Mobile Caching Policies for Device-to-Device (D2D) Content Delivery Networking

Hye J. Kang, Kown Y. Park, Kumin Cho and Chung G. Kang
School of Electrical Engineering
Korea University
Seoul, Korea
{dreamftr, tpower84, kumin84, ccgkang}@korea.ac.kr

***Abstract* — We consider a mobile content delivery network (mCDN) in which a special form of mobile devices designated as caching servers (caching-server device: CSD) can provide the near-by devices with some popular contents on demand via device-to-device (D2D) communication links. On the assumption that mobile CSDs 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 identify the important characteristics of the optimal caching policies in the mobile environment that would serve as a useful aid in designing an mCDN.**

***Index Terms*— Mobile Contents Distribution Network (CDN), Device-to-device Communication, Caching Probability, Caching Server Device, Poisson Point Process**

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, it tends that a small portion of popular contents, especially associated with the common interests, incurs the enormous amount of traffic that would take up a major portion of the overall traffic, as social network service (SNS) allows them to be shared dynamically and steadily among the public. In this situation, 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 distribution 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, offloading the network and server.

On the top of the global aspects, meanwhile, many issues over SNS are bounded by the local characteristics. Since one's interests in the various issues of politics, economics, society and culture are centered on his or her daily social interactions, those who are living in the proximity tend to consume the same contents more often than the others. Based on this nature, a concept of CDN recently 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]. And the main advantage of D2D communication in a cellular system is spatial reuse gained by enabling multiple direct links between two near-by devices at the same time. However, in the current commercial system, D2D communication links would be rarely activated owing to the fact that most traffic is originated mainly by the current 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, 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 CSDs with content-caching capability differ from conventional caching servers in the network in many ways. First, their service coverage is limited owing to a wireless link. Second, caching servers are subject to mobility, implying that the availability of contents would be spatially random. Third, caching capability is limited by the physical nature of mobile devices, e.g., limited storage capacity. As a result of these characteristics, content-caching policies must differ from the conventional ones. Content on demand can be provided only when a mobile device exists within the coverage of the corresponding caching server. In practice, the advantage of mCDN may be limited when a significant number of mobile devices in a broad range must be served with a finite caching storage on the move. Therefore, caching policies become essential for dealing with the

This research was funded by the MSIP (Ministry of Science, ICT & Future Pl anning), Korea in the ICT R&D Program 2013.

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

mobility and limited storage of mobile CSDs. In other words, the probability of caching individual content, e.g., if it must be cached or not, must be determined by the content demand rate, the coverage of the caching server device, and the distribution of mobile devices. For example, only the most popular contents can be cached within the storage capacity of a caching server, or some of the contents must be selectively cached, possibly with a given caching probability. In the event that a specific content cannot be found upon request, a server is subject to caching failure.

In this paper, we formulate an optimization problem to determine the optimum caching probability for each content. Then, we propose a low-complexity solution approach, the optimum dual-solution searching algorithm (ODSA), which minimizes the average caching failure probability. 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, as compared to 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 present ODSA. In Section IV, numerical results are presented to provide the performance of the different caching policies. Finally, conclusions are drawn in Section

II. System Model & Problem Formulation

A. System model

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

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

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 on average) that are obtained upon its own request or overhearing what other devices have transmitted and providing these contents to any neighbor device via a direct link upon the request. In fact, D2D communication links are used to share contents among the CSDs 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 have transmitted it. Therefore, those CSDs with content i are distributed by a PPP with average intensity $\lambda \cdot p_i$. Let D be a random variable representing the distance between a reference receiver that is requesting content i and the closest CSD with content i. In other words, as shown in Figure 1, this corresponds to the situation in which there exists no CSD with content i within a circular range of radius D centered at the reference receiver. Then, the cumulative distribution function (CDF) of D, $F_D^{(i)}(d)$, is given as

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

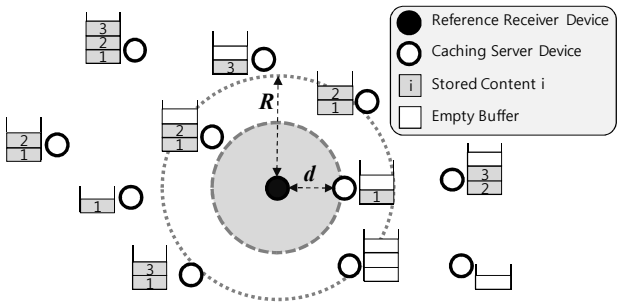

Figure 1. Distribution of caching-server devices (CSDs): Illustration

Content caching failure is defined as an event in which the reference receiver that requests a specific content cannot find the corresponding CSD within a distance R that is the maximum transmission range of the CSD. The caching failure rate of each content i for a reference receiver with a radius R, denoted as $P_f^{(i)}(R)$, is given by the probability of caching failure as follows:

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

Meanwhile, let g_i denote the probability that content i is requested by a device. We assume that a content with a smaller index has a larger probability of being requested by a device, i.e., $g_i \geq g_j$ if $i < j$. Given content caching probabilities $\mathbf{p} = \{p_1, p_2, ..., p_N\}$, 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 of requesting the corresponding content, as follows:

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

B. Problem formulation

We intend to determine the caching probability for each content, $\mathbf{p} = \{p_1, p_2, ..., p_N\}$, 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.,

$$\sum_{i\in\mathbf{I}} p_i \le M \tag{5}$$

In other words, M corresponds to the maximum number of contents that can be cached in a CSD on average. Now, our optimization problem can be formulated as

$$\min_{\mathbf{p}=\{p_1,p_2,\cdots,p_N\}} \sum_{i\in\mathbf{I}} g_i e^{-\pi\lambda R^2 p_i} \tag{6}$$

subject to
$$\sum_{i\in\mathbf{I}} p_i - M \le 0 \tag{7}$$

$$p_i - 1 \le 0,\ i=1,2,\cdots,N \tag{8}$$

$$-p_i \le 0,\ i=1,2,\cdots,N. \tag{9}$$

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.

III. Optimum Dual-solution Searching Algorithm

Note that the solution to our optimization problem (6)-(9) is a vector $\mathbf{p}=\{p_1,p_2,....,p_N\}$ with real elements. This type of problem can be solved using iterative schemes such as the subgradient method [4]. 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:

$$\frac{\partial L}{\partial p_i} = -\pi\lambda R^2 e^{-\pi\lambda R^2 p_i} g_i + \mu + \gamma_i - \sigma_i = 0 \tag{10}$$

where

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

- Primal conditions:

$$\sum_{i\in\mathbf{I}} p_i - M \le 0 \tag{11}$$

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

- 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,

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

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, γ_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

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

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

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

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

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

Proof: Left out due to lack of space. ■

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., $\mathbf{I}_0 = \{i \mid p_i = 0,\ i\in\mathbf{I}\}$. Similarly, define $\mathbf{I}_1$ to denote a set of contents that is always stored in all devices, i.e., $\mathbf{I}_1 = \{i \mid p_i = 1,\ i\in\mathbf{I}\}$. 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 $\mu < \pi\lambda R^2 g_i$ for $p_i > 0$ and $\mu > \pi\lambda R^2 g_i e^{-\pi\lambda R^2}$ for $p_i < 1$:

$$\mathbf{I}_0(\mu) = \{i \mid \pi\lambda R^2 g_i \le \mu, i \in [1, N]\} \quad (22)$$

$$\mathbf{I}_1(\mu) = \{i \mid \pi\lambda R^2 g_i e^{-\pi\lambda R^2} \ge \mu, \ \ i \in [1, N]\} \quad (23)$$

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* μ^* *is given as*

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

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

Proof: Left out due to lack of space. ■

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* μ *is the optimal dual solution* μ^* *iff* $\sum_{i\in\mathbf{I}} p_i(\mu) = M$.

Proof: Left out due to lack of space. ■

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:*

$$\pi\lambda R^2 g_M e^{-\pi\lambda R^2} \le \mu^* \le \pi\lambda R^2 g_M \quad (25)$$

Proof: Left out due to lack of space. ■

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

$$\begin{aligned}\boldsymbol{\mu} &= \{\mu_1, \mu_2, \cdots\cdots, \mu_N\} \\ &= \left\{\pi\lambda R^2 g_m e^{-\pi\lambda R^2} \mid m \in [1, M]\right\} \\ &\quad \bigcup \left\{\pi\lambda R^2 g_m \mid g_m > g_M e^{-\pi\lambda R^2}, m \in [M, N]\right\}\end{aligned} \quad (26)$$

where we have assumed that $\mu_i \le \mu_j$ for $i > j$ without loss of generality. Note that there are a maximum of N elements in (26). To describe the proposed search algorithm, we define the following dual-solution update function:

$$f(x) = x \cdot \exp\left\{\frac{\pi\lambda R^2}{N - |\mathbf{\Phi}(x)|}\left(\sum_{i\in\mathbf{I}} p_i(x) - M\right)\right\}. \quad (27)$$

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

$\boldsymbol{\mu} \leftarrow \{\mu_1, \mu_2, \cdots\cdots, \mu_N\}$

$= \left\{\pi\lambda R^2 g_m e^{-\pi\lambda R^2} \mid m \in [1, M]\right\} \bigcup \left\{\pi\lambda R^2 g_m \mid m \in [M, N]\right\}$

where $\mu_i \le \mu_j$ for $i > j$;

// initialization of starting point, step size, and dual solution

$n \leftarrow \lceil N/2 \rceil$; $t \leftarrow \lceil N/2 \rceil$; $\mu \leftarrow \mu_n$;

While $\sum_{i\in\mathbf{I}} p_i(f(\mu)) \neq M$ *// repeat until* $\sum_{i\in\mathbf{I}} p_i(f(\mu)) = M$

  $t \leftarrow \lceil t/2 \rceil$;

  If $\sum_{i\in\mathbf{I}} p_i(\mu) - M > 0$ *// set the next searching point*

    $n \leftarrow \max(0, n - t)$;

    else if $\sum_{i\in\mathbf{I}} p_i(\mu) - M < 0$

    $n \leftarrow \min(N, n + t)$; *// set the next searching point*

  end if;

  $\mu \leftarrow \mu_n$;

end while;

$\mu^* \leftarrow f(\mu)$;

$\mathbf{p}^* \leftarrow \mathbf{p}(\mu^*)$;

Figure. **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.

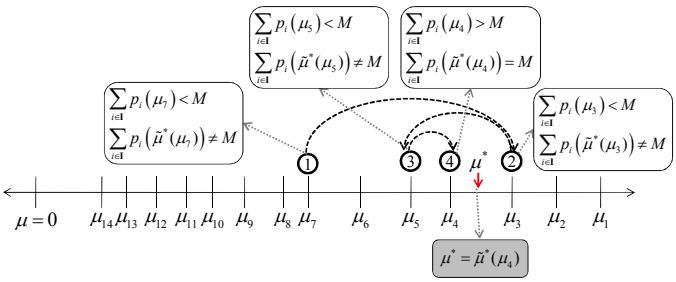

Figure 2. Illustrative example of Algorithm 1: $|\boldsymbol{\mu}| = 14$

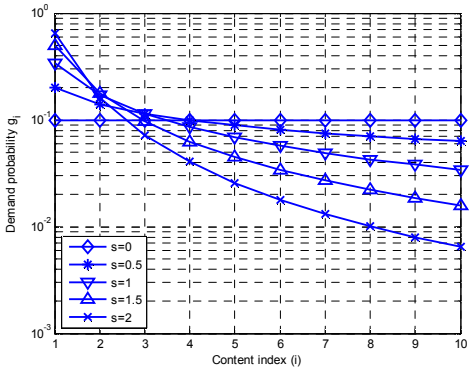

Figure 3. Zipf distribution with varying demand dominance factor s: illustration

IV. Numerical Results

In this section, we first compare the performance of the proposed ODSA and the subgradient method 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 = 100 m, and a PPP distribution of devices with average λ (devices/m^2). For R = 100 m, an average of $10,000\lambda\pi$ devices are uniformly distributed over a circular range with a radius of 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:

$$g_i(s,N) = \frac{(1/i)^s}{\sum_{k=1}^{N}(1/k)^s} \tag{28}$$

where s is a demand dominance factor. Figure 3 illustrates the distribution in (28) with varying s.

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 a high-priority-first selection (HPF caching policy). The EP policy is to cache all contents with equal probability, e.g., with the following probability:

$$p_i = M/N, \ \forall i \in \mathbf{I} \tag{29}$$

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

$$p_i = \begin{cases} 1 & \text{if } i \le M \\ 0 & \text{if } i > M \end{cases} \tag{30}$$

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

Figure 4 shows the size of the search space for ODSA as varying N with $M = 10$, $s = 1$, and $\lambda = 0.01/\pi$. As shown in Figure 4, the number of iterations for ODSA is bounded by $\lceil \log_2 N \rceil$, e.g., 14 for N = 10,000, while achieving the actual number of iterations below the bound by breaking the rule in Theorem 2.

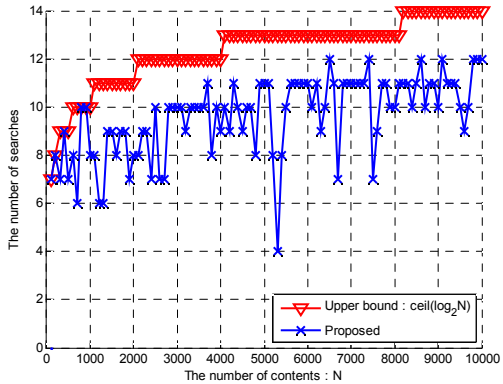

Figure 4. Computational complexity as varying the number of contents, N: $M = 10$, $s = 1$, $\lambda = 0.01/\pi$

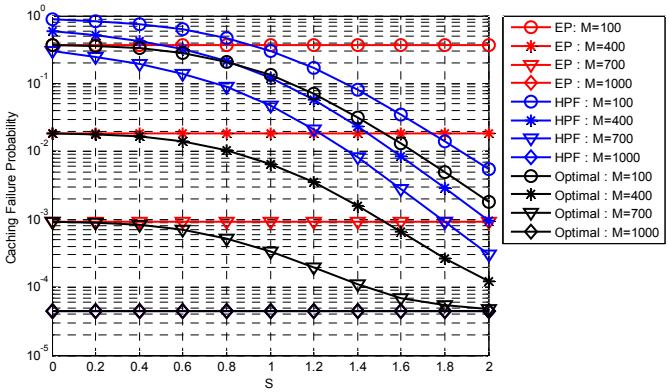

Figure 5. The average caching failure probability as varying the demand dominance factor s: $N = 1,000$, $R = 100m$, $\lambda = 0.001/\pi$

Figure 5 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.

Figure 6 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 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.

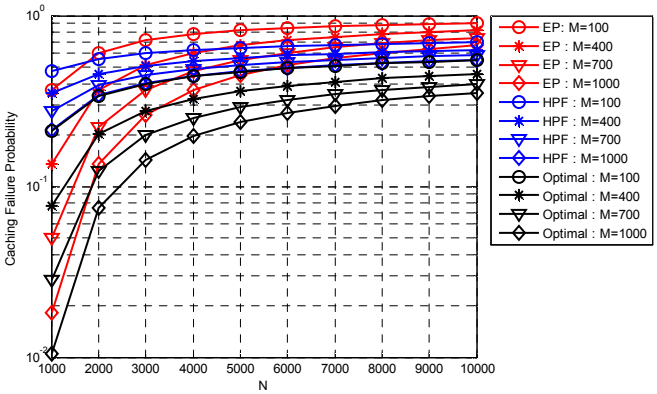

Figure 6. The average caching failure probability as varying the number of caching-service devices, N: $s = 1$, $R = 100m$, $\lambda = 0.001/\pi$

Taking the results from Figures 5 and 6 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 N, 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, since storage size is strictly limited in the mCDN, we attempted to determine which contents must be stored in CSDs, 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 be cached still 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. Even if we have not taken the power consumption issue into account for CSD in this paper, some means of selecting the CSD that leads to longer battery lifetime would be critical in practice.

As the small-cell approach becomes an essential means of coping with mobile traffic explosion, the practicality of mobile caching devices would be 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 leading to 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.

References

[1] N. Golrezaei, A. F. Molisch, A. G.. Dimakis, and G. Caire, "FemtoCaching and Device-to-Device Collaboration: A New Architecture for Wireless Video Distribution," arXiv:1204.1595v1 [cs.NI], Apr. 2012.

[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. Mühlethaler, "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. Networks, vol. 7, no. 6, pp.861-874, Dec. 1999.

[6] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-Device Communication as an Underlay to LTE-Advanced Networks," IEEE Commun. Mag., Dec. vol. 47, no. 12. pp. 42-94, Dec. 2009.

[7] IEEE P802.16.1a/D2, WirelessMAN-Advanced Air Interface for Broadband Access Systems-Draft Amendment: Higher Reliability Networks, Apr. 2012.

[8] http://www.statista.com/chart/1009/mobile-internet-traffic-growth/