Small-cell Caching Mechanism for Multi-Service Providers

Zhiwen Hu, Zijie Zheng, Tao Wang and Lingyang Song
School of Electrical Engineering and Computer Science, Peking University, Beijing, China.

I. Introduction

The rapid growth of data traffic to mobile devices is now challenging the design of future wireless networks, especially at busy hours. Endowed with content-aware and learning technics, the peak traffic demands in busy hours can be reduced through estimating users' preferences and caching popular files at small cell base stations (SBSs) in idle time. This leads users' downloading served by SBSs directly, avoiding crowded and low-speed backhaul connectivity to the content providers. Because of the heterogeneity among users' preferences and types of contents, how to effectively pre-allocate the contents to SBSs is a kernel issue to be discussed.

To deal with the caching problem, several schemes have been proposed in previous works such as [1] and [2]. However, most of the current studies neglect the fact that contents are usually from different providers, who have the incentives to compete for the caching storage to better serve its own users. Another unheeded fact is that the high density of SBSs generates multiple overlapping areas, making the caching process among different SBSs need to be designed jointly. To cope with these two problems, we propose a multi-step multi-object auction mechanism, where providers evaluate their contents and bid for caching space of SBSs at each step and consider the influence of overlapping areas in previous steps. The simulation results shows that our proposition can achieve an effective performance, which keeps improving with higher overlapping coverage of SBSs.

II. System Model

We consider a system with L content providers denoted by $\mathcal{P} = \{P_l \,|\, l = 1, 2 \cdots L\}$. And contents of providers are given by $\mathcal{C} = \{C_{l,k} \,|\, l = 1, 2 \cdots L\}$, where $C_{l,k}$ is the k^{th} content of P_l. The i^{th} SBS in the system is denoted by SBS$_i$, whose storage size is B_i, where $1 \leq i \leq I$. Assume that all the contents are of the same size and one SBS can only cache no more than B_i contents at a time. We use Ω_i to denote the coverage region of SBS$_i$, which makes $\Omega_i \cap \Omega_j$ the overlapping region of SBS$_i$ and SBS$_j$. Users are distributed under the coverage of SBSs, and once being accessed, a cached content can result in a lower delay. To minimize the average delay, the caching should be designed properly, since the overlapping, the user density, and the popularity of contents can jointly influence the effectiveness. The demonstration of our model is shown in Fig. 1.

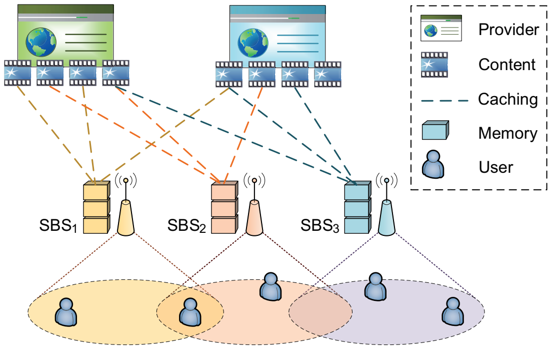

Fig. 1. System model for femto caching among different content providers.

A. Traffic delay, user density and popularity of contents

Since user density and traffic are different in different hours of a day, we suggest that the delay of requesting an uncached content is $\theta(t)$, where $1 \leq t \leq 24$, and the delay of requesting an cached content is a costant θ_c. The user density is given by $u(t, \boldsymbol{x})$, which depends on both time and space. So the average user number at the t^{th} hour can be given by $\iint_\Omega u(t, \boldsymbol{x}) d\boldsymbol{x}$, where Ω is a given region.

We use $\alpha_{l,k}^t$ to denote the time-dependent popularity of $C_{l,k}$, which has a similar profile with the logarithmic normal probability density function. The maximum popularity and lifespan of different contents can be distinct from each other, so we add extra parameters to the original function as $\alpha_{l,k}^t = \int_0^1 af(\frac{t-t_0}{b})dt$, where a denotes the maximum popularity, b denotes the lifespan, and t_0 is the time when $C_{l,k}$ is uploaded. β_l is used to denoted the popularity of provider P_l, therefore the possibility of a user to visit $C_{l,k}$ in a certain hour t can be given by $\phi_{l,k}^t = \alpha_{l,k}^t \cdot \beta_l$. Since each user can request several contents in an hour, the summation of all $\phi_{l,k}^t$ equals to Φ, which is unnecessary to be normalized.

B. Allocation and valuation

To record the caching result, a matrix Γ is used, where $\gamma_{l,k}^{t,i} = 1$ if $C_{l,k}$ is cached in SBS$_i$ at t, and $\gamma_{l,k}^{t,i} = 0$ if not. The constraint is $\sum_{l,k} \gamma_{l,k}^{t,i} \leq B_i$, which means SBS$_i$ can cache no more than B_i contents at a time.

Consider a certain content cached in several SBSs, the value it makes depends on the coverage of these SBSs. The effective coverage region of $C_{l,k}$ at t is given by $\Omega_{l,k}^t = \bigcup_{i=1}^I [\Omega_i \cdot \gamma_{l,k}^{t,i}]$. In this way, the number of users who benefit from the caching of $C_{l,k}$ at time t can be written as $u_{l,k}^t = \iint_{\Omega_{l,k}^t} u(t, \boldsymbol{x}) d\boldsymbol{x}$.

978-1-4673-7131-5/15/$31.00 ©2015 IEEE

It is obvious that a better feedback from users brings higher profit to the provider, so the evaluation of a content is based on the average decrease in latency it can bring by caching. A given caching result can quantify the value of each content, given by $v_{l,k}^t = \phi_{l,k}^t \cdot \big(\theta(t) - \theta_c\big) \cdot u_{l,k}^t$. And the the marginal benefit of the caching the content $C_{l,k}$ in SBS$_i$ is given by $v_{l,k}^{t,i} = \phi_{l,k}^t \cdot \big(\theta(t) - \theta_c\big) \cdot u_{l,k}^{t,i}$. Each provider tends to maximize its own profit through competitions on caching space of SBSs.

C. Problem formulation and constraint

Our main purpose is to minimize the average delay of the access to contents with the consideration of competitions among providers, the objective function is given by:

$$\begin{cases} \min \sum\limits_{l,k} \left[\theta(t)U_s - \big(\theta(t) - \theta_c\big)u_{l,k}^t\right] \phi_{l,k}^t \Big/ (\Phi U_s), \ \ \forall t, \\ s.t. \ \sum\limits_{l,k} \gamma \le B_i, \forall\ i. \end{cases} \tag{1}$$

where U_s is the number of users, and $\phi_{l,k}^t/\Phi$ is the probability of $C_{l,k}$ being requested of each random access.

III. Multi-step Multi-object Auctions

A. The mechanism of multi-step multi-object auction

At each certain hour, $\sum_{i \in I} B_i$ different memory blocks are used to cache contents. The complexity of optimal solution makes it almost impossible to allocate all the contents in only one auction. Our proposal is to auction for $\max\{B_i\}$ times, by which way the b^{th} memory block in each BS is auctioned off in the b^{th} auction. The auctions are held only in the late night when SBSs are lightly loaded, determining the next 24 hours' caching scheme. In order to reduce the frequency of content replacement which burdens the backhaul of SBSs, we also suggest that additional price Δp are charged for replacing the original content if the new one wasn't cached in this BS the last time.

In each auction, the contents from different providers play the role of bidders while the memory blocks in all SBSs play the role of objects, and each bidder can only acquire no more than one object. We denote the bidders as C_n, where $1 \le n \le N$, and the objects as D_m, where $1 \le m \le M$ and $N > M$. The valuation that C_n estimates D_m is denoted by $v_{n,m}$, which is the highest price that C_n would pay for D_m. The profit for C_n to pay for D_m is given by $v_{n,m} - p_m - \Delta p$, where p_m is the price of D_m, and Δp can be omitted if additional price is unnecessary.

B. A graph-based algorithm of perfect matching

Our algorithm of multi-object auction is originated from Demange, Gale and Sotomayor's work [3], which uses bipartite graph to get a perfect matching among bidders and objects. This algorithm can get the maximum social welfare, and is proved to satisfy the VCG principle, where bidders' best strategy is to bid truthfully.

The first step is to add $N - M$ virtual objects with zero valuation therefore the numbers of bidders and objects are equal, and also, the initial prices of objects are all set as zero. Given the valuation matrix $V_{M \times M}$ and the price vector P_M of objects, a preferred-seller graph can be established, which is essentially a bipartite graph where each bidder node is only connected to the object nodes which bring most profit. To get the optimal result, a perfect matching should be found. By continually using the breadth-first-search (BFS) algorithm, the matching is enlarged until no more augmenting path exists. If we get a perfect matching, then the auction is completed, otherwise a set of constricted object nodes can be found in an unsuccessful BFS. We raise the price of this set of objects until at least one of bidders changes his most preferred objects. And if the lowest price of D_m is non-zero, denoted by δ, then the price vector P_m should subtract δ to keep the price as the lowest market-clearing price. After this step, we rebuild the preferred-seller graph until a perfect matching is found.

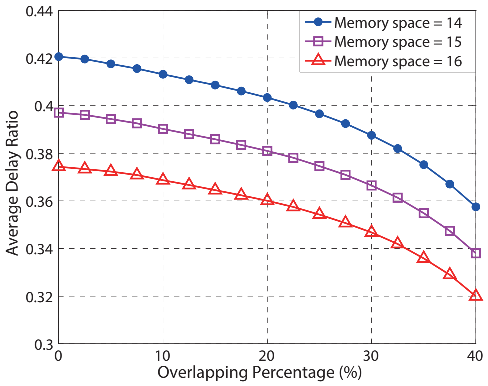

Fig. 2. Average delay ratio with different overlapping percentage and different memory space in each SBS.

IV. Simulation Results

We set 10 uniformly distributed SBSs and 5 different providers in our system, where each SBS's coverage radius is 50m, and the total number of the contents is 50. The simulation result is shown is Fig. 2. We simulate a 24-hour situation and calculate the average delay ratio, which is defined as the ratio of average delay with caching to the average delay without caching. The overlapping percentage represents the degree of overlapping coverage of SBSs, which can be calculated by $\left[\sum_{i=1}^{I} S(\Omega_i) - S(\Omega_s)\right] \big/ \left[S(\Omega_s)\right]$, where $S(\Omega_s)$ is the area of the whole region and $S(\Omega_i)$ is the coverage area of SBS$_i$.

Our algorithm has achieved prominent effectiveness in caching strategy. For instance, when SBSs have no overlapping coverage, the average delay with caching is about 0.4 of that without caching. In addition, given a fixed memory space of each SBS, a higher overlapping coverage results in a lower latency. And given a fixed overlapping value, an increase in memory space assures better performance.

References

[1] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," *INFOCOM12*, Orlando, FL, Mar., 2012.

[2] H. J. Kang, K. Y. Park, KM Cho, C. G. Kang, "Mobile caching policies for device-to-device (D2D) content delivery networking," *INFOCOM14*, Toronto, ON, May, 2014.

[3] G. Demange, D. Gale, and M. Sotomayor, "Multi-Item Auctions," *Journal of Political Economy*, vol. 94, no. 4, pp. 863-872, Aug., 1986.