

Proactive Push with Energy Harvesting Based Small Cells in Heterogeneous Networks

Jie Gong*, Sheng Zhou*, Zhenyu Zhou[†], Zhisheng Niu*

*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: gongj13@mail.tsinghua.edu.cn

Abstract—Motivated by the recent development of energy harvesting communications, and the trend of multimedia contents caching and push at the access edge and user terminals, this paper considers how to design an effective push mechanism of energy harvesting powered small-cell base stations (SBSs) in heterogeneous networks. The problem is formulated as a Markov decision process by optimizing the push policy based on the battery energy, user request and content popularity state to maximize the service capability of SBSs. We extensively analyze the problem and propose an effective policy iteration algorithm to find the optimal policy. According to the numerical results, we find that the optimal policy reveals a state dependent threshold based structure. Besides, more than 50% performance gain is achieved by the optimal push policy compared with the non-push policy.

I. INTRODUCTION

Due to the rapidly growing multimedia traffic over the air and the critical concern regarding CO₂ emissions, green wireless communications are urgently required. There have been some candidate technologies which are demonstrated as effective ways to achieve green wireless access, such as energy harvesting (EH), multicast and heterogeneous networks. EH technology [1], [2], which utilizes the energy from natural sources such as solar, wind, and kinetic activities, can greatly reduce the wireless communication power consumption from the conventional power supply, i.e., power grid. Wireless multicast [3], [4] holds the promise of achieving huge energy efficiency gain via delivering commonly interested multimedia contents to multiple users simultaneously by broadcasting a single data stream to different users, which avoids duplicated retransmissions of the same content. Heterogeneous networks provide higher data rate to users by cutting down the distance between users and base stations (BSs) with densely deployed small-cell BSs (SBSs). However, each technology has its limitations based on the state of the art. Because of the limited battery capacity, energy waste or shortage will occur when energy and traffic arrivals mismatch with each other. On the other hand, to enable wireless multicast, some user requests need be delayed to wait for concurrent transmission, which may severely damage the quality of service (QoS) of the earlier demands. Finally, the deployment of SBSs is not flexible as it may cause high cost for deploying the supporting power lines and high-speed backhaul links.

To break through the limitations for higher energy efficiency, we introduce the proactive push mechanism [5] to combine

the technologies mentioned above. Powering the SBSs with EH devices greatly increases the flexibility of heterogeneous network deployment. Based on the EH status and content popularity distribution, the SBSs proactively cache and push the contents earlier than the actual demands. In reward, the time duration in which the desired content can be delivered is greatly extended, so that the delivery can flexibly match to the EH process. On the other hand, from the energy point of view, as the harvested energy can be effectively and timely used, the energy waste due to the battery capacity limitation can be avoided. In other words, proactive push is a novel way of information and power transfer over the hyper dimension of space (small cell to users) and time (now to the future) respectively, which is different from the joint transfer over space only [6].

Proactive push is supported by the recent trends on the development of last-mile wireless access hardwares and mobile devices. To reduce the core network overhead and enhance user experiences in terms of delay and rate, contents are suggested to be cached at the SBSs [7], [8] or relay nodes [9], with proactive caching schemes [10]. Also, there have been some commercial products such as HiWiFi [11] with large storage for caching. On the other hand, with the rapid improvement of data storage capacity, user devices are capable of storing large amount of data for potential user requests. And the network capacity gain provided by proactive push in the integrated broadcast and communication network is analyzed in Ref. [12]. With the large user storage capacity and the available contents at the edge nodes, proactive push by EH powered SBSs is considered to be practical and effective.

Recently, EH based SBSs are used to cache contents [13] for the deployment flexibility and energy consumption reduction, and the GreenDelivery framework for content delivery with EH powered small cells is proposed in [14]. As far as we know, the proactive push optimization is still an open problem in EH powered SBSs. And the problem is not trivial since it needs to jointly consider energy state, traffic load as well as content popularity. Pushing a content to a set of users typically consumes more energy than unicasting a required content to a single user as push needs to guarantee the data rate of the worst-channel user. While the more contents are pushed, the fewer unicast requests are generated since more contents can be found in users' local storage. Hence, there is a tradeoff between high energy consumption and low request generation rate by push mechanism which needs extensive study.

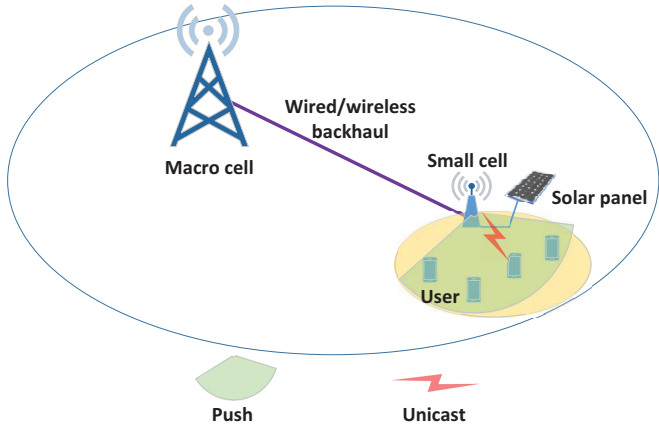


Fig. 1. Two-tier heterogeneous cellular network. There are multiple small cells in a macro cell. Only one of them is depicted as we focus on a single small-cell analysis.

In this paper, we try to optimize the proactive push policy of a EH powered SBS in heterogeneous wireless networks. The objective is to minimize the ratio of user requests handled by the macro BS (MBS) which happens when the SBS is of low energy or is pushing another content. We formulate the problem using Markov decision process (MDP) [15] tool with detailed modeling of state, action, cost function and state transition probability, and find the optimal stationary policy via policy iteration algorithm. Numerical results are provided to illustrate the structure of optimal policy and the performance gain compared with non-push policy.

The rest of the paper is organized as follows. Section II presents the system model. The problem is formulated and analyzed in Section III. Some numerical results are provided in Section IV for performance evaluation. Finally, Section V concludes the paper.

II. SYSTEM MODEL

We consider a second-tier small-cell with radius R in a two-tier heterogeneous cellular network as shown in Fig. 1. The SBS is powered by renewable energy solely, and the harvested energy can be stored in a battery with finite capacity B_{\max} . There is only one frequency channel for data transmission in each small cell, and the SBS has a high-speed wired/wireless backhaul link to the MBS to fetch any content immediately when required. When the SBS has sufficient energy, it can either unicast a required content to the specific user who requires it, or multicast a popular content to all the users in its coverage, i.e., push. When the battery energy is not enough, the SBS enters into sleep mode and the content request will be handled by the MBS. The SBS can also choose to sleep even though the battery is sufficient for transmission. In this way, there will be more energy available in the later times. At the user side, if a required content is in the users cache, it can directly access the content and does not need to trigger a transmission from the SBS or the MBS. In this paper, we assume each user has sufficient caching capacity so that any pushed contents can be successfully stored, and focus on how to design push policy to fully utilize the renewable energy in SBSs.

Assume there are a total of N contents of equal length that the users are interested in. Each content has a minimum average data rate requirement r_0 . Hence, the content transmission time is identical for all contents if they are transmitted with rate r_0 . Then the system is slotted with the length of each period equal to the content transmission time, denoted by T_p . 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 [16], [7]. Specifically, the popularity of the i -th ranked content among the N contents can be expressed as

$$f_i = \frac{1/i^v}{\sum_{j=1}^N 1/j^v}, \quad (1)$$

where $v \geq 0$ is the skew parameter. In the real network, people are more interested in the contents with higher popularity, which will result in higher request probability. In addition, as people's interest changes over time, some contents may be outdated and replaced by new ones. We assume in each period the probability that a piece of content leaves the system and is replaced by a new one is $p_c \in [0, 1]$. The leaving content is randomly chosen among all the contents $1, 2, \dots, N$.

In each period, there is a content request with probability $p_u \in [0, 1]$. And the user generating the request is assumed uniformly distributed in the small cell. The channel model considers large-scale pathloss effect as well as small-scale fast fading. For each content transmission, the data rate can be calculated as

$$r = \mathbf{E}_h \left[W \log_2 \left(1 + \frac{P_t |h|^2 \beta d^{-\alpha}}{\sigma^2 + I} \right) \right], \quad (2)$$

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. Assume the SBSs and the MBS are allocated with orthogonal frequency bands. As a result, there is no inter-tier interference, and the interference is only caused by the randomly and densely deployed SBSs working in the same frequency band. Hence, according to the law of large numbers, the noise plus interference together can be considered as additive white Gaussian noise (AWGN) with variance $\sigma^2 + I$. \mathbf{E}_h is the expectation operator with respect to h . Based on the channel model, the required power for sending a content to a user with distance d can be obtained by setting $r = r_0$ and solving (2) numerically.

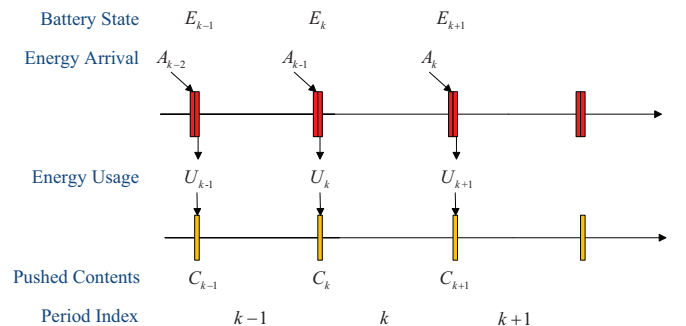


Fig. 2. Timeline of the slotted system.

Next, we describe the slotted system model in detail. As illustrated in Fig. 2, at the beginning of the period indexed by k , the battery energy is denoted by E_k , and the number of pushed contents is denoted by C_k . In our analysis, we always push the most popular contents to the users. According to this simple push policy, the pushed contents are those ranked from 1 to C_k . Based on the situation that whether there is a user request or not, whether the requested content is pushed or not, and how much energy is required for unicasting the required content, the SBS decides its action, i.e., unicast the required content, push a content, or sleep. Then at the beginning of the next period $k + 1$, the battery energy state is updated as

$$E_{k+1} = \min\{B_{\max}, E_k - U_k + A_k\}, \quad (3)$$

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 i.i.d. If the SBS decides to sleep, $U_k = 0$. C_{k+1} takes values as

$$C_{k+1} \in \{\max\{0, C_k - 1\}, C_k, \min\{N, C_k + 1\}\} \quad (4)$$

according to the SBS's action and the content update behavior. At this moment, the SBS takes its action based on the renewed system status. When a user requests a content that is not in its cache, but the SBS decides not to unicast for some reason, it needs to be handled by the MBS, which causes additional energy and resource allocation from the MBS. Intuitively, the ratio of user requests handled by the MBS indicates the harvested energy utilization efficiency. The lower ratio of user requests handled by the MBS is achieved, the more efficiently the harvested energy is used. In the next section, we will provide the problem formulation aiming at minimizing the ratio.

III. PROBLEM FORMULATION AND MDP SOLUTION

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. Mathematically, the objective can be expressed as

$$\min \lim_{K \rightarrow +\infty} \frac{\bar{K}}{K}, \quad (5)$$

where \bar{K} is the number of user requests handled by the MBS and K is the number of total periods. Notice that the objective function in (5) is not the ratio of user requests handled by the MBS over the total user requests but related with it as

$$\frac{\bar{K}}{K} = \frac{\bar{K}}{\bar{K}} \frac{\bar{K}}{K} = \frac{\bar{K}}{\bar{K}} p_u, \quad (6)$$

where \bar{K} is the total number of user requests during K time periods, and recall that p_u is the content request probability. For a given p_u , minimizing the ratio of user requests handled by the MBS over the total user requests is equivalent with (5).

To solve the problem (5), we need to decide the SBS's action in each period. As the per-period action is determined based on the system state at the beginning of each period, the problem can be modeled as a MDP optimization problem [15]. MDP, also termed as dynamic programming (DP) [15], is an effective tool and 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 stage¹ is based only on the current system state. A standard MDP problem contains the following elements: state, action, cost function, and state transition. Next, we re-formulate our problem as a MDP optimization problem by describing the elements one by one.

1) *System State*: The state of the system in stage k is denoted by

$$x_k = (E_k, Q_k, C_k), \quad (7)$$

where as mentioned before, E_k and C_k is the battery energy and the number of pushed contents, respectively. Q_k is the user request state. We set $Q_k = 0$ 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 , $Q_k = P_t(d)T_p$, where $P_t(d)$ is the transmission power obtained by solving (2) with $r = r_0$. Denote the state space as \mathcal{S} .

As energy and user locations take continuous values, there is a continuous state space, which makes the problem difficult to be solved. So we further discretize the state space \mathcal{S} into a finite set to make the problem tractable. The energy is discretized with unit energy E_{unit} . Then the energy state is $E_k \in \{0, 1, \dots, E_{\max}\}$ with $E_{\max}E_{\text{unit}} = B_{\max}$. $E_k = i$ corresponds to iE_{unit} amount of energy, and similarly for energy arrival A_k . To discretize Q_k , we select a series of distances $0 < d_1 < d_2 < \dots < d_M = R$ so that $P_t(d_i)T_p = l_i E_{\text{unit}}$ where l_i is a positive integer for any $i = 1, 2, \dots, M$. For any user with distance to the SBS ranging from d_{i-1} to d_i , we unicast the required content with energy $P_t(d_i)T_p$, which guarantees the minimum data rate r_0 for all the users in this area. And we set $l_0 = 0$ denoting that the required energy for unicast is zero. Then we have $Q_k \in \{0, 1, \dots, M\}$, where $Q_k = i$ corresponds to the case that $l_i E_{\text{unit}}$ amount of energy is required for unicasting the content.

With the discretization procedure, the state space \mathcal{S} is of dimension $(E_{\max} + 1) \times (M + 1) \times (N + 1)$.

2) *SBS Action*: The SBS has three actions to choose: sleep, unicast the required content, and push the most popular unpushed content. We define the action takes values in set $\mathcal{U} = \{0, 1, 2\}$ as

$$u_k = \begin{cases} 0, & \text{sleep} \\ 1, & \text{unicast the required content} \\ 2, & \text{push the most popular unpushed content} \end{cases} \quad (8)$$

Notice that in different states, the SBS may not be able to take all the three actions. A simple example is that if $E_k = 0$, the SBS can do nothing but sleep, i.e., $u_k = 0$. Hence, the action space is state-dependent, which can be expressed as $u_k \in \mathcal{U}_k(x_k)$. If $E_k \geq Q_k$, $1 \in \mathcal{U}_k(x_k)$, i.e., the energy for unicast can be satisfied. 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. Then we conclude that if $E_k \geq l_M$, $2 \in \mathcal{U}_k(x_k)$, i.e., the energy for push can be satisfied.

¹In this paper, the term "stage" is equivalent with the term "period".

3) *Cost Function*: The cost function depends on both the system state and the action, hence is 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. Mathematically, we can express it as

$$g_k(x_k, u_k) = \begin{cases} 1, & \text{if } Q_k > 0, u_k \neq 1 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

4) *State Transition*: The state transition is expressed as the conditional probability

$$\begin{aligned} & p_{x_k \rightarrow x_{k+1} | u_k} \\ &= \Pr(E_{k+1}, Q_{k+1}, C_{k+1} | E_k, Q_k, C_k, u_k) \\ &= \Pr(E_{k+1} | E_k, Q_k, u_k) \Pr(C_{k+1} | C_k, u_k) \Pr(Q_{k+1} | C_{k+1}), \end{aligned} \quad (10)$$

where the second equality is derived based on the law of total probability and the fact that for the given action u_k , E_{k+1} only depends on E_k and Q_k according to (3), and C_{k+1} only depends on C_k according to (4). While Q_{k+1} depends on C_{k+1} because C_{k+1} decides the probability with which a content has been pushed, hence influences the probability with which a unicast is required.

We calculate the state transition probability according to (10). Firstly, to calculate the energy state transition probability, we denote $p_a(i), i = 0, 1, \dots$ as the probability that $i E_{\text{unit}}$ 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 $p_a(i) = 0, \forall i = -1, -2, \dots$. Then we have

$$\Pr(E_{k+1} | E_k, Q_k, u_k) = \begin{cases} p_a(E_{k+1} - E_k), & \text{if } u_k = 0, E_{k+1} < E_{\max} \\ 1 - \sum_{i=0}^{E_{\max} - E_k - 1} p_a(i), & \text{if } u_k = 0, E_{k+1} = E_{\max} \\ p_a(E_{k+1} - E_k + l_{Q_k}), & \text{if } 0 < l_{Q_k} \leq E_k, u_k = 1, E_{k+1} < E_{\max} \\ 1 - \sum_{i=0}^{E_{\max} - E_k + l_{Q_k} - 1} p_a(i), & \text{if } 0 < l_{Q_k} \leq E_k, u_k = 1, E_{k+1} = E_{\max} \\ p_a(E_{k+1} - E_k + l_M), & \text{if } l_M \leq E_k, u_k = 2, E_{k+1} < E_{\max} \\ 1 - \sum_{i=0}^{E_{\max} - E_k + l_M - 1} p_a(i), & \text{if } l_M \leq E_k, u_k = 2, E_{k+1} = E_{\max} \end{cases} \quad (11)$$

Note that the action $u_k = 0$ can be taken in any states, while $u_k = 1$ can be taken under the condition that $0 < l_{Q_k} \leq E_k$, and $u_k = 2$ with condition $l_M \leq E_k$ and $C_k < N$. Also note that when $E_{k+1} = E_{\max}$, the energy arrival may exceed the battery capacity. So the probability is calculated by summarizing all the possible energy arrival conditions.

Secondly, as in each stage, at most one content is pushed to users, and also at most one content will be replaced by a new one, C_k can only transit to its neighboring values $C_{k+1}, C_k - 1$

or keeps constant. The pushed content state is updated as

$$\Pr(C_{k+1} | C_k, u_k) = \begin{cases} p_c \frac{C_k}{N}, & \text{if } u_k < 2, C_{k+1} = C_k - 1 \geq 0 \\ 1 - p_c \frac{C_k}{N}, & \text{if } u_k < 2, C_{k+1} = C_k \\ 1 - p_c \frac{C_k}{N}, & \text{if } u_k = 2, C_{k+1} = C_k + 1 \leq N \\ p_c \frac{C_k}{N}, & \text{if } u_k = 2, C_{k+1} = C_k < N \\ 0, & \text{else} \end{cases} \quad (12)$$

Note that when a pushed content is replaced by a new one, it is removed from users' cache. While when the replaced content is not pushed at all, there is no influence to C_k .

Finally, the user request state transition is

$$\Pr(Q_{k+1} | C_{k+1}) = \begin{cases} (1 - p_u) + p_u \sum_{i=1}^{C_{k+1}} f_i, & \text{if } Q_{k+1} = 0 \\ p_u (1 - \sum_{i=1}^{C_{k+1}} f_i) \frac{d_m^2 - d_{m-1}^2}{R^2}, & \text{if } Q_{k+1} = m > 0 \end{cases} \quad (13)$$

where f_i is calculated according to (1) and $d_0 = 0$. $Q_{k+1} = 0$ means that either there is no user request generated or the user request can be satisfied by caching, i.e., the required content has been pushed. Otherwise, as the users are assumed uniformly distributed in the cell, the request is generated with distance to 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.

A. MDP Problem Formulation and Optimization

Based on the above MDP-based system modeling, the original optimization problem (5) can be re-written as

$$\min \lim_{K \rightarrow +\infty} \frac{1}{K} \mathbf{E} \left[\sum_{k=0}^{K-1} g(x_k, u_k(x_k)) \right]. \quad (14)$$

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 $\{u_1, u_2, \dots\}$. It can be proved that for any two states, there is a stationary policy \mathbf{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 \mathbf{u}^* [15, Sec 4.2].

According to [15, Prop. 4.2.1], the optimal average cost λ^* together with some vector $\mathbf{h}^* = \{h^*(x) | x \in \mathcal{S}\}$ satisfies the Bellman's equation

$$\lambda^* + h^*(x) = \min_{u \in \mathcal{U}(x)} \left[g(x, u) + \sum_{y \in \mathcal{S}} p_{x \rightarrow y | u} h^*(y) \right]. \quad (15)$$

Further more, if $u^*(x)$ attains the minimum value of (15) for each x , the stationary policy \mathbf{u}^* is optimal. Based on the Bellman's equation, instead of the long term average cost minimization, we only need to deal with (15) which only relates with per-stage cost $g(x, u)$ and state transition $p_{x \rightarrow y | u}$. The policy iteration algorithm [15, Sec. 4.4] can effectively solve the problem, which will be detailed in the next subsection.

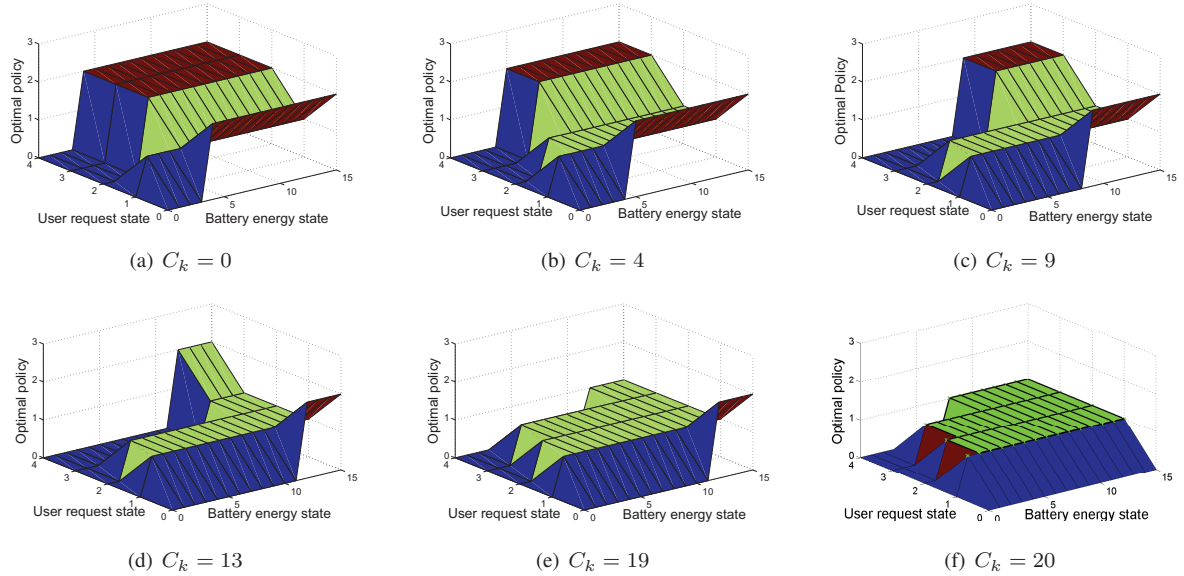


Fig. 3. Optimal policy sampled w.r.t. pushed content state with parameter $p_c = 0.3, p_u = 0.7, \bar{a} = 0.8$.

B. Policy Iteration Algorithm

The policy iteration algorithm 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 $\mathbf{u}^{(j)}$. Based on this policy, we perform *policy evaluation* [15, Sec. 4.4] step, i.e., we solve the following linear equations

$$\lambda^{(j)} + h^{(j)}(x) = g(x, u^{(j)}(x)) + \sum_{y \in \mathcal{S}} p_{x \rightarrow y | u^{(j)}(x)} h^{(j)}(y) \quad (16)$$

for $\forall x \in \mathcal{S}$ to get the average cost $\lambda^{(j)}$ and vector $\mathbf{h}^{(j)}$. Notice that there are $(E_{\max} + 1) \times (M + 1) \times (N + 1)$ equations but $(E_{\max} + 1) \times (M + 1) \times (N + 1) + 1$ unknown parameters, hence more than one solutions exist, which are different with each other by a constant value for all $h^{(j)}(x)$. Without loss of generality, we can set for example

$$h^{(j)}(E_{\max} + 1, M + 1, N + 1) = 0, \quad (17)$$

then the solution for (16) is unique.

As $\mathbf{u}^{(j)}$ may not be the optimal policy, we subsequently perform *policy improvement* [15, Sec. 4.4] step to find the policy $\mathbf{u}^{(j+1)}$ which minimizes the right hand side of Bellman's equation

$$u^{(j+1)}(x) = \arg \min_{u \in \mathcal{U}(x)} \left[g(x, u) + \sum_{y \in \mathcal{S}} p_{x \rightarrow y | u} h^{(j)}(y) \right]. \quad (18)$$

If $\mathbf{u}^{(j+1)} = \mathbf{u}^{(j)}$, the algorithm terminates, and the optimal policy is obtained $\mathbf{u}^* = \mathbf{u}^{(j)}$. Otherwise, repeat the procedure by replacing $\mathbf{u}^{(j)}$ with $\mathbf{u}^{(j+1)}$. It is proved that the policy iteration algorithm terminates in finite number of iterations [15, Prop. 4.4.1]. To sum up, the policy iteration algorithm is summarized in Algorithm 1.

Algorithm 1 Policy Iteration Algorithm

- 1: Set $u^{(1)}(x) = 0$ for all $x \in \mathcal{S}$.
 - 2: Set $j = 0$.
 - 3: **Do**
 - 4: Set $j = j + 1$.
 - 5: Calculate $\lambda^{(j)}$ and $\mathbf{h}^{(j)}$ by solving (16) and (17).
 - 6: Calculate $\mathbf{u}^{(j+1)}$ according to (18).
 - 7: **While** ($\mathbf{u}^{(j+1)} \neq \mathbf{u}^{(j)}$)
-

IV. NUMERICAL RESULTS

We run some simulations to study the structure of the optimal policy as well as evaluate its performance. We set the cell radius $R = 50\text{m}$, the required content delivery spectrum efficiency $r_0/W = 1\text{bps/Hz}$, the pathloss parameters $\beta = 10\text{dB}$ and $\alpha = 2$, $T_p = 1\text{s}$, $N = 20$, and the Zipf parameter $v = 0.5$. The battery capacity is discretized so that $E_{\max} = 15$, and we set $M = 4$, $P_t(R) = 1\text{Watt}$, E_{unit} and $\sigma^2 + I$ are set so that $l_M = M$ and (2) holds for $r = r_0, d = d_M, P_t = P_t(R)$, and then d_1, \dots, d_{M-1} are selected so that $l_i = i, i = 0, 1, \dots, M - 1$. Assume the energy arrival process follows a Poisson distribution with average arrival rate \bar{a} units of energy.

Fig. 3 shows the optimal policy structure with parameters $p_c = 0.3, p_u = 0.7, \bar{a} = 0.8$. Based on the results, we can have the following observations. Firstly, given user request state and pushed content state, the optimal policy w.r.t. battery energy state shows a threshold-based structure, i.e., the SBS will keep sleep until the battery energy exceeds some value, and then it will not sleep for any battery energy state larger than the value. It is because when the amount of battery energy is large, the SBS tends to greedily use it in case of battery overflow. Secondly, for the users close to the SBS (user request state 1), unicast is always preferred. As these users experience very good channel quality, unicast consumes very little energy, and hence is more beneficial than transferring the request to the MBS. Thirdly, the more contents are pushed, the less tendency

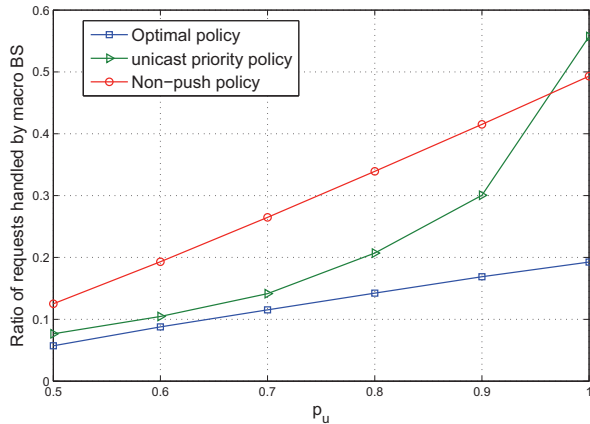


Fig. 4. The ratio of requests handled by the MBS with/without proactive push. $p_c = 0.3$, $\bar{a} = 0.8$.

the system decides to push. For $C_k = 0$, i.e., no contents are pushed, the SBS will push the popular contents for most states except that the users are close to the SBS. However, when the number of pushed contents approaches its maximum (e.g. $C_k = 19$), the SBS will push only when the system is idle ($Q_k = 0$) and the energy battery is almost full ($E_k \geq 12$).

Then we evaluate the performance gain obtained by push mechanism, which is illustrated in Fig. 4. Here, the unicast priority policy [14] is a simple greedy policy in which the SBS always satisfy the unicast request in the first place. The push action is taken only when there is no user request. While the non-push policy only takes actions including sleep and unicast, and it is also optimized using MDP approach, which follows the similar procedure of Sec. III by removing the push action. It is shown that compared with non-push optimal policy, the optimal push mechanism reduces the ratio of requests handled by the MBS by more than 50% and the gain increases as the traffic load increases (For the full buffer case where $p_u = 1$, the ratio is reduced by 60%). On the other hand, the unicast priority policy performs close to the optimal push policy at low traffic load regime, but performs even worse than the non-push policy when the traffic load is high. For the low traffic load case, it performs well since there is sufficient idle period for the system to push. While for the high traffic load case, very few contents can be pushed and the unicast priority policy converges to the non-push policy. For the full buffer case, it reduces to greedy non-push policy, and hence performs worse than the optimal non-push policy.

V. CONCLUSION

In this paper, proactive push in EH based SBS is optimized with MDP tools by properly discretizing the system energy and user request states. With policy iteration policy, the optimal policy is found and is shown by numerical results that it reveals a threshold based structure, i.e., the SBS sleeps until the battery energy exceeds some threshold. Then it keeps its unicast/push action for the rest battery energy states. In addition, compared with non-push policy, the push based optimal policy reduces the ratio of requests handled by MBS by more than 50%. It is shown that the push mechanism has great potential for performance enhancement. As this paper mainly focuses on

problem formulation and algorithm design for optimal policy, future work includes the analysis of the structure of the optimal policy. Also, integrating the non-ideal content fetch and caching in SBS is also a potential research direction.

ACKNOWLEDGMENT

This work is sponsored in part by the National Basic Research Program of China (No. 2012CB316001), and the Nature Science Foundation of China (No. 61201191 and 61401250), the Creative Research Groups of NSFC (No. 61321061), the Sino-Finnish Joint Research Program of NSFC (No. 61461136004), and Hitachi R&D Headquarter.

REFERENCES

- [1] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1732–1743, 2011.
- [2] 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.
- [3] 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.
- [4] 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 Transactions on Networking*, vol. 21, no. 1, pp. 110–120, Feb. 2013.
- [5] I. Podnar, M. Hauswirth, and M. Jazayeri, "Mobile push: delivering content to mobile users," in *Proceedings. 22nd International Conference on Distributed Computing Systems Workshops*, 2002, pp. 563–568.
- [6] X. Zhou, R. Zhang, and C. K. Ho, "Wireless information and power transfer: Architecture design and rate-energy tradeoff," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4754–4767, November 2013.
- [7] 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.
- [8] K. Shanmugam, N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching: wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
- [9] 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 (INFOCOM WKSHPS)*, Apr. 2011, pp. 259–264.
- [10] 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.
- [11] [Online]. Available: <http://www.hiwifi.com/j2>
- [12] K. Wang, Z. Chen, and H. Liu, "Push-based wireless converged networks for massive multimedia content delivery," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2894–2905, May 2014.
- [13] N. Sharma, D. Krishnappa, D. Irwin, M. Zink, and P. Shenoy, "Green-cache: augmenting off-the-grid cellular towers with multimedia caches," in *Proc. ACM MMsys13*, Feb. 2013.
- [14] 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*, to appear.
- [15] D. P. Bertsekas, *Dynamic programming and optimal control, Volume II*, 3rd edition. Athena Scientific Belmont, MA, 2005.
- [16] 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 *Proceedings of the 7th ACM SIGCOMM Conference on Internet measurement*. ACM, 2007, pp. 1–14.