{p}$, $\nabla L(\mathbf{p}^*(\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma}), \mu, \boldsymbol{\gamma}, \boldsymbol{\sigma}) = 0$. Therefore, $\hat{\mathbf{p}}(\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma})$ minimizing $L(\mathbf{p}, \mu, \boldsymbol{\gamma}, \boldsymbol{\sigma})$is found by the following equation

$$\left.\frac{\partial L(\mathbf{p}, \mu, \boldsymbol{\gamma}, \boldsymbol{\sigma})}{\partial p_i}\right|_{\mathbf{p}=\hat{\mathbf{p}}} = -\pi\lambda r_{\max}^2 e^{-\pi\lambda r_{\max}^2 \hat{p}_i} g_i + \mu + \gamma_i - \sigma_i = 0. \tag{V-A.44}$$

Solving (V-A.44) for $\hat{\mathbf{p}}(\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma})$,

$$\hat{p}_i(\mu, \boldsymbol{\gamma}, \boldsymbol{\sigma}) = \frac{1}{\pi\lambda r_{\max}^2} \log\left\{\frac{\pi r_{\max}^2 \lambda g_i}{\mu + \gamma_i - \sigma_i}\right\}. \tag{V-A.45}$$

Applying the projected subgradient method to the constrained optimization problem defined in [4] and [5], the optimal $(\mu^*, \boldsymbol{\gamma}^*, \boldsymbol{\sigma}^*)$ can be determined through the following iterative steps.

$$\mu(\tau+1) = \left[\mu(\tau) + \Delta_\mu \left\{M - \sum_{i\in\mathbf{I}} \hat{p}_i\left(\mu(\tau), \boldsymbol{\gamma}(\tau), \boldsymbol{\sigma}(\tau)\right)\right\}\right]^+, \tag{V-A.46}$$

$$\gamma_i(\tau+1) = \left[\gamma_i(\tau) + \Delta_\gamma\left(1 - \hat{p}_i(\mu(\tau), \boldsymbol{\gamma}(\tau), \boldsymbol{\sigma}(\tau))\right)\right]^+, \tag{V-A.47}$$

$$\sigma_i(\tau+1) = \left[\sigma_i(\tau) + \Delta_\sigma \hat{p}_i(\mu(\tau), \boldsymbol{\gamma}(\tau), \boldsymbol{\sigma}(\tau))\right]^+ \tag{V-A.48}$$

where $[x]^+ = \max(x, 0)$, Δ_μ, Δ_γ, and Δ_σ are positive real values as the step sizes for μ, γ, and σ, respectively. Assuming strong duality, the optimal caching probability p_i^* can be derived by substituting $(\mu^*, \boldsymbol{\gamma}^*, \boldsymbol{\sigma}^*)$ into (V-A.45). In order to satisfy the constraints (7)–(9), the elements of $\mathbf{p}^*$ are normalized as follows

$$p_i^* = \min\left(\frac{M[\hat{p}_i(\mu^*, \boldsymbol{\gamma}^*, \boldsymbol{\sigma}^*)]^+}{\sum_{i\in\mathbf{I}}[\hat{p}_i(\mu^*, \boldsymbol{\gamma}^*, \boldsymbol{\sigma}^*)]^+}, 1\right). \tag{V-A.49}$$

REFERENCES

[1] N. Golrezaei *et al.*, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, 2013.

[2] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Wireless device-to-device communications with distributed caching," *arXiv:1205.7044v1 [cs.IT]*, May 2012.

[3] F. Baccelli, B. Blaszczyszyn, and P. Muhlethaler, "An Aloha Protocol for Multihop Mobile Wireless Networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 421–436, Feb. 2006.

[4] S. Boyd and A. Mutapcic, Subgradient Methods, Notes for EE364b, Stanford University, Apr. 2008.

[5] S. Low and D. E. Lapsley, "Optimization flow control, I: Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[6] K. Doppler *et al.*, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12. pp. 42–94, Dec. 2009.

[7] WirelessMAN-advanced air interface for broadband access systems-draft amendment: Higher reliability networks, IEEE P802.16.1a/D2, Apr. 2012.

[8] H. Yu *et al.*, "Understanding User Behavior in Large-Scale Video-on-Demand Systems," in *Proc. ACM EuroSys*, Apr. 2006.

[9] M. Cha *et al.*, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. ACM SIGCOMM*, 2007.

Hye Joong Kang received the B.S. and M.S. degrees in Electrical Engineering from Korea University, Seoul in 2007 and 2009, respectively. Since then, he has been with the Department of IT Convergence in Korea University as a Ph.D student and a Researcher. He is a recipient of full scholarship from Samsung Electronics, Inc. for his advanced program. His research interests include radio resource management and cross-layer design for mobile radio communication system and wireless networks.

Chung Gu Kang received the B.S. degree in Electrical Engineering from the University of California, San Diego in 1987 and the M.S. and Ph. D. degrees both in Electrical and Computer Engineering from the University of California, Irvine in 1989 and 1993, respectively. While working on his Ph.D. dissertation from June 1991 to May 1992, he was also with the Aerospace Corporation in El Segundo, California as a Part-Time Member of Technical Staff (MTS). In 1993, he joined Rockwell International Inc. in Anaheim, California, where he worked on the signaling system no. 7 and other telecommunication systems development. Since March 1994, he has been with the Department of Radio Communication & Engineering and later, with the Department of Electrical Engineering at the Korea University, Seoul, Republic of Korea, where he is currently a Full Professor. In the academic year 2000–2001, he was a Visiting Associate Professor at the University of California, San Diego, where he was jointly affiliated with the Center for Wireless Communications. His research interests include next generation mobile radio communication system and wireless networks, with special emphasis on physical layer/medium access control layer design and performance analysis. He developed the various types of system-level simulators and service engineering tools (e.g., MAXIM, ROMEOS, SimP2SON) for 3G, 4G, and Wireless LAN, which have been actively employed for service & network planning in the field. His recent research is focused on the cross layer design issues for MIMO/multiple access schemes for mobile broadband wireless access systems and MAC/routing protocols for mobile ad hoc networks. He has over 200 refereed publications in international journals and conference proceedings in the areas of communications network, CDMA cellular systems, OFDM systems, and wireless local area/personal area networks. He recently coauthored a reference textbook on the MIMO-OFDM wireless system, entitled "MIMO-OFDM Wireless Communication with MATLAB" (Wiley, 2010). He has been a Consultant to wireless industries, including Samsung Electronics and SK Telecom. He is currently serving as an Editor of JCN (Journal of Communication and Network). He is also a Chair of 2.3GHz IMT-WiBro Project Group (PG702) in Telecommunications Technology Association (TTA), which is a standard development organization (SDO) of mobile communication in Korea. He is a Member of IEEE COMSOC, IT, and VT, and KICS, having served as a Chair of KICS Mobile Communication Technical Activity Group.