Dynamic Content Updates in Heterogeneous Wireless Networks

Mehdi Salehi Heydar Abad [1], Emre Ozfatura[2], Ozgur Ercetin[1], and Deniz Gündüz[2]

[1]Faculty of Engineering and Natural Sciences, Sabanci University
[2]Department of Electrical and Electronic Engineering Imperial College London
[1]{*mehdis,oercetin*}*@sabanciuniv.edu*
[2]{*m.ozfatura,d.gunduz*}*@imperial.ac.uk*

***Abstract*—Content storage at the network edge is a promising solution to mitigate excessive content traffic. To this end, cache-enabled cellular architectures can be utilized to reduce the network cost. However, content storage strategies should be able to take into account user satisfaction, which depends on the age of a dynamic content. The ratio of satisfied users can be increased with more frequent cache refreshment, albeit at an increased network cost. In this paper, we introduce a cache refreshment strategy that strikes a balance between user satisfaction and the infrastructure cost.**

***Index Terms*—Content caching, content refreshment, Markov decision process, multi-armed bandit**

I. Introduction

While proactive content caching has received significant interest in recent years, most existing strategies in the literature (both with uncoded [1] and coded placement [2]) are based on known content popularities. Although it is possible to observe the global popularity of contents in on-demand video streaming services, such as YouTube, small-cell base stations (SBSs) usually serve a small geographical area, where the local content popularity might not be aligned with the global popularity [3]. This mismatch between the local and global content popularities requires the design of predictive caching policies that aim to learn the local content popularity from user requests. Predictive caching policies can be classified into two groups, namely *predictive caching with unknown content popularity* [4] and *predictive caching with time-varying content popularity* [5]. In [4], the authors focus on a single SBS and model the predictive caching problem as a multi-armed bandit (MAB) problem, in which the received user requests are utilized to predict the content popularities, and the optimal caching strategy is obtained by taking into account the cost of content replacements. In [6], this approach has been extended to a cooperative caching framework, where the SBSs fetch a requested content from neighboring SBSs if it is cached there. In [5], a cache replacement strategy has been introduced for time-varying popularity scenario to maximize the local service rate with a minimum replacement cost, while a more theoretical approach is taken in [7], which studies the cache update policy in the case of time-varying content popularities.

This work was in part supported by EC H2020-MSCA-RISE-2015 programme under grant number 690893.

Although the predictive caching framework is highly effective in increasing the efficiency of edge caching, it has certain limitations. Most of the aforementioned strategies are designed to predict only the content popularity. However, in many applications, e.g., news, weather, etc., freshness of the content is an important factor for user satisfaction. Content caching and refreshment problem has been previously studied in [8, 9]. In this paper, we consider a heterogeneous cellular network with cache-enabled SBSs, and propose a cost-aware content update policy. We first show that the structure of the optimal periodic content update policy that minimizes the network cost for given users' tolerance to the age of the content is of threshold type. Then, we formulate the resultant cache-refreshment problem as a MAB problem, where we learn users' tolerance to the age of the content. Accordingly, we design an online cache refreshment policy to minimize the overall network cost. To the best of our knowledge, this is the first work that utilizes a reinforcement learning framework to analyze user behavior based on content age.

II. System model and problem formulation

We consider a cellular network with a macro base station (MBS) and a SBS serving the users in the cell. We assume that both the MBS and the SBS are equipped with cache memories, storing a library of N distinct dynamic contents, denoted by $S_1,\dots,S_N$. Each dynamic content has a different popularity, i.e., the probability that a user requests content S_n is p_n.

Dynamic contents, such as news videos, traffic and weather updates may change frequently over time. We assume that the MBS, thanks to its relatively high-bandwidth connection to the content server in the core network, always has the most up-to-date contents, whereas the stored contents at the SBS are less frequently refreshed to reduce the load on its limited-bandwidth backhaul connection. We consider a discrete time system model with equal-length time slots. At the beginning of each time slot the SBS decides on which contents to be updated. The decision vector at time t is denoted by $\mathbf{d}(t) = (d_1,\dots,d_N)$, where we set $d_n(t) = 1$ if content S_n is updated at the end of time slot t, and $d_n(t) = 0$ otherwise. We denote by $h_n(t)$ the *age* of content S_n in the SBS cache at time slot t. We assume a maximum age T_{max} at which a content becomes obsolete. In other words, the age of a content increases until

ISBN 978-3-903176-13-3

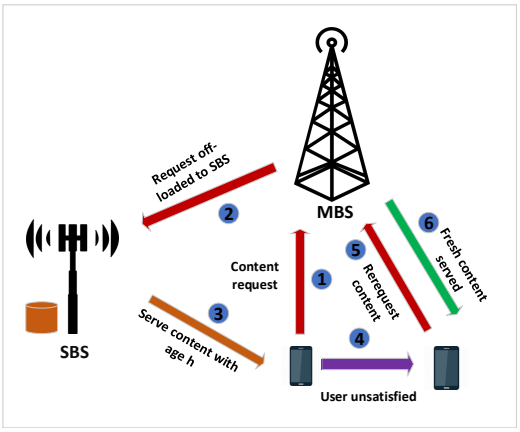

Figure 1: User requests are first off-loaded to the SBS; however, users who are unsatisfied with the freshness of the content served by the SBS are re-directed to the MBS.

it becomes obsolete. Accordingly, the age of content S_n, $n = 1,\ldots,N$, evolves over time as follows:

$$h_n(t+1) = \max\{(1-d_n(t))(h_n(t)+1), T_{max}\}. \quad (1)$$

We denote the length-N vector of ages associated with all the dynamic contents in the library by $\mathbf{h}(t)$.

A. User behavior

Let $\lambda(t)$ denote the number of users that request a content at time slot t. Whenever a user requests a content, the request is first off-loaded to the SBS. Users have different tolerance levels to the age of the contents they receive. Hence, we consider that, with probability $P^{(n)}_{redirect}(h)$, a user is not satisfied with the age h of content S_n; and thus, it places another request for S_n, which is served directly by the MBS with a fresh content. Let $\lambda_n(t)$ be the number of users that request content n, $n = 1,\ldots,N$, in time slot t, i.e., $\sum_{n=1}^{N}\lambda_n(t) = \lambda(t)$, which is governed by the popularity profile p_n. The users that request content n are split into two disjoint sets, where the first set consists of $\lambda_{rn}(t)$ users redirected to the MBS, while the second set consists of $\lambda_{an}(t)$ users satisfied with the content served by the SBS. Note that $\lambda_{rn}(t)$ and $\lambda_{an}(t)$ are governed by the random process $P^{(n)}_{redirect}(h_n(t))$. Let $\boldsymbol{\lambda}_r(t) = (\lambda_{r1}(t),\ldots,\lambda_{rN}(t))$ be the vector associated with the number of redirected users for each content. Expected values of parameters λ_n and λ_{rn} are given by:

$$\mathbb{E}[\lambda_n|\lambda] = \lambda p_n, \text{ and } \mathbb{E}[\lambda_{rn}|\lambda,h_n] = \lambda p_n P_r^{(n)}(h_n). \quad (2)$$

B. Decision model and the problem formulation

Let $C(\boldsymbol{\lambda}_r(t),\mathbf{d}(t))$ be the cost associated with serving the users redirected to the MBS at time t, and the backhaul cost associated with updating the contents, if there is any. In this work, we assume that this cost is linear in λ_{rn}[1]. Hence,

$$C(\boldsymbol{\lambda}_r(t),\mathbf{d}(t)) = \sum_{n=1}^{N} C_n(\lambda_{rn}(t),d_n(t)). \quad (3)$$

Similarly, we define the back-haul cost function $C_{BH}(\mathbf{d}(t))$ which is also a linear function.

$$C_{BH}(\mathbf{d}(t)) = \sum_{n=1}^{N} C^{(n)}_{BH}(d_n(t)), \quad (4)$$

where $C^{(n)}_{BH}(d_n(t)) = d_n(t)\mathscr{E}_n$, with $\mathscr{E}_n$ being the average back-haul cost of updating content n. If the SBS decides to update content n (or multiple contents), the age of the content is updated at the end of the time slot. Hence,

$$C_n(\lambda_{rn}(t),d_n(t)) = \beta_n + \alpha_n\lambda_{rn}(t) + d_n(t)\mathscr{E}_n. \quad (5)$$

Note that updating a content has an immediate cost which is larger than not-updating. However, the incurred extra cost in updating the content enables more users to be served by the SBS. We aim at minimizing the average total cost as follows:

$$\min_{\mathbf{d}(t)\in\{0,1\}^N} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[C(\boldsymbol{\lambda}_r(t),\mathbf{d}(t))\right]. \quad (6)$$

III. MDP FORMULATION

Define the state of the system to be $\mathbf{h}$. We denote by $V(\mathbf{h})$ the differential value function at state $\mathbf{h}$. The differential Bellman equations can be written as:

$$V(\mathbf{h}) + \mu^* = \min_{\mathbf{d}} V_{\mathbf{d}}(\mathbf{h}), \quad (7)$$

where μ^* is the optimal average cost and $V_{\mathbf{d}}(\mathbf{h})$ is the differential action-value function defined by:

$$V_{\mathbf{d}}(\mathbf{h}) \triangleq \bar{C}(\mathbf{h},\mathbf{d}) + \mathbb{P}(\acute{\mathbf{h}}|\mathbf{h},\mathbf{d})V(\acute{\mathbf{h}}), \quad (8)$$

where $\mathbb{P}(\acute{\mathbf{h}}|\mathbf{h},\mathbf{d})$ is the transition probability from state $\mathbf{h}$ into $\acute{\mathbf{h}}$ when action $\mathbf{d}$ is taken which is governed by (1), and

$$\bar{C}(\mathbf{h},\mathbf{d}) = \sum_{\boldsymbol{\Lambda}} \mathbb{P}(\boldsymbol{\lambda}_r = \boldsymbol{\Lambda})C(\boldsymbol{\lambda}_r,\mathbf{d}). \quad (9)$$

The MDP associated with the average cost minimization problem can be solved by the well known value iteration algorithm. However, the cardinality of the state (i.e., $(T_{max})^N$) and action spaces (i.e., 2^N) grow exponentially with the number of contents. Hence, the curse of dimensionality is the bottleneck for an efficient solution. To bypass this bottleneck, we note that the cost function in (3) is linear and the transition probabilities of a content does not affect the others. Hence, we can separate the value function in (7) into N independent value functions each representing a distinct content. For each content n, we have

$$V^{(n)}(h_n) + \mu_n^* = \min_{d_n} V_d^{(n)}(h_n) \quad (10)$$

[1]For example, in OFDMA, a user re-directed to the MBS is assigned a subcarrier, and the power allocated to that subcarrier adds linearly to the energy cost.

$$V_d^{(n)}(h_n) = \bar{C}_n(h_n, d_n) + d_n V^{(n)}(0) + (1-d_n)V^{(n)}(h_n+1). \tag{11}$$

We have developed a framework that has enabled distributed policies with respect to individual contents. We will show that for each content, there exists a threshold policy on the age of the content for which it is optimal to update the content. The following lemma establishes the key property used to prove the structure of the optimal policy.

Lemma 1. *The differential value function $V^{(n)}(h_n)$ for all $n = 1,\dots,N$ is non-decreasing with respect to the age of the content, h_n.*

Proof. We use the value iteration algorithm to prove the lemma. We start by an arbitrary $V_0^{(n)}(h)$ differential value function and obtain the k-step differential value function $V_k^{(n)}(h)$ as follows:

$$V_{k+1}^{(n)}(h) = \min_d(-\mu_n^* + \bar{C}_n(h_n, d_n) + d_n V_k^{(n)}(0) + (1-d_n)V_k^{(n)}(h+1)) \tag{12}$$

Note that $\lim_{k\to\infty} V_k^{(n)}(h) = V^{(n)}(h)$. The proof is by induction. For $k=1$, $V_1^{(n)}(h) = \min_{d_n}(-\mu_n^* + \bar{C}_n(h_n,d_n) + d_n V_0^{(n)}(0) + (1-d_n)V_0^{(n)}(h_n+1))$, which is the minimum of two non-decreasing functions, and thus, itself is a non-decreasing function in h_n. Assume that the lemma holds for k. Then, according to (12), $V_{k+1}^{(n)}(h_n)$ is also a non-decreasing function with respect to h_n. By letting $k\to\infty$, we conclude the proof by showing that $V^{(n)}(h_n)$ is also non-decreasing in h_n. □

Theorem 1. *For each content the optimal policy minimizing the average cost is a threshold policy.*

Proof. The monotonicity of the differential value functions prove the optimality of the threshold policy [10, Chapter 7]. Intuitively, due to the non-decreasing property of the $V^{(n)}(h_n)$, at some age, it would be optimal to update the content. Since the differential value function is non-decreasing, a larger, or smaller age would not be able to yield a smaller average cost. □

IV. Learning Content Popularity and Age Tolerance

In the previous section, we showed that the problem is separable and thus, the optimization can be performed for each content separately. Second, we proved that the policy minimizing the cost is a threshold policy. Hence, the SBS by monitoring the age of the contents individually, needs to optimize according to a single threshold for each content. Under the threshold policy, the age of a content increases linearly until it reaches the threshold wherein the content will be updated and the age will refresh to a value of zero. Thus the minimum cost associated with content n is the solution of:

$$\mu_n^* = \min_{H_n} \frac{1}{H_n+1}\left(\sum_{h=0}^{H_n} \bar{C}_n(h,0) + \mathscr{E}_n\right), \tag{13}$$

where

$$\bar{C}_n(h,0) = \beta_n + \alpha_n \lambda\, p_n P_{redirect}^{(n)}(h). \tag{14}$$

Considering the linearity of the cost functions, the average cost optimization becomes:

$$\min_{H_n}\left\{\beta_n + \frac{\mathscr{E}_n + \alpha_n p_n \sum_{h=0}^{H_n} P_{redirect}^{(n)}(h)}{H_n+1}\right\}. \tag{15}$$

The equivalent optimization problem depends on the redirection probabilities $P_{redirect}^{(n)}(h)$, that are unknown. Hence, in the following we resort to reinforcement learning methods to infer the redirection probabilities.

We consider a sequential learning framework in which the SBS at each iteration of the learning algorithm faces choosing a threshold $0 \le H_n \le T_{max}$. After choosing the threshold, the SBS will observe a random cost associated with its decision;

$$\hat{C}_n(H_n) = \frac{\mathscr{E}_n + \sum_{t=1}^{H_n} C_n(\lambda_{rn}(t), 0)}{H_n+1} \tag{16}$$

The learning algorithm should provide the SBS a method to adjust its strategy by observing the outcomes of its decisions. This can be formulated as MAB problem, where each action (i.e., thresholds) has an expected return value, corresponding to the *value* of that action. We denote the true value of action H_n by $q_n(H_n) = \frac{1}{H_n+1}\left(\sum_{h_n=0}^{H} \bar{C}_n(h,0) + \mathscr{E}_n\right)$. If the SBS knows the $q(\cdot)$ values, it can simply choose the action with the minimum expected cost. A well-studied algorithm for learning those values is the ε-greedy algorithm [11], which starts by an arbitrary estimate $Q_n(\cdot)$ of the value of the actions, and interacts with the environment to update its initial estimates, eventually converging to the true estimates. Two critical aspects of the ε-greedy algorithm are the exploitation and exploration stages. The agent utilizing the estimates greedily chooses an action, and thus, it exploits what it knows already. Meanwhile, if it chooses an action completely random regardless of the estimates we say that it explores. Exploitation is necessary to act upon the experience while exploration helps to improve the estimate values and it facilitates convergence to the true action values. The ε-greedy algorithm is presented in Algorithm 1.

Algorithm 1 ε-greedy

1: **for** $i = 1,2,\dots$ **do**
2: $H_n \leftarrow \begin{cases} \arg\max_{H_n} Q_n(H_n) & \text{with probability } 1-\varepsilon, \\ \text{random action} & \text{with probability } \varepsilon. \end{cases}$
3: Apply H_n and observe $\hat{C}_n(H_n)$
4: $Q_n(H_n) \leftarrow (1-\zeta)Q_n(H_n) + \zeta \hat{C}_n(H_n)$

V. Numerical Results

In this section, we aim at evaluating the performance of the ε-greedy algorithm in finding the optimal thresholds that minimize the total cost of the system. Due to the separability, we consider only one content and we note that the learning processes for all the contents are the same. The popularity

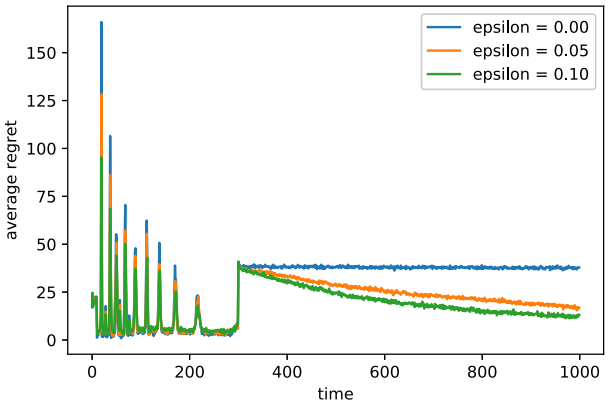

Figure 2: Average regret of the ε-greedy algorithm for $\varepsilon = 0, 0.05, 0.1$ in a non-stationary environment.

of the contents are modeled by a Zipf distribution with an exponent of 1.1. We assume that on average a given user becomes dissatisfied with a content of age h with probability of $e^{-0.4h}$. Users arrive at the system according to a Poisson distribution with rate 100 users per time slot. The cost of re-directed users to MBS is $C_n(\lambda_{rn}(t)) = 10 \cdot \lambda_{rn}(t)$ and the backhaul cost is assumed to be 500 initially and then at time $t = 300$ the backhaul cost decreases to a value of 400 to simulate a time-varying scenario. We illustrate the performance of the ε-greedy algorithm for $\varepsilon = 0,\ 0.05,\ 0.1$ by adopting average regret as the metric. The regret of a learning algorithm is defined to be the difference between the cost achieved by the learner and the optimal cost. Here, we obtain the optimal cost by assuming that $P_{redirect}(h)$ is known, and by numerically solving (13). Note that the estimates of the action values, $Q(H)$, is initialized to be 0 for all $H = 0, \ldots, T_{max}$. Note also that, the greedy algorithm's (i.e., $\varepsilon = 0$) regret converges to zero initially ($t < 300$) even if it always exploits. This is not too surprising considering that the action-values are initialized opportunistically (i.e., the initial costs are believed to be zero by the agent). At the beginning, i.e., $t = 0$, the greedy algorithm believes that every action returns a value of zero. However, by trying each action it gets disappointed in that action and tries the rest. In other words the estimates are biased. Opportunistic initialization is a simple method to incentivize exploration. However, it can only happen once and at the beginning of time. This method quickly fails in non-stationary environments.

We can see that the greedy algorithm cannot adapt to the non-stationary ($t > 300$) environment and it gets stuck in a sub-optimal threshold. Meanwhile, for 0.05 and 0.1-greedy algorithm, it is able to adapt to the environment thanks to their exploration strategy. A large value of ε results in more exploration, and thus, we can see that 0.1-greedy algorithm has a faster decay in terms of the average regret compared to the 0.05-greedy algorithm. However, note that ε-greedy algorithm is expected to be at least ε away from the optimal cost. Thus, there is a trade-off between the rate of convergence and the value of convergence.

VI. Conclusion

We have considered a cost minimization problem in a dynamic content caching setting, and seeked a balance between the number of unsatisfied users redirected to the MBS, and the cost of accessing the backhaul link by the SBS to refresh its dynamic contents. We have formulated the cost minimization problem as an MDP, and proved that a threshold policy in the age of the contents is optimal. Subsequently, to identify the optimal thresholds, we resorted to learning algorithms since users' preferences are not known and vary over contents. To that extend, we represented the problem in MAB framework and through numerical results, we showed that it is possible to make the expected regret of the learning algorithm arbitrarily close to zero. As a future work, we extend the model by considering non-linear cost functions and the heterogeneous cellular network architecture with energy harvesting SBSs.

References

[1] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation Algorithms for Mobile Data Caching in Small Cell Networks," *IEEE Trans. Comm.*, Oct 2014.

[2] K. Shanmugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec 2013.

[3] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE J. Sel. Areas in Commun.*, vol. 35, no. 5, pp. 1076–1089, May 2017.

[4] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," in *IEEE Int. Conf. Commun. (ICC)*, June 2014, pp. 1897–1903.

[5] A. Sadeghi, F. Sheikholeslami, and G. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Topics Signal Proc.*, vol. 12, no. 1, pp. 180–190, Feb 2018.

[6] J. Song, M. Sheng, T. Quek, C. Xu, and X. Wang, "Learning-based content caching and sharing for wireless networks," *IEEE Trans. Comm.*, Oct 2017.

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

[8] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Information freshness and popularity in mobile caching," in *IEEE ISIT*, June 2017.

[9] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger, "Age-optimal constrained cache updating," in *IEEE ISIT*, June 2017, pp. 141–145.

[10] D. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific, 1995.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.