Poster: Wireless Caching in Large-Scale Edge Access Points: A Local Distributed Approach

Ge Ma, Zhi Wang, Jiahui Ye, Wenwu Zhu
Tsinghua University, Beijing, China
{mg15,yejh16,wwzhu}@mails.tsinghua.edu.cn;wangzhi@sz.tsinghua.edu.cn

ABSTRACT

Today's mobile users achieve unsatisfactory quality of experience mainly due to the large network distance to the centralized infrastructure. To improve users' experiences, caching at the wireless access points (APs) has been proposed for bringing the contents closer to users. However, the wireless content placement is challenging as the placement is affected by many realistic constraints, such as a large number of APs, interaction among neighboring APs, various local content popularities. In this paper, we study the wireless caching problem, i.e., which contents should be stored by which APs. First, we fulfil these constraints to formulate our problem and introduce an objective function that maximizes the total cache hit rate of all APs. Next, we prove the NP-hardness of the problem and propose a local distributed caching algorithm to address it. Furthermore, we provide a game theoretic perspective on the problem and prove that the proposed algorithm can converge to the Nash Equilibrium in polynomial time. Finally, we perform simulations on a real-world dataset to demonstrate the effectiveness of our algorithm.

KEYWORDS

Wireless edge networks; collaborative caching; potential game

1 INTRODUCTION

Recently, wireless content delivery has dominated the Internet traffic. Wireless traffic accounts for 52 percent of the total Internet traffic, and the ratio is expected to grow to 67 percent by the end of 2021 [4]. This expected increase motivates changes to the operations of wireless networks, as the current infrastructure cannot cope with this increase. One of the most promising ways to handle the above challenge is to introduce caching at the wireless access points (e.g., Wi-Fi or base stations), essentially bringing the content closer to users. The idea is to cache the most popular contents at the wireless edge and use the backhaul to update the cached contents, which can (i) reduce the data traffic going through the backhaul, (ii) reduce the latency for content delivery, and (iii) help in smoothing the traffic during peak hours.

Caching in wireless access points (APs) has been extensively studied under different settings and for different objectives [6, 7]. The authors in [9] propose a heuristic algorithm to minimize the content access latency of all users, while [5] develops an offline caching algorithm to maximize the profit of the mobile network operator (MNO). However, existing studies focus on either summarizing commonalities of content popularity across different APs or solving an optimization problem under offline settings, while ignoring the realistic constraints faced by the MNO:

- **Millions of APs.** In the real world, the MNO has deployed a considerable number of APs at the wirelss edge, even there are millions of APs in one city [8]. It is inefficient and impractical to control millions of APs through a centralized caching algorithm in an offline manner.
- **Interaction among neighboring APs.** In wireless networks, it is common that each area is covered by multiple APs and each user has connection to multiple APs [2]. So which content one AP should be cached not only depends on the content popularity, but also on the content stored by its neighboring APs.
- **Various local content popularities.** Content popularities in different APs (even adjacent APs) can be significantly different from each other due to the influence of small population [7], so that the caching policy becomes non-trivial.

To incorporate these realistic constraints, in this paper, we propose a local distributed caching algorithm that requires communication only between APs with overlapping coverage areas. In the basic algorithm, each AP will selfishly update its stored content by maximizing the local cache hit rate and by considering the content stored by neighboring APs. More specially, based on the game theory, we model the wireless caching problem as a potential game and prove that our algorithm can converge to the Nash Equilibrium in polynomial time. Finally, our simulation results show that our algorithm can approximate to the global optimum.

2 SYSTEM MODEL AND PROBLEM FORMULATION

In our system, we make the following assumption for the formulation convenience. (i) The popularity distribution of the contents changes slowly [2], and it can be learned through some popularity prediction algorithm [7]. (ii) Each content has an identical size 1 (if not, the original content can be divided into fixed-size segments). (iii) We assume homogeneous cache capacity for all the APs.

2.1 Network Model

We consider a general network of $N \times N$[1] APs that are uniformly located in a rectangular plane $\mathbb{R}^2$. To simplify the analysis, we partition the original plane into $N \times N$ grids with the same size

[1]Note that the variables in this paper are integer variables, unless otherwise specified.

MobiCom'18, October 29-November 2, 2018, New Delhi, India.

ACM ISBN 978-1-4503-5903-0/18/10.
DOI: http://dx.doi.org/10.1145/3241539.3267741

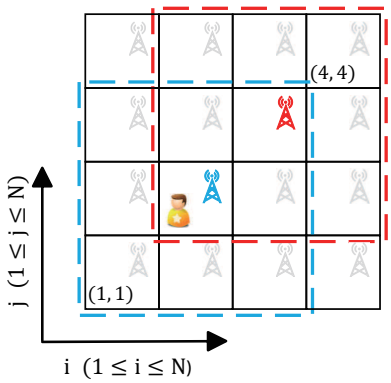

Figure 1: Example of a grid network indicating how grids are covered by APs ($N = 4, M = 3$, and dashed lines represent the coverage areas of APs).

and each grid has only one AP[2]. Each AP is covering a certain area of the plane and all APs have the same rectangular coverage area with $M \times M$ grids ($M \leq N$). Let $\mathcal{H} = \{h_{i,j} : 1 \leq i, j \leq N\}$ denote the total APs in the $N \times N$ grids, where $h_{i,j}$ is the AP in the grid (i, j). The set of grids covered by AP $h_{i,j}$ is denoted as $\mathcal{A}_{i,j}$, namely $\mathcal{A}_{i,j} = \{(m, n) : i - 1 \leq m \leq i + 1, j - 1 \leq n \leq j + 1\}$. A simple example for a 4×4 grid network is illustrated in Fig. 1. In this example, the AP $h_{2,2}$ is covering 9 grids and $\mathcal{A}_{2,2} = \{(i, j) : 1 \leq i, j \leq 3\}$. Naturally, each grid is covered by multiple APs, so that the coverage areas of neighboring APs are overlapped. There are 4 common grids covered by both AP $h_{2,2}$ and $h_{3,3}$, i.e., $\mathcal{A}_{2,2} \cap \mathcal{A}_{3,3} = \{(2, 2), (2, 3), (3, 2), (3, 3)\}$.

Each AP is equipped with a cache that can be used to store contents from a set $\mathcal{V} = \{1, 2, \cdots, V\}$ of V contents. The cache capacity of each AP is C ($C < V$), meaning that each AP can store C contents. However, our work can be extended to the network where APs have different capacities.

We consider a heterogeneous content popularity model, where the local content popularities are different across different APs. Let $\boldsymbol{p}_{i,j} = \{p^v_{i,j} : \forall v \in \mathcal{V}, 1 \leq i, j \leq N\}$ denote the content popularity in grid (i, j), where $p^v_{i,j}$ is the probability that content v is requested in grid (i, j).

2.2 Problem Definition

In our wireless caching problem, our goal is to devise the caching policy that maximizes the total cache hit rate of all APs. We denote the caching policy for AP $h_{i,j}$ as a V-tuple, $\boldsymbol{x}_{i,j} = \{x^v_{i,j} : \forall v \in \mathcal{V}\}$, where $x^v_{i,j}$ indicates whether content v is stored in AP $h_{i,j}$ ($x^v_{i,j} = 1$) or not ($x^v_{i,j} = 0$). The overall caching policy for the network is denoted by $\boldsymbol{X} = \{\boldsymbol{x}_{i,j} : 1 \leq i, j \leq N\}$ as an $NV \times N$ matrix. Thus, our caching problem can be formulated as the following optimization function:

$$\max_{\{\mathrm{X}\}} \quad f(\mathrm{X}) = 1 - \frac{1}{N^2} \sum_{(i,j)\in\mathcal{H}} \sum_{v\in\mathcal{V}} p^v_{i,j} \prod_{(m,n)\in\mathcal{A}_{i,j}} (1 - x^v_{m,n}), \qquad \text{(U)}$$

[2]Note that other distributions of APs' locations (e.g., random distribution) can be formulated using different partition granularities.

subject to

$$\sum_{v\in\mathcal{V}} x^v_{i,j} \leq C, \quad \forall i, j. \qquad (1)$$

Clearly, the problem (U) is very hard to solve optimally. [2] proves that the wireless collaborative caching problem which only considers two specific contents can be formulated as the 2-Disjoint Set Cover Problem, which is an NP-hard problem. Thus, our problem considering V contents is also NP-hard.

3 ONLINE ALGORITHM

We will provide an online distributed algorithm to address the above problem in which we iteratively update the caching policy in each AP. Let $\mathcal{A}_{[-(i,j)]} = \mathcal{A}_{i,j} \backslash \{i, j\}$ denote the grids covered by AP $h_{i,j}$ except grid (i, j) and $\boldsymbol{X}_{[-(i,j)]} = \{\boldsymbol{x}_{m,n} : (m, n) \in \mathcal{H}\}$ denote the caching policies of APs except AP $h_{i,j}$. Given an AP $h_{i,j}$, its cache hit rate $f_{i,j}$ (i.e., *utility function*) is denoted by,

$$f_{i,j}(\boldsymbol{x}_{i,j}|\boldsymbol{X}_{[-(i,j)]}) = 1 - \frac{1}{M^2} \sum_{v\in\mathcal{V}} \sum_{\substack{(m,n)\\ \in\mathcal{A}_{i,j}}} p^v_{m,n}(1 - x^v_{i,j}) \prod_{\substack{(k,l)\in\\ \mathcal{A}_{[-(i,j)]}\cap\mathcal{A}_{m,n}}} (1 - x^v_{k,l}). \qquad (2)$$

The basic idea of our algorithm is that AP $h_{i,j}$ tries selfishly to maximize the utility function $f_{i,j}(\boldsymbol{x}_{i,j}|\boldsymbol{X}_{[-(i,j)]})$, according to the caching policies of neighboring APs in the grids $\mathcal{A}_{[-(i,j)]}$. This procedure can be regarded as a non-cooperative game: each player (i.e., AP) continues optimizing its policy until no further improvements can be made (i.e., no player has an incentive to change unilaterally his own policy). At this point, $\boldsymbol{X}$ is a *Nash Equilibrium* policy, satisfying

$$f_{i,j}(\boldsymbol{x}_{i,j}|\boldsymbol{X}_{[-(i,j)]}) \geq f_{i,j}(\tilde{\boldsymbol{x}}_{i,j}|\boldsymbol{X}_{[-(i,j)]}), \forall i, j, \tilde{\boldsymbol{x}}_{i,j}.$$

Theorem 1. *The caching problem defined in problem (U) is a potential game with the potential function $f(\boldsymbol{X})$.*

Proof. To prove the game is potential, we need to verify that the change in the potential function is proportional to the change in the utility function of each AP, i.e.,

$$f(\tilde{\boldsymbol{x}}_{i,j}, \boldsymbol{X}_{[-(i,j)]}) - f(\boldsymbol{x}_{i,j}, \boldsymbol{X}_{[-(i,j)]}) = \beta[f_{i,j}(\tilde{\boldsymbol{x}}_{i,j}|\boldsymbol{X}_{[-(i,j)]}) - f_{i,j}(\boldsymbol{x}_{i,j}|\boldsymbol{X}_{[-(i,j)]})]. \qquad (3)$$

First, we have

$$f_{i,j}(\tilde{\boldsymbol{x}}_{i,j}|\boldsymbol{X}_{[-(i,j)]}) - f_{i,j}(\boldsymbol{x}_{i,j}|\boldsymbol{X}_{[-(i,j)]}) = \frac{1}{M^2} \sum_{v\in\mathcal{V}} \sum_{\substack{(m,n)\\ \in\mathcal{A}_{i,j}}} p^v_{m,n}(\tilde{x}^v_{i,j} - x^v_{i,j}) \prod_{\substack{(k,l)\in\\ \mathcal{A}_{[-(i,j)]}\cap\mathcal{A}_{m,n}}} (1 - x^v_{k,l}).$$

Since

$$f(\tilde{\boldsymbol{x}}_{i,j}, \boldsymbol{X}_{[-(i,j)]}) = 1 - \frac{1}{N^2}[\sum_{\substack{(m,n)\\ \in\mathcal{H}\backslash\mathcal{A}_{i,j}}} \sum_{v\in\mathcal{V}} p^v_{m,n} \prod_{\substack{(k,l)\in\\ \mathcal{A}_{m,n}}} (1 - x^v_{k,l}) + \sum_{\substack{(m,n)\\ \in\mathcal{A}_{i,j}}} \sum_{v\in\mathcal{V}} p^v_{m,n}(1 - \tilde{x}^v_{i,j}) \prod_{\substack{(k,l)\in\\ \mathcal{A}_{[-(i,j)]}\cap\mathcal{A}_{m,n}}} (1 - x^v_{k,l})],$$

we have

$$f(\tilde{\boldsymbol{x}}_{i,j}, \boldsymbol{X}_{[-(i,j)]}) - f(\boldsymbol{x}_{i,j}, \boldsymbol{X}_{[-(i,j)]}) = \frac{1}{N^2} \sum_{\substack{(m,n)\\ \in\mathcal{A}_{i,j}}} \sum_{v\in\mathcal{V}} p^v_{m,n}(\tilde{x}^v_{i,j} - x^v_{i,j}) \prod_{\substack{(k,l)\in\\ \mathcal{A}_{[-(i,j)]}\cap\mathcal{A}_{m,n}}} (1 - x^v_{k,l}) = \frac{M^2}{N^2}[f_{i,j}(\tilde{\boldsymbol{x}}_{i,j}|\boldsymbol{X}_{[-(i,j)]}) - f_{i,j}(\boldsymbol{x}_{i,j}|\boldsymbol{X}_{[-(i,j)]})].$$

□

The details of our proposed algorithm are presented in Algorithm 1. It adopts the value iteration technique, which utilizes the utility function defined in Eq. (2), to iteratively compute the expected maximal accumulated cache hit rate. As such, a Nash Equilibrium policy can be derived. In particular, the algorithm starts with an empty policies (line 1). At each iteration, a random AP is chosen uniformly from the total AP set $\mathcal{H}$ and updates its policy storing C contents with the highest marginal values. The algorithm stops when $\boldsymbol{X}$ converges to the Nash Equilibrium (line 2 – line 11).

Algorithm 1: Local Distributed Algorithm (LDA)

1 $\boldsymbol{x}_{i,j} \leftarrow \mathbf{0}, \theta_{i,j} \leftarrow 0, \forall i,j$
2 **while** $\sum_{(i,j)\in\mathcal{H}} \theta_{i,j} \neq N^2$ **do**
3 $(i,j) \leftarrow \text{Uniform}(N \times N), \tilde{\boldsymbol{x}}_{i,j} \leftarrow \mathbf{0}, \theta_{i,j} \leftarrow 1$
4 **for** $t = 1$ to C **do**
5 $v^* \leftarrow \arg\max_v \{f_{i,j}(\tilde{x}^v_{i,j} = 1|\boldsymbol{X}_{[-(i,j)]}) - f_{i,j}(\tilde{x}^v_{i,j} = 0|\boldsymbol{X}_{[-(i,j)]})\}, \forall v \quad s.t. \quad \tilde{x}^v_{i,j} = 0$
6 $\tilde{x}^{v^*}_{i,j} \leftarrow 1$
7 **end**
8 **if** $f_{i,j}(\tilde{\boldsymbol{x}}_{i,j}|\boldsymbol{X}_{[-(i,j)]}) - f_{i,j}(\boldsymbol{x}_{i,j}|\boldsymbol{X}_{[-(i,j)]}) > 0$ **then**
9 $\boldsymbol{x}_{i,j} \leftarrow \tilde{\boldsymbol{x}}_{i,j}, \theta_{i,j} \leftarrow 0$
10 **end**
11 **end**
12 **Output** X

Theorem 2. *The propose Algorithm 1 can coverage the Nash Equilibrium in polynomial time.*

Proof. In Algorithm 1, each AP has M^2 neighboring APs (including itself) and their caches can store at most CM^2 contents. In order to maximize the cache hit rate of each AP, the contents that need to be cached will be a subset of the most popular CM^2 contents. Therefore, the complexity of each iteration in Algorithm 1 is $O(N^2CM^2)$. Furthermore, since we only have a finite number of caching policies (at most $\frac{CM^2!}{C!(CM^2-C)!}$) and in a potential game our algorithm provides a non-negative improvement in the potential function, the complexity of Algorithm 1 will still be polynomial. Thus, Algorithm 1 is guaranteed to coverage the Nash Equilibrium in polynomial time. □

4 EXPERIMENTS

We perform the simulations on a real-world dataset. *Traces of mobile video sessions* is collected by the most popular video providers in China. The dataset was collected in March 2016, containing 2 million users watching 0.3 million unique videos in Beijing city. Each trace item records the user ID, the timestamp and the content title and the location where the user watches the video. To simplify and speed up our simulation, we choose a rectangular area (39.7°N − 40.1°N, 116.2°E − 116.6°E) in the central Beijing. We then partition this area into 40×40 grids, i.e., each grid can be abstracted as a $0.01° \times 0.01°$ geographic area with a size of $0.72km^2$. In the following experiments, we consider a 40×40 grid network and each AP has the same coverage area with 3×3 girds,

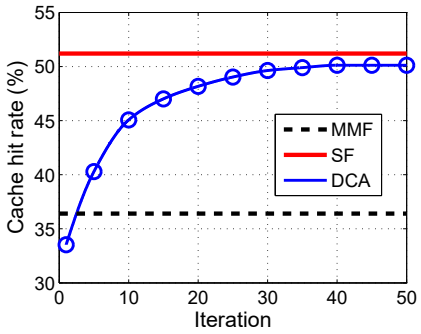

Figure 2: Cache hit rate for different algorithms ($V = 100,000$, $C = 10$).

i.e., $N = 40$ and $M = 3$. We compare our local distributed algorithm (LDA) to the following global and local caching algorithms. (1) Simulated annealing algorithm (SSA) [3] can obtain the global optimum based on global information. (2) Popularity-based algorithm (PA) [1] stores the contents with the highest popularities based on local information. Fig. 2 shows cache hit rate for different algorithms. We observe that the cache hit rate of LDA can iteratively approximate to the global optimum (i.e., SSA); and LDA performs significantly better than PA, because neighboring APs share more information between each other at each iteration leading to higher cache hit rate. In summary, our proposed algorithm only based on the local communication between neighboring APs can achieve the performance of global optimum.

ACKNOWLEDGMENTS

The work was supported in part by the National Natural Science Foundation of China under Grant No. U1611461, 61872215 and 61531006, the National Basic Research Program of China under Grant No. 2015CB352300, and the Tsinghua-Berkeley Shenzhen Institute Grant.

REFERENCES

[1] L. Chen, Y. Zhou, M. Jing, and R. TB Ma. 2015. Thunder crystal: a novel crowdsourcing-based content distribution platform. In *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. ACM, 43–48.
[2] Negin Golrezaei, Karthikeyan Shanmugam, Alexandros G Dimakis, Andreas F Molisch, and Giuseppe Caire. 2012. Femtocaching: Wireless video content delivery through distributed caching helpers. In *Proceedings of the 31st IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 1107–1115.
[3] B. Hajek. 1988. Cooling schedules for optimal annealing. *Mathematics of operations research* 13, 2 (1988), 311–329.
[4] Cisco Visual Networking Index. 2017. Global Mobile Data Traffic Forecast update, 2016-2021. *San Jose, USA: Cisco White paper* (2017).
[5] A. Khreishah and J. Chakareski. 2015. Collaborative caching for multicell-coordinated systems. (2015), 257–262.
[6] A. Khreishah, J. Chakareski, A. Gharaibeh, I. Khalil, and Y. Jararweh. 2015. Joint data placement and flow control for cost-efficient data center networks. In *Information and Communication Systems (ICICS), 2015 6th International Conference on*. IEEE, 274–279.
[7] G. Ma, Z. Wang, M. Chen, and W. Zhu. 2017. APRank: Joint mobility and preference-based mobile video prefetching. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. 7–12.
[8] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu. 2017. Understanding performance of edge content caching for mobile video streaming. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1076–1089.
[9] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C M Leung. 2014. Cache in the air: exploiting content caching and delivery techniques for 5G systems. *IEEE Communications Magazine* 52, 2 (2014), 131–139.