Fair Caching Networks

Yuezhou Liu
Northeastern University

Yuanyuan Li
Northeastern University

Qian Ma
Sun Yat-sen University

Stratis Ioannidis
Northeastern University

Edmund Yeh
Northeastern University

ABSTRACT

We study *fair* content allocation strategies in caching networks through a utility-driven framework, where each request achieves a utility of its caching gain rate. The resulting problem is NP-hard. Submodularity allows us to devise a deterministic allocation strategy with an optimality guarantee factor arbitrarily close to $1-1/e$. When $0 < \alpha \leq 1$, we further propose a randomized strategy that attains an improved optimality guarantee, $(1-1/e)^{1-\alpha}$, in expectation. Through extensive simulations over synthetic and real-world network topologies, we evaluate the performance of our proposed strategies and discuss the effect of fairness.

Keywords

Caching Network, fairness, resource allocation

1. INTRODUCTION

In-network caching is a fundamental enabler of many applications, such as information-centric networks (ICNs), content delivery networks (CDNs), and femtocell networks. Motivated by a series of recent papers (e.g. [1]), we study fairness considerations in the context of the so-called *caching gain rate*. Informally, given a caching strategy X, the caching gain rate of a flow of requests for an item is given by:

$$\lambda \cdot \Delta C(X),$$

where λ is the rate with which the item is requested, and $\Delta C(X)$ is the reduction of routing costs due to caching. Intuitively, this metric incorporates both the popularity of an item, as captured by λ, as well as the benefit of caching in routing (measured in hops, distance traveled, or delay incurred). In contrast to, e.g., cache hit rates or throughputs/rates alone, it incorporates routing costs in the network design objective. Such costs are important: requests served with hit rate 1 at a distant server, in reality, have a lower utility than requests served locally with a lower hit rate.

We propose a fair caching framework to achieve different degrees of fairness w.r.t. requests, content items, as well as users, respectively, in a caching network with arbitrary topology. We aim to find an optimal storage resource allocation that maximizes the total utility as a function of the caching gain rate. To the best of our knowledge, this is the first work that studies fair caching w.r.t. caching gain rates in a multi-hop caching network with arbitrary topology.

Our analysis provides new insights on how fairness w.r.t. caching gain rate affects caching decisions. We observe, for example, that the intuitive behavior of caching highly requested content towards the edge indeed occurs for $\alpha < 1$, but is reversed when $\alpha > 1$. From a technical standpoint, our analysis establishes the submodularity of classic α-fairness objectives when applied to caching gain rates, making it amenable to polynomial-time approximation with a $1-1/e$ approximation. Moreover, our algorithm under the stationary randomized regime is novel, and improves upon the above approximation ratio in the $\alpha \in (0, 1]$ regime.

The full paper of this abstract is available in [2].

2. MODEL

(1) Caching networks: We represent the caching network by a directed graph $G(V, E)$, where V is a set of cache nodes and E is a set of bidirectional edges. We denote by $\mathcal{C}$ the set of items of equal size (see Section 6 of the full paper for an extension to unequal sizes) to be cached. Let

$$x_{vi} \in \{0, 1\}, \quad \text{for all } v \in V, i \in \mathcal{C}, \tag{1}$$

be the indicator variable indicating whether node v stores item i. We denote by the matrix $X = [x_{vi}]_{v \in V, i \in \mathcal{C}}$, the global caching strategy. Each node $v \in V$ is equipped with a cache that can store $c_v \in \mathbb{N}_+$ items, so

$$\textstyle\sum_{i \in \mathcal{C}} x_{vi} \leq c_v, \quad \text{for all } v \in V. \tag{2}$$

We denote each content request by a pair (i, p), where $i \in \mathcal{C}$ is the item requested and $p \subseteq V$ is the pre-established path to the server over which the request message is routed. Let $\mathcal{R}$ be the set of all such requests. Request arrivals follow independent Poisson processes with rate $\lambda_{(i,p)} \geq 0$. A request terminates upon a cache hit (at the serer or an intermediate cache), and a response message carrying the requested item is sent back over reverse path.

We denote by $w_{uv} \geq 0$ the routing cost incurred when transferring an item across edge $(u, v) \in E$. The routing cost for serving request (i, p) is determined by the downstream cost, given by

$$C_{(i,p)}(X) = \sum_{k=1}^{|p|-1} w_{p_{k+1}p_k} \prod_{k'=1}^{k} (1 - x_{p_{k'}i}). \tag{3}$$

We define the difference between the routing costs without caching and with caching as the *caching gain*:

$$\begin{aligned} F_{(i,p)}(X) &= C_{(i,p)}(\mathbf{0}) - C_{(i,p)}(X) \\ &= \sum_{k=1}^{|p|-1} w_{p_{k+1}p_k} \left(1 - \prod_{k'=1}^{k} (1 - x_{p_{k'}i})\right). \end{aligned} \tag{4}$$

Reducing the routing cost of request (i,p) is equivalent to increasing the caching gain.

(2) Utility function and utility maximization: We aim to allocate the cache storage resource fairly for requests using a utility-driven framework. We consider the utility of the *caching gain rate* associated with each request. Mathematically, given caching gain $F_{(i,p)}(X)$ and request rate $\lambda_{(i,p)}$, request (i,p) achieves utility $U(\lambda_{(i,p)}F_{(i,p)}(X))$. To capture fairness, we consider a class of α-fair utility functions, parameterized by $\alpha \in \mathbb{R}_+$:

$$U(z) = \begin{cases} \dfrac{z^{1-\alpha}}{1-\alpha} & \text{when } 0 \le \alpha < 1, \\ \log(z+\epsilon) & \text{when } \alpha = 1, \text{ or} \\ \dfrac{(z+\epsilon)^{1-\alpha}}{1-\alpha} & \text{when } \alpha > 1, \end{cases} \tag{5}$$

where $\epsilon \ge 0$ is a constant. Our goal is to maximize the total utility under the cache storage constraints:

$$\text{Maximize:} \quad G(X) = \sum_{(i,p)\in\mathcal{R}} U(\lambda_{(i,p)}F_{(i,p)}(X)) \tag{6a}$$

$$\text{s.t.} \quad X \in \mathcal{D}_1, \tag{6b}$$

where $\mathcal{D}_1$ is the set of $X \in \mathbb{R}^{|V|\times|\mathcal{C}|}$ satisfying (1) and (2).

3. MAIN RESULTS

We derive the following key results in the paper:

(1) Submodular maximization: The objective G can be naturally expressed as a set function. For $S \subseteq V \times \mathcal{C}$, let $X_S \in \{0,1\}^{|V|\times|\mathcal{C}|}$ be the binary vector whose support is S. We can interpret our objective $G : \{0,1\}^{|V||\mathcal{C}|} \to \mathbb{R}_+$ as a set function $G : V \times \mathcal{C} \to \mathbb{R}_+$ via $G(S) \triangleq G(X_S)$. We show the monotonicity and submodularity of set function G:

Theorem 1. *The objective function $G(S) \triangleq G(X_S)$ of Prob. (6) is a non-decreasing and submodular set function.*

Constraints (1) and (2) define a matroid. Hence, Prob. (6) is a submodular maximization problem under matroid constraints. This problem is NP-hard in general; we discuss polynomial-time approximation algorithms as follows.

(2) Deterministic offline strategy: The greedy algorithm produces a solution within 1/2 approximation factor from the optimal. We can further improve the approximation guarantee to $1-1/e \approx 0.63$, using the *continuous-greedy* algorithm. It maximizes the multilinear extension of the objective $G(X)$ over the reals, obtaining a fractional solution Y in the convex hull of $\mathcal{D}_1$. Solution Y is then rounded to produce an integer solution in $\mathcal{D}_1$ by pipage rounding [3]. Applied to our setting, this yields the following result:

Theorem 2. *If $\hat{X} \in \mathcal{D}_1$ is the integer solution produced by pipage rounding and $X^* \in \mathcal{D}_1$ is the optimal solution of Problem (6), then with high probability we have: $G(\hat{X}) \ge (1-\frac{1}{e})G(X^*)$, for $0 \le \alpha < 1$, and $G(\hat{X}) - G(\mathbf{0}) \ge (1-\frac{1}{e})(G(X^*) - G(\mathbf{0}))$, for $\alpha \ge 1$.*

(3) Stationary randomized strategy: In this setting, we assume that time is slotted, and that at each time slot, a random caching strategy X is sampled from a joint distribution μ over $\mathcal{D}_1$: $\mu(X) = \prod_{v\in V}\mu_v(x_{v1},\ldots,x_{v|\mathcal{C}|})$, where μ_v is the distribution of node v. We aim to decide μ to maximize the total utility of expected caching gain rate of the network, which is defined as:

$$\textstyle\sum_{(i,p)\in\mathcal{R}} U(\mathbb{E}_\mu[\lambda_{(i,p)}F_{(i,p)}(X)]). \tag{7}$$

We denote by y_{vi}, $v \in V$, $i \in \mathcal{C}$ the marginal probability that node v stores item i, i.e., $\mathbb{E}_{\mu_v}[x_{vi}] = \mathbf{P}_{\mu_v}[x_{vi}] = y_{vi}$, and let $Y = [y_{vi}]_{v\in V, i\in\mathcal{C}}$. We propose an *L-method* that maximizes (7) and produces a solution Y within a $(1-1/e)^{1-\alpha}$ factor from the optimal deterministic solution of Prob. (6). It extends the method by Ageev and Sviridenko [3] used earlier in the linear case ($\alpha = 0$). When $0 < \alpha < 1$, this factor is better than the $1-1/e$ ratio of continuous-greedy algorithm.

Theorem 3. *Let Y^* be the optimal solution that maximizes (7) and Y^{**} be the solution produced by L-method, we have: for $\alpha \ne 1$, $G(Y^{**}) \ge (1-\frac{1}{e})^{1-\alpha}G(Y^*)$, and for $\alpha = 1$, $G(Y^{**}) \ge G(Y^*) - c$, where $c = |\mathcal{R}|\log\frac{e}{e-1}$.*

Given the marginal probability matrix $Y \in \mathcal{D}_2$ produced by L-method, a randomized rounding policy is required at each node $v \in V$ to produce a joint distribution μ over $\mathcal{D}_1$. The randomized rounding strategy we adopt is Alg. 2 in [1].

(4) Numerical results: In Fig. 1, we plot the time-average total utilities achieved by different algorithms (Greedy (`GRD`), continuous-greedy (`CG`), and L-method (`L`) as well as path replication combined with the `LRU`, `LFU`, `FIFO` and random replacement (`RR`)) in nine network topologies for the case when $\alpha = 0.8$. We can see that in all topologies, `GRD`, `CG` and `L` outperform four path replication algorithms.

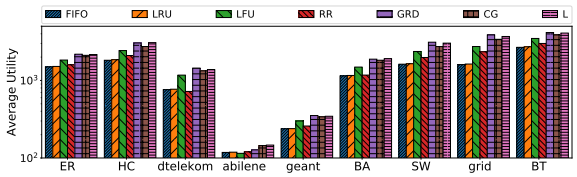
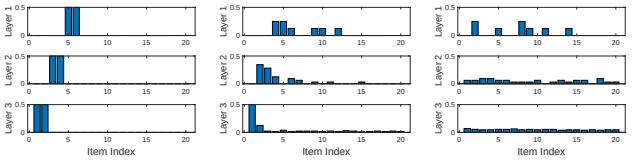


Figure 1: Comparison of average utilities (in log scale).

We also show that content items are more fairly allocated when considering the proposed fair caching framework. Fig. 2 presents the results of content allocation, from which we can observe that content items are more evenly distributed in the caches of each layer as α increases.



(a) $\alpha = 0$ (b) $\alpha = 0.4$ (c) $\alpha = 0.8$

Figure 2: The content allocation in a balanced tree caching network where items with lower indices have higher request rates. A bar at position $i \in \{1,\ldots,20\}$ represents the fraction of total cache space in a layer that is allocated to item i.

4. REFERENCES

[1] S. Ioannidis and E. Yeh, "Adaptive caching networks with optimality guarantees," *ACM SIGMETRICS Performance Evaluation Review*, vol. 44, no. 1, pp. 113–124, 2016.

[2] Y. Liu, Y. Li, Q. Ma, S. Ioannidis, and E. Yeh, "Fair caching networks," *Performance Evaluation*, 2020. `https://doi.org/10.1016/j.peva.2020.102138`.

[3] A. A. Ageev and M. I. Sviridenko, "Pipage rounding: A new method of constructing algorithms with proven performance guarantee," *Journal of Combinatorial Optimization*, vol. 8, no. 3, pp. 307–328, 2004.