Joint Optimization of Content Caching and Push in Renewable Energy Powered Small Cells

Jie Gong*, Sheng Zhou†, Zhenyu Zhou‡, Zhisheng Niu†
*School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
†Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
‡State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources,
School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China
Email: gongj26@mail.sysu.edu.cn

***Abstract*—In this paper, we explore the content information to design the joint caching and push mechanism in the small-cell base stations (SBSs) powered by renewable energy. The problem is formulated as a Markov decision process by exploring the features of content popularity and renewal and by taking into consideration the energy consumption for both content fetch from core network and push to the users. The objective is to minimize the number of requests which cannot be met by the SBSs. We adopt the policy iteration algorithm to obtain the optimal caching and push policy. According to the numerical results, the performance gain with large SBS cache size is marginal due to the limited energy. We also find that the optimal policy reveals noticeable performance gain compared with the greedy fetch policy and the non-push policy. In addition, simulations shows the tradeoff between the number of cached contents in the SBS and the available energy for content push.**

I. Introduction

By utilizing the renewable energy from natural sources such as solar, wind, kinetic activities and so on to support wireless communication systems, energy harvesting (EH) technology [1] has been considered as one of the candidate technologies to reduce the CO_2 emissions. However, due to the randomness of the energy arrival process and the limitation on the battery capacity, energy waste or shortage will occur when the energy harvesting process and the traffic pattern mismatches with each other in either spatial or time domain. To improve the efficiency of the harvested energy, one should adjust the power allocation policy using the traffic information to re-shape the energy profile to match the traffic profile.

In the literature, there have been extensive studies on how to match harvested energy with traffic profile. Refs. [2], [3] study the optimal power allocation with random packet arrival under additive white Gaussian noise (AWGN) channel and fading channel, respectively. The long-term spatial and temporal variations of mobile traffic are considered in [4], [5], and the renewable energy is adapted via resource allocation and base station (BS) sleeping. However, the existing research mainly depends on the bit-level information, i.e., if there are bits to be transmitted or how many bits need to be transmitted. The content information carried inside the bit stream is not explored. In fact, as users may be interested in the same content (latest news, popular videos and etc.), there are lots of repeated transmissions if the content information is not used, which is not energy-efficient.

The content caching and push mechanism is viewed as a promising way to improve the efficiency of content delivery in wireless network. To reduce the core network overhead, contents are suggested to be cached at the small-cell BSs (SBSs) [6], [7] or relay nodes [8] with proactive caching [9]. On the other hand, with the improvement of data storage capacity, user devices are capable of storing large amount of data. Hence, the content push mechanism [10] is developed based on wireless multicast [11], [12] which broadcasts commonly interested multimedia contents to multiple users simultaneously. People further analyzed the capacity gain by push in the integrated broadcast and communication network [13]. With renewable energy, ref. [14] uses EH based SBSs to cache contents for the deployment flexibility and energy consumption reduction, and ref. [15] designs the energy-aware resource allocation algorithm with limited content cache. However, joint content caching and push policy design using renewable energy is still an open problem.

In this paper, we combine the EH technology with the content caching and push by considering EH powered SBSs in heterogeneous cellular networks [16] under the recently proposed GreenDelivery framework [17]. Previous work [18] has studied the proactive push policy optimization. This paper further extends to joint optimization of the content caching and push considering caching overhead. Specifically, with the non-negligible energy consumption of fetching contents from the core network, the SBS can not cache all the contents due to the limited renewable energy. It needs to decide when to fetch, push or unicast contents depending on the energy condition. We optimize the joint caching and push policy using Markov decision process (MDP) [19] approach. Numerical results are provided to illustrate the influence of cache size under different parameter settings as well as the tradeoff between the number of cached contents in the SBS and the available energy for content push.

The rest of the paper is organized as follows. Section II presents the system model and the problem formulation. Section III provides MDP-based formulation and analysis, and obtains the optimal policy. Some numerical results are provided in Section IV for performance evaluation. Finally, Section V concludes the paper.

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

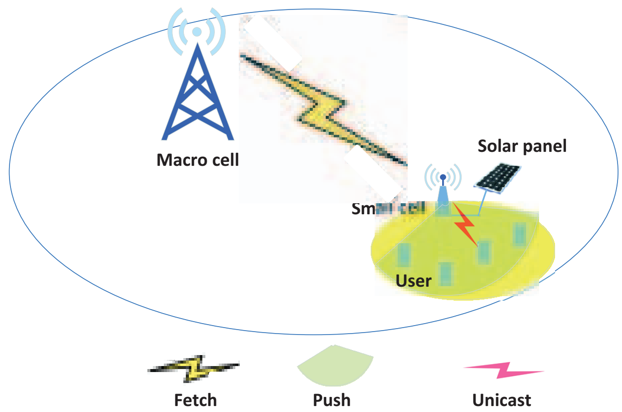

Fig. 1. Two-tier heterogeneous cellular network. There are actually densely deployed multiple small cells in a macro cell. Only one of them is depicted as we focus on a single small-cell analysis.

II. System Model and Problem Formulation

Consider a two-tier heterogeneous cellular network shown in Fig. 1, where a second-tier small-cell with radius R is considered. The macro-cell BS (MBS) is powered by the power grid and the SBS is powered by renewable energy, and the harvested energy can be stored in a battery with capacity $B_{\max}$. There is a dedicated wired/wireless backhaul link for the SBS to fetch contents from the core network through the MBS, which consumes a fixed amount of energy E_f. The SBS has a limited content cache size N. It can either unicast a cached content to the specific user, or multicast a popular content to all the users in its coverage with a fixed amount of energy E_p, i.e., push. The instantaneous transmission rate for a user is

$$r = W \log_2 \left(1 + \frac{P_t |h|^2 \beta d^{-\alpha}}{\sigma^2 + I}\right), \tag{1}$$

where W is the bandwidth of the SBS, P_t is the transmit power, h is the small-scale fast fading coefficient, β and α represent the pathloss constant and the pathloss exponent, respectively, d is the transmission distance, $\sigma^2 + I$ is the noise plus interference power. Once a content is pushed to the user side, the requests for this content can be immediately satisfied by the users' local cache. Then it is not necessary to transmit the data from the SBS or the MBS.

A. Content Distribution and Cache Model

The system is slotted with time slot length T_s. The contents are assumed of the same length and can be completely delivered in a time slot. The popularity of the contents varies from each other. Statistical researches have shown that the content popularity distribution is well fitted by the Zipf distribution [20]. Due to the limitation of the cache size in the SBS, we only focus on the top N popular contents. Specifically, the popularity of the i-th ranked content can be expressed as

$$f_i = \frac{1/i^v}{\sum_{j=1}^{N} 1/j^v}, \tag{2}$$

where $v \geq 0$ is the skew parameter. As people's interest changes over time, some contents may be outdated and replaced by new ones. Assume in each slot, a content leaves the system and is replaced by a new one with probability $p_c \in [0, 1]$. The leaving content is uniformly chosen among all the contents $1, 2, \cdots, N$.

There are two content cache states in this system, i.e., the number of cached contents in the SBS $\hat{C}_k$ and those at users $\tilde{C}_k$, where k is the time index. It is obvious that the most efficient cache strategy is to cache the most popular contents. As a result, we can use the number of cached content to denote the caching states, and $\hat{C}_k$ indicates that the contents cached in the SBS are those ranked from 1 to $\hat{C}_k$, where $\hat{C}_k \leq N$. It also holds for $\tilde{C}_k$. As in each slot, the SBS fetches or pushes at most one content, and at most one content leaves the system, we have

$$\hat{C}_{k+1} \in \{\max\{0, \hat{C}_k - 1\}, \hat{C}_k, \min\{N, \hat{C}_k + 1\}\}, \tag{3}$$
$$\tilde{C}_{k+1} \in \{\max\{0, \tilde{C}_k - 1\}, \tilde{C}_k, \min\{\hat{C}_k, \tilde{C}_k + 1\}\}. \tag{4}$$

B. SBS Action and Power Consumption Model

The SBS's action includes: fetch a content from the MBS, unicast the required content to a specific user, push a content to all users, or sleep. Assume that the link between the MBS and the SBS is orthogonal to the link between the SBS and the users. Hence, the fetch action can be taken simultaneously with push or unicast. The SBS's action can be denoted by $u_k = (\hat{u}_k, \tilde{u}_k)$, where

$$\hat{u}_k = \begin{cases} 0, & \text{do nothing} \\ 1, & \text{fetch a content} \end{cases} \tag{5}$$

$$\tilde{u}_k = \begin{cases} 0, & \text{do nothing} \\ 1, & \text{unicast the required content} \\ 2, & \text{push a content} \end{cases} \tag{6}$$

The user request is assumed to follow the Bernoulli distribution, i.e., there is a content request with probability $p_u \in [0, 1]$ in each time slot. The user generating the request is assumed uniformly distributed in the coverage of the SBS, and the required content is generated according to the popularity of the contents. To transmit a content from the SBS to the user with distance d, the required average data rate needs to be achieved, i.e., $r_0 = \mathbb{E}_h[r]$, where $\mathbb{E}_h$ is the expectation operator with respect to h, and r is calculated by (1). By solving it numerically, the required power for sending a content to a user can be obtained. Assume that the time slot T_s is much larger than the fast channel fading time. Hence, the randomness of the channel is averaged out and the transmit power is only distance-dependent, denoted by $P_t(d)$. Then the user request can be represented by $Q_k = P_t(d)T_s$, which is the amount of energy required for unicasting a content. Similarly, to push a content, it must be guaranteed that all the users in the small cell coverage can receive the content with rate r_0. So the user at cell edge (distance to the SBS is R) must be covered, i.e., $E_p = P_t(R)T_s$. We also set $Q_k = 0$ to indicate either there is no request or the requested content is available in the user's local cache. With the above energy consumption model, the battery energy state E_k is updated as

$$E_{k+1} = \min\{B_{\max}, E_k - U_k + A_k\}, \tag{7}$$

where U_k is the energy used for transmission which satisfies $U_k \leq E_k$, and A_k is the amount of harvested energy in period k, which is assumed ergodic and i.i.d. with average $\bar{A}$. U_k may includes the energy for fetch, push and unicast. If $\hat{u}_k = \tilde{u}_k = 0$, we have $U_k = 0$, i.e., the SBS turns to sleep and the energy consumption is negligiable.

At the beginning of each time slot, the SBS takes its action based on the system states. When a user requests a content that is not in its cache, and at the same time, the required content is not in the SBS's cache, or the SBS decides not to unicast the content, the request has to be handled by the MBS, which causes relatively higher energy consumption as the distance from the MBS to the user is larger. The ratio of user requests handled by the MBS is adopted as the performance metric. The lower ratio of user requests handled by the MBS is achieved, the more efficiently the harvested energy is used. The problem is formulated as follows.

C. SBS Blocking Probability Minimization

Our problem can be described as minimizing the ratio of user requests handled by the MBS over the total user requests by adjusting the behavior of the SBS under the energy constraint. Note that the ratio of user requests handled by the MBS is equivalent to the SBS blocking probability which is the probability that a user request is denied by the SBS. For the ease of description , we use the blocking probability as the objective instead of the ratio of user requests handled by the MBS in the rest of the paper. Mathematically, the objective is expressed as

$$\min \lim_{K \to +\infty} \frac{\bar{K}}{\tilde{K}}, \tag{8}$$

where K is the number of time slots, $\bar{K}$ is the number of blocked requests and $\tilde{K}$ is the number of requests. Notice that the blocking probability of the requests and the blocking probability over K time slots is related as

$$\frac{\bar{K}}{K} = \frac{\bar{K}}{\tilde{K}} \frac{\tilde{K}}{K} = \frac{\bar{K}}{\tilde{K}} p_u, \tag{9}$$

where recall that p_u is the content request probability. Hence for a given p_u, minimizing the blocking probability of user requests is equivalent with minimizing the blocking probability over time. We can reformulate our problem as

$$\min \lim_{K \to +\infty} \frac{\bar{K}}{K}. \tag{10}$$

III. Optimal Policy Design

To find the optimal solution for the problem (10), we need to decide the SBS's action based on the system state at the beginning of each time slot. MDP [19], also termed as dynamic programming (DP), is an effective tool to solve this type of problems and is widely used for the control optimization of stochastic process. It deals with the set of problems with controlled Markov process where the control action in each stage is based only on the current system state. In this paper, the term "stage" is equivalent with the term "time slot". A standard MDP problem contains the following elements: state, action, cost function, and state transition. Notice that the energy state and the channel state are continuous in our problem. To make the problem trackable, we first descretize the system state, and then re-formulate our problem as a MDP optimization problem and propose a policy iteration algorithm.

A. System State Descretization

The state of the system in stage k is denoted by

$$x_k = (E_k, Q_k, \hat{C}_k, \tilde{C}_k), \tag{11}$$

where as mentioned before, E_k is the battery energy state, Q_k is the user request state, $\hat{C}_k$ and $\tilde{C}_k$ are the cache states in the SBS and at users' side, respectively. Recall that $Q_k = 0$ if there is no user request or the requested content is already in the user's cache. Otherwise, Q_k represents the energy consumption for completing the required content transmission. For a user request generated at distance d, $Q_k = P_t(d)T_s$, where $P_t(d)$ is the transmission power obtained by solving (1) with $r = r_0$. Denote the state space as $\mathcal{S}$.

We discretize the state space into a finite set by discretizing the energy state and the user request state. The energy is discretized with unit energy E_{unit}. Then the energy state can be denoted by $E_k \in \{0, 1, \cdots, E_{\max}\}$ with $E_{\max}E_{\text{unit}} = B_{\max}$. $E_k = i$ corresponds to iE_{unit} amount of energy in the battery, and similarly, $A_k = i$ corresponds to iE_{unit} amount of energy arrived in stage k.

To discretize Q_k, we select a series of distances $0 < d_1 < d_2 < \cdots < d_M = R$ so that $P_t(d_i)T_s = l_i E_{\text{unit}}$ where l_i is a positive integer for any $i = 1, 2, \ldots, M$. For any user with distance to the SBS ranging from d_{i-1} to d_i, we unicast the required content with energy $P_t(d_i)T_s$, which guarantees the average data rate r_0 for all the users in this area. And we set $l_0 = 0$ denoting that there is no content transmission request. Then the user request state can be denoted by $Q_k \in \{0, 1, \cdots, M\}$, where $Q_k = i$ corresponds to the case that $l_i E_{\text{unit}}$ amount of energy is required for unicasting the content. Notice that the required content is not always available in the SBS as $\hat{C}_k \leq N$. In this sense, we further set $Q_k = -1$ if the required content is not cached in the SBS, which will cause a blocking event. The energy for push is quantized as $E_p = l_M E_{\text{unit}}$. Also we have $E_f = l_j E_{\text{unit}}$, where the value of j depends on the link capacity.

B. MDP Problem Formulation and Optimization

Besides the system state described in the previous subsection, we need to further clarify the action, the cost function and the state transition to complete the MDP problem formulation. The action u_k has been modeled as (5) and (6). Notice that in different states, the SBS may not be able to take all the three actions due to the limitation of available energy. A simple example is that if $E_k = 0$, the SBS can do nothing but sleep, i.e., $\hat{u}_k = \tilde{u}_k = 0$. Hence, the action space is state-dependent, which can be expressed as $u_k \in \mathcal{U}_k(x_k)$.

The cost function depends on both the system state and the action, denoted by $g_k(x_k, u_k)$. In our problem, the cost happens if and only if there is a user request handled by the MBS. Hence, we have $g_k(x_k, u_k) \in \{0, 1\}$. $g_k(x_k, u_k) = 1$ if the user request is handled by the MBS, and $g_k(x_k, u_k) = 0$ otherwise. We can express it as

$$g_k(x_k, u_k) = \begin{cases} 1, & \text{if } Q_k = -1, \text{ or } Q_k > 0, u_k \neq 1 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

The state transition is expressed as

$$
\begin{aligned}
&p_{x_k \to x_{k+1}|u_k} \\
=&\Pr(E_{k+1}, Q_{k+1}, \hat{C}_{k+1}, \tilde{C}_{k+1} | E_k, Q_k, \hat{C}_k, \tilde{C}_k, u_k) \\
=&\Pr(E_{k+1}|E_k, Q_k, u_k) \cdot \Pr(\hat{C}_{k+1}, \tilde{C}_{k+1}|\hat{C}_k, \tilde{C}_k, u_k) \cdot \\
&\quad \Pr(Q_{k+1}|\hat{C}_{k+1}, \tilde{C}_{k+1}),
\end{aligned} \tag{13}
$$

where the second equality is derived by the definition of conditional probability and the fact that for the given action u_k, E_{k+1} only depends on E_k and Q_k according to (7), $\hat{C}_{k+1}$ depends on $\hat{C}_k$ and fetch action $\hat{u}_k$ according to (3), and $\tilde{C}_{k+1}$ depends on both cache states $\hat{C}_k$ and $\tilde{C}_k$ as well as push action $\tilde{u}_k$ according to (4), and they correlate with each other due to the content updating process. While Q_{k+1} only depends on $\hat{C}_{k+1}$ and $\tilde{C}_{k+1}$ as they jointly decide the probability with which a requested content has been pushed, hence influence the probability with which a unicast is required.

We calculate the state transition probability according to (13). Firstly, to calculate the energy state transition probability $\Pr(E_{k+1}|E_k, Q_k, u_k)$, we denote $p_a(i), i = 0, 1, \cdots$ as the probability that iE_{unit} amount of energy is arrived, which satisfies $p_a(i) \in [0, 1], \sum_i p_a(i) = 1$. To simplify the description, we set $p_a(i) = 0, \forall i = -1, -2, \cdots$. Then we have

$$
\Pr(E_{k+1}|E_k, Q_k, u_k) =
\begin{cases}
p_a(E_{k+1} - E_k), & \text{if } E_{k+1} < E_{\max}, u_k = (0,0) \\
1 - \sum_{i=0}^{E_{\max} - E_k - 1} p_a(i), & \text{if } E_{k+1} = E_{\max}, u_k = (0,0) \\
p_a(E_{k+1} - E_k + l_{Q_k}), & \text{if } E_{k+1} < E_{\max}, u_k = (0,1) \\
1 - \sum_{i=0}^{E_{\max} - E_k + l_{Q_k} - 1} p_a(i), & \text{if } E_{k+1} = E_{\max}, u_k = (0,1) \\
p_a(E_{k+1} - E_k + l_M), & \text{if } E_{k+1} < E_{\max}, u_k = (0,2) \\
1 - \sum_{i=0}^{E_{\max} - E_k + l_M - 1} p_a(i). & \text{if } E_{k+1} < E_{\max}, u_k = (0,2) \\
p_a(E_{k+1} - E_k + l_j), & \text{if } E_{k+1} < E_{\max}, u_k = (1,0) \\
1 - \sum_{i=0}^{E_{\max} - E_k + l_j - 1} p_a(i), & \text{if } E_{k+1} = E_{\max}, u_k = (1,0) \\
p_a(E_{k+1} - E_k + l_j + l_{Q_k}), & \text{if } E_{k+1} < E_{\max}, u_k = (1,1) \\
1 - \sum_{i=0}^{E_{\max} - E_k + l_j + l_{Q_k} - 1} p_a(i), & \text{if } E_{k+1} = E_{\max}, u_k = (1,1) \\
p_a(E_{k+1} - E_k + l_j + l_M), & \text{if } E_{k+1} < E_{\max}, u_k = (1,2) \\
1 - \sum_{i=0}^{E_{\max} - E_k + l_j + l_M - 1} p_a(i). & \text{if } E_{k+1} = E_{\max}, u_k = (1,2)
\end{cases} \tag{14}
$$

Note that the action $u_k = (\hat{u}_k, \tilde{u}_k) = (0, 0)$ can be taken in any states, while $u_k = (0, 1)$ can be taken under the condition that $Q_k \geq 1$ and $l_{Q_k} \leq E_k$, and $u_k = (0, 2)$ with condition $l_M \leq E_k$ and $\tilde{C}_k < \hat{C}_k$. On the other hand, $u_k = (1, 0)$ can be taken if $\hat{C}_k < N$ and $l_j \leq E_k$, $u_k = (1, 1)$ can be taken if $\hat{C}_k < N$, $Q_k \geq 1$ and $l_{Q_k} + l_j \leq E_k$, and $u_k = (1, 2)$ with condition $\hat{C}_k < N$, $l_M + l_j \leq E_k$ and $\tilde{C}_k < \hat{C}_k$. Also note that when $E_{k+1} = E_{\max}$, the energy arrival may exceed the battery capacity. In this case, the probability is calculated by summarizing all the possible energy arrival conditions.

Secondly, we calculate the content updating probability $\Pr(\hat{C}_{k+1}, \tilde{C}_{k+1}|\hat{C}_k, \tilde{C}_k, u_k)$. Notice that the content states $\hat{C}_k$ and $\tilde{C}_k$ can only increase or decrease by 1 or keep constant. For notation simplicity, denote $C_{k+1} = (\hat{C}_{k+1}, \tilde{C}_{k+1})$. Depending on the content updating process, the cache state is updated as

$$
\Pr(C_{k+1}|\hat{C}_k, \tilde{C}_k, u_k) =
\begin{cases}
p_c \frac{\tilde{C}_k}{N}, & \text{if } \hat{u}_k = 0, \tilde{u}_k < 2, C_{k+1} = (\hat{C}_k - 1, \tilde{C}_k - 1) \\
& \text{or } \hat{u}_k = 1, \tilde{u}_k < 2, C_{k+1} = (\hat{C}_k, \tilde{C}_k - 1) \\
p_c \frac{\tilde{C}_k + 1}{N}, & \text{if } \hat{u}_k = 0, \tilde{u}_k = 2, C_{k+1} = (\hat{C}_k - 1, \tilde{C}_k) \\
& \text{or } \hat{u}_k = 1, \tilde{u}_k = 2, C_{k+1} = (\hat{C}_k, \tilde{C}_k) \\
p_c \frac{\hat{C}_k - \tilde{C}_k}{N}, & \text{if } \hat{u}_k = 0, \tilde{u}_k < 2, C_{k+1} = (\hat{C}_k - 1, \tilde{C}_k) \\
& \text{or } \hat{u}_k = 1, \tilde{u}_k = 2, C_{k+1} = (\hat{C}_k, \tilde{C}_k + 1) \\
p_c \frac{\hat{C}_k - \tilde{C}_k + 1}{N}, & \text{if } \hat{u}_k = 1, \tilde{u}_k < 2, C_{k+1} = (\hat{C}_k, \tilde{C}_k) \\
p_c \frac{\hat{C}_k - \tilde{C}_k - 1}{N}, & \text{if } \hat{u}_k = 0, \tilde{u}_k = 2, C_{k+1} = (\hat{C}_k - 1, \tilde{C}_k + 1) \\
1 - p_c \frac{\hat{C}_k}{N}, & \text{if } \hat{u}_k = 0, \tilde{u}_k < 2, C_{k+1} = (\hat{C}_k, \tilde{C}_k) \\
& \text{or } \hat{u}_k = 0, \tilde{u}_k = 2, C_{k+1} = (\hat{C}_k, \tilde{C}_k + 1) \\
1 - p_c \frac{\hat{C}_k + 1}{N}, & \text{if } \hat{u}_k = 1, \tilde{u}_k < 2, C_{k+1} = (\hat{C}_k + 1, \tilde{C}_k) \\
& \text{or } \hat{u}_k = 1, \tilde{u}_k = 2, C_{k+1} = (\hat{C}_k + 1, \tilde{C}_k + 1) \\
0, & \text{else}
\end{cases} \tag{15}
$$

For notation simplicity, we omit the boundary constraints on the cache states. Specifically, if $\hat{u}_k = 1$, there is an additional constraint that $\hat{C}_k < N$. If $\tilde{u}_k = 2$, we have $\tilde{C}_k < \hat{C}_k$. Also, $\hat{C}_{k+1}$ and $\tilde{C}_{k+1}$ are nonnegative integers, and always satisfy $\hat{C}_{k+1} \geq \tilde{C}_{k+1}$.

Finally, the user request state transition is

$$
\Pr(Q_{k+1}|\hat{C}_{k+1}, \tilde{C}_{k+1}) =
\begin{cases}
p_u (1 - \sum_{i=1}^{\hat{C}_{k+1}} f_i), & \text{if } Q_{k+1} = -1 \\
(1 - p_u) + p_u \sum_{i=1}^{\tilde{C}_{k+1}} f_i, & \text{if } Q_{k+1} = 0 \\
p_u \sum_{i=\tilde{C}_{k+1}+1}^{\hat{C}_{k+1}} f_i \frac{d_m^2 - d_{m-1}^2}{R^2}, & \text{if } Q_{k+1} = m > 0
\end{cases} \tag{16}
$$

where f_i is calculated according to (2) and $d_0 = 0$. As the users are assumed uniformly distributed in the cell, the request is generated with distance to the SBS ranging from d_{m-1} to d_m with the probability equal to the ratio of the circular ring area to the cell area.

Based on the above MDP-based system modeling, the original optimization problem (10) can be re-written as

$$
\min \lim_{K \to +\infty} \frac{1}{K} \mathbb{E} \left[\sum_{k=0}^{K-1} g(x_k, u_k(x_k)) \right]. \tag{17}
$$

The expectation operation is taken over all the random parameters including energy arrival, user request, and content update. The optimization is taken over all the possible policies $\{u_1, u_2, \cdots\}$. It can be proved that for any two states, there is a stationary policy $\boldsymbol{u}$ so that one state can be accessed with non-zero probability from the other with finite steps. Consequently, the optimization is irrelevant with the initial state x_0, and there exists an optimal stationary policy $\boldsymbol{u}^*$ [19, Sec 4.2].

According to [19, Prop. 4.2.1], the optimal average cost λ^* together with some vector $\boldsymbol{h}^* = \{h^*(x)|x \in \mathcal{S}\}$ satisfies the Bellman's equation

$$
\lambda^* + h^*(x) = \min_{u \in \mathcal{U}(x)} \left[g(x, u) + \sum_{y \in \mathcal{S}} p_{x \to y|u} h^*(y) \right]. \tag{18}
$$

Further more, if $u^*(x)$ attains the minimum value of (18) for each x, the stationary policy $\boldsymbol{u}^*$ is optimal. Based on the Bellman's equation, instead of the long term average cost minimization, we only need to deal with (18) which only relates with per-stage cost $g(x,u)$ and state transition $p_{x\to y|u}$. The *policy iteration algorithm* [19, Sec. 4.4] can effectively solve the problem. It starts with any feasible stationary policy, and improves the objective step by step. Suppose in the j-th step, we have a stationary policy denoted by $\boldsymbol{u}^{(j)}$. Based on this policy, we perform *policy evaluation* [19, Sec. 4.4] step, i.e., we solve the following linear equations

$$\lambda^{(j)}+h^{(j)}(x)=g(x,u^{(j)}(x))+\sum_{y\in\mathcal{S}}p_{x\to y|u^{(j)}(x)}h^{(j)}(y) \quad (19)$$

for $\forall x\in\mathcal{S}$ to get the average cost $\lambda^{(j)}$ and vector $\boldsymbol{h}^{(j)}$. By setting

$$h^{(j)}(E_{\max}+1,M+1,N+1,N+1)=0, \quad (20)$$

a unique solution for (19) exists.

As $\boldsymbol{u}^{(j)}$ may not be the optimal policy, we subsequently perform *policy improvement* [19, Sec. 4.4] step to find the policy $\boldsymbol{u}^{(k+1)}$ which minimizes the right hand side of Bellman's equation

$$u^{(j+1)}(x)=\arg\min_{u\in\mathcal{U}(x)}\left[g(x,u)+\sum_{y\in\mathcal{S}}p_{x\to y|u}h^{(j)}(y)\right]. \quad (21)$$

If $\boldsymbol{u^{(j+1)}}=\boldsymbol{u^{(j)}}$, the algorithm terminates, and the optimal policy is obtained $\boldsymbol{u^*}=\boldsymbol{u^{(j)}}$. Otherwise, repeat the procedure by replacing $\boldsymbol{u^{(j)}}$ with $\boldsymbol{u^{(j+1)}}$. It is proved that the policy iteration algorithm terminates in finite number of iterations [19, Prop. 4.4.1].

IV. Numerical Results

In this section, we run some numerical simulations for performance evaluation. We set the cell radius $R=50$m, the required content delivery spectrum efficiency $r_0/W=1$bps/Hz, the pathloss parameters $\beta=10$dB and $\alpha=2$, and the Zipf parameter $v=1$. The maximum transmit power or equivalently the transmit power for cell-edge user is set $P_t(R)=1$Watt. The channel coefficient h follows Rayleigh fading, whose mean value and the interference plus noise power σ^2+I are set so that (1) holds for $r=r_0, d=R, P_t=P_t(R)$. The users are sorted into $M=3$ classes and the energy is quantized with unit $E_{\text{unit}}=\frac{P_t(R)T_s}{M}$. Hence, the number of energy units for push is M, and the user request in class m consumes m units of energy. The quantized battery capacity is set to $E_{\max}=12$. Assume the energy arrival process follows a Poisson distribution with average arrival rate $\bar{A}$ units of energy.

Firstly, we study the influence of the cache size N on the performance. We set $N=1,2,\ldots,N_{\max}$ and run the policy iteration algorithm for each N. For comparison fairness, we study the blocking probability associated with the fixed number of contents $N_{\max}=15$. In this sense, the user requests are divided into two parts. The first is the set of requests for the contents ranked in top N. The blocking of these requests is calculated by the proposed algorithm. The second contains the rest requests, which definitely causes blocking as they will never be cached in the SBS. We combine the two parts and calculate the overall blocking probability, which is depicted in Fig. 2. It can be seen that the blocking probability continuously degrades if there is no fetch energy consumption, as we can cache as many contents as possible in the SBS to improve the performance without additional cost. While if the fetch energy E_f is not negligible, the blocking probability is almost constant when the cache size becomes large. In this case, the system is energy limited. Setting up a large cache size in the SBS is not necessary. In addition, Fig. 3 shows that with different energy arrival rates, the speed with which the performance gain becomes marginal varies. Specifically, the performance gain becomes marginal with $N\geq 7$ for $\bar{A}=0.4$. While for $\bar{A}=0.8$, there is still noticeable gain when $N=12$.

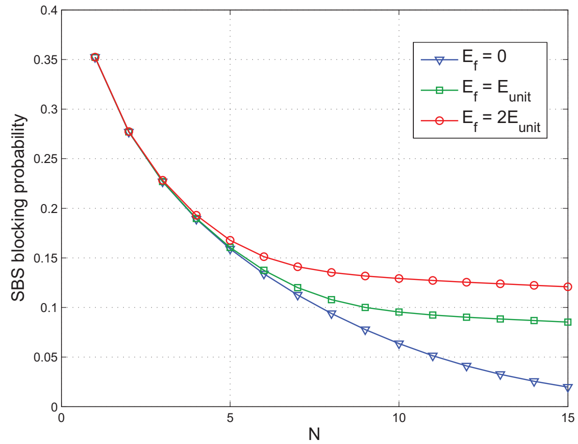

Fig. 2. The influence of the cache size in the SBS on the performance with various fetching energy. $p_u=0.5, p_c=0.3, \bar{A}=0.6$.

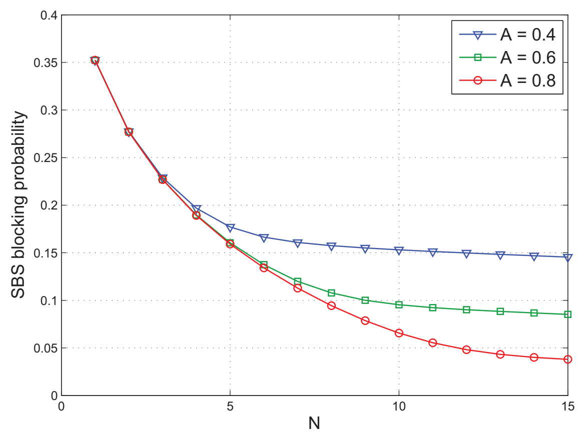

Fig. 3. The influence of the cache size in the SBS on the performance with various energy arrival rate. $p_u=0.5, p_c=0.3, E_f=E_{\text{unit}}, \bar{A}=A$.

We further compare the proposed algorithm with some heuristic algorithms to demonstrate the importance of joint optimization of content caching and push. We consider the following baseline algorithms: *greedy fetch policy*, in which the SBS always fetches contents as long as there is sufficient energy and the cache in the SBS is not full, *threshold fetch*

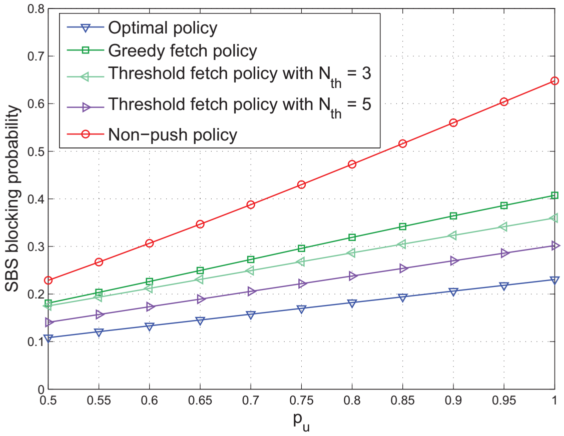

Fig. 4. Comparison with some heuristic algorithms. $p_u = 0.5, p_c = 0.3, \bar{A} = 0.6, E_f = E_{\text{unit}}, N = 8$.

policy, in which the SBS fetches contents if the cache size in the SBS does not achieve a pre-defined threshold N_{th}, and *non-push policy*, in which the SBS only unicasts the required contents to the users on demand. Notice that the greedy fetch policy is equivalent to the threshold fetch policy with $N_{\text{th}} = N$. Compared with greedy fetch policy, our optimal policy reduces the SBS blocking probability by more than 40%. The results of threshold fetch policies with different N_{th} show a tradeoff between the number of cached contents in the SBS and the available energy for content push. Larger N_{th} allows more contents available in the SBS, but reduces the energy available for push/unicast, and vice versa. Carefully choosing N_{th} can achieve the minimum blocking probability. In our settings, the optimal threshold is $N_{\text{th}} = 5$. In addition compared with non-push policy, the greedy fetch policy can achieve more than 20% blocking probability reduction, which illustrates the great performance improvement by push.

V. Conclusion

In this paper, content caching and push mechanism in EH-powered SBSs is jointly optimized. To apply MDP tools, we descretize the system energy and the user request into finite states. The proposed policy iteration algorithm solves the problem, and the obtained optimal policy performs much better than the heuristic greedy fetch policy and non-push policy. As the renewable energy is limited, the cache size in the SBS does not need to be large, which greatly simplifies the hardware design in the SBS. Also, using more energy to fetch improves the content availability in the SBS, but degrades the energy availability for push/unicast. The proposed optimal policy well balances the content availability and the energy availability. Future work includes the extension to the multiple SBSs case, where energy-aware distributed content caching and cooperative push needs to be further investigated.

Acknowledgment

This work is sponsored in part by the National Basic Research Program of China (973 Program: No.2012CB316001), the Nature Science Foundation of China (61571265, 61321061, 61461136004) and Hitachi R&D Headquarter.

References

[1] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, "Designing intelligent energy harvesting communication systems," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 210–216, Jan. 2014.

[2] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Trans. on Communications*, vol. 60, no. 1, pp. 220–230, Jan. 2012.

[3] J. Gong, S. Zhou, and Z. Niu, "Optimal power allocation for energy harvesting and power grid coexisting wireless communication systems," *IEEE Trans. on Communications*, vol. 61, no. 7, pp. 3040–3049, Jul. 2013.

[4] T. Han and N. Ansari, "On optimizing green energy utilization for cellular networks with hybrid energy supplies," *IEEE Trans. on Wireless Communications*, vol. 12, no. 8, pp. 3872–3882, Aug. 2013.

[5] J. Gong, J. Thompson, S. Zhou, and Z. Niu, "Base station sleeping and resource allocation in renewable energy powered cellular networks," *IEEE Trans. on Communications*, vol. 62, no. 11, pp. 3801–3813, Nov. 2014.

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

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

[8] X. Wang, Y. Bao, X. Liu, and Z. Niu, "On the design of relay caching in cellular networks for energy efficiency," in *IEEE Conference on Computer Communications Workshops*, Apr. 2011.

[9] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[10] I. Podnar, M. Hauswirth, and M. Jazayeri, "Mobile push: Delivering content to mobile users," in *Proc. 22nd International Conference on Distributed Computing Systems Workshops*, 2002.

[11] Z. Niu, L. Long, J. Song, and C. Pan, "A new paradigm for mobile multimedia broadcasting based on integrated communication and broadcast networks," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 126–132, Jul. 2008.

[12] J. Liu, W. Chen, Y. J. Zhang, and Z. Cao, "A utility maximization framework for fair and efficient multicasting in multicarrier wireless cellular networks," *IEEE/ACM Trans. on Networking*, vol. 21, no. 1, pp. 110–120, Feb. 2013.

[13] K. Wang, Z. Chen, and H. Liu, "Push-based wireless converged networks for massive multimedia content delivery," *IEEE Trans. on Wireless Communications*, vol. 13, no. 5, pp. 2894–2905, May 2014.

[14] N. Sharma, D. Krishnappa, D. Irwin, M. Zink, and P. Shenoy, "Greencache: Augmenting off-the-grid cellular towers with multimedia caches," in *Proc. ACM MMsys13*, Feb. 2013.

[15] A. Kumar and W. Saad, "On the tradeoff between energy harvesting and caching in wireless networks," in *IEEE International Conference on Communications (ICC), London, U.K.*, 2015.

[16] G. Piro, M. Miozzo, G. Forte, N. Baldo, L. Grieco, G. Boggia, and P. Dini, "HetNets powered by renewable energy sources: Sustainable next-generation cellular networks," *IEEE Internet Computing*, vol. 17, no. 1, pp. 32–39, Jan. 2013.

[17] S. Zhou, J. Gong, Z. Zhou, W. Chen, and Z. Niu, "Greendelivery: Proactive content caching and push with energy-harvesting-based small cells," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 142–149, Apr. 2015.

[18] J. Gong, S. Zhou, Z. Zhou, and Z. Niu, "Proactive push with energy harvesting based small cells in heterogeneous networks," in *IEEE International Conf. on Communications, London, U.K.*, 2015.

[19] D. P. Bertsekas, *Dynamic programming and optimal control, Volume II,* 3rd edition. Athena Scientific Belmont, MA, 2005.

[20] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. 7th ACM SIGCOMM Conference on Internet measurement*. ACM, 2007.