In-network Caching Assisted Wireless AP Storage Management: Challenges and Algorithms

Zhongxing Ming
Tsinghua University
4-104 FIT Building, Haidian, Beijing, 100084
mzx09@tsinghua.edu.cn

Mingwei Xu
Tsinghua University
4-104 FIT Building, Haidian, Beijing, 100084
xmw@cernet.edu.cn

Dan Wang
The Hong Kong Polytechnic University
Hung Hom, KL, Hong Kong
csdwang@comp.polyu.edu

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Design

Keywords

Wireless caching, algorithm, information-centric networking

1. INTRODUCTION

Wireless mobile data traffic is expected to increase by a factor of 40 over the next five years [2]. Due to the scarcity of resources, wireless data network has started to pose unacceptable delays to end users. The delays experienced by end users have important economic consequences. Zona Research reported that the potential losses in 2001 due to unacceptably slow response times are estimated to be over $25 billion [2].

Caching of objects near clients is a commonly used approach to overcome such challenges. Many studies suggest caching at mobile devices. Such approaches often have high difficulty in motivating the participants' incentives to share their batteries and limited bandwidth. Another approach is caching at wireless access points (APs), which relieves the limitations of handsets. However, AP caching often faces the challenge of limited storage capacity.

Recently we see the emergence of Information-Centric Networking (ICN). One of the major features of ICN is the universal in-network caching, which enables routers to cache contents along the path. A recent trace-driven analysis found that beyond the benefits of traditional edge caching, a further 21% of requests can be served within only a single AS hop by enabling in-network caching [5]. In-network caching has attracted much research attention (e.g., DONA [3]), and vendors (e.g., Huawei) are developing routers with caching capability.

SIGCOMM'13, August 12–16, 2013, Hong Kong, China.
ACM 978-1-4503-2056-6/13/08.

The goal of this paper is to improve wireless AP caching by leveraging in-network caching. We observe that by treating routers as an in-network storage extension, we can relieve the storage limitation of APs. The unique challenge is that APs and routers cannot have a full collaboration, which makes the problem different from traditional cooperative caching problems. This is because, first, the scale of the Internet is too large for APs and routers to globally collaborate. Second, there are potential marketing, policy, security and privacy concerns that prevent in-network caches from making decisions on behalf of edge APs. However, in spite of lacking full collaboration, APs can indeed get some caching information from in-network caches. Many ICN proposals support the dissemination of caching status. For example, [1] provides an Event Notification Service that notifies all Subscribers of the stored contents. As a result, we study how APs can optimize caching decisions by using in-network caching information without controlling routers.

We prove that the problem is NP-complete. We show that in the ideal situation that global in-network caching information is known, we can develop a 2-competitive polynomial offline algorithm. We also show that no online algorithm can achieve a constant competitive ratio against the offline algorithm. We finally develop an in-network cache assisted AP caching algorithm (ICA) for the practical situation that part of in-network caching information is known. ICA adaptively leverage different levels of these information.

2. IN-NETWORK CACHE ASSISTED WIRELESS AP CACHING FRAMEWORK

Figure 1 provides an overview of the framework. Upon receiving a request for information object, an AP responds to the mobile node if it finds the object in a local cache, otherwise it sends the request to the network. APs leverage routers' caching information to make caching decision but do not control routers. The equation for maximizing delay reduction can be represented as:

$$BF = \sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij} BV_j(c_i) = \sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij} \times d_j(c_i) \times p_j(c_i) \quad (1)$$

where c_i denotes content i, s_i denotes i's size, B_j denotes AP_j's storage capacity, $d_j(c_i)$ denote the delay incurred when transferring c_i to AP_j, $p_j(c_i)$ denote the popularity of c_i at AP_j. The 0-1 decision variable x_{ij} indicates whether c_i is stored in AP_j. Given the existence of in-network caches, we study how to specify x_{ij} to maximize Eq.(1) subject to AP storage capacity B_j.

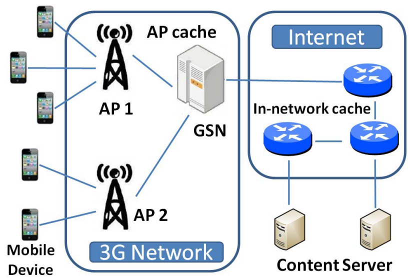


Figure 1: The network model with simplified 3G architecture

3. PRELIMINARY FINDINGS

We first analyze the difficulty of the problem.

Theorem 1: The problem of finding the optimal configuration of Eq.(1) is NP-complete.

The intuition of the theorem's proof is that we can reduce N independent 0-1 knapsack problems to the problem defined by Eq.(1). We then report our preliminary findings by three perspectives.

1) ***Offline strategy***: This is the ideal case that APs have perfect knowledge of in-network caches. We propose the following 2-competitive offline algorithm for this case. We use the offline strategy as a performance bound for further analysis.

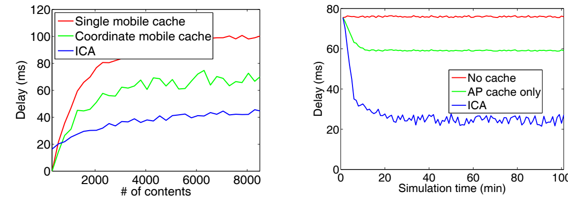
Algorithm 1 Offline Greedy

Each time when an object c_i which is not currently cached arrives, calculate its unit benefit value, which is defined as $d_j(c_i)p_j(c_i)/s(c_i)$. If c_i has a higher unit benefit value than the object with the minimum unit benefit value that is current stored, then replace the object with c_i.

2) ***Online strategy***: This is the worst case that APs do not have any in-network caching information. We show that although it is difficult to get an optimal solution, APs may make bad decisions if they are unaware of in-network caching information. The intuition is that online algorithms may perform poorly when APs always select contents which are cached by nearby routers.

Theorem 2: No online algorithm can achieve a constant competitive ratio as opposed to the offline algorithm in term of total delay reduction.

3) ***In-network caching assisted strategy (ICA)***: We then study the practical case that APs can get part of in-network caching information. We propose a simple yet efficient ICA algorithm which works as follows. Define $BV_j(c_i) = d_j(c_i)p_j(c_i)/s(c_i)$ if AP_j has c_j's in-network caching knowledge, and $BV_j(c_i) = \infty$ otherwise. When content c_i arrives at AP_j, AP_j performs LRU algorithm if the c_i has already been cached. If c_i is not cached, AP_j checks if c_i's in-network caching information is known. If yes, AP_j first places c_i to the head of the LRU queue and then evicts the item (including c_i) with the minimum BV. If not, AP_j sets c_i the head of the queue and performs either of the following operation 1) evict the the tail of the queue if $BV_j(c_{tail\_of\_queue}) = \infty$ or 2) evict the object with the minimum benefit value if $BV_j(c_{tail\_of\_queue}) \neq \infty$.



(a) ICA vs. Handset Caching (b) ICA vs. AP Caching

Figure 2: Preliminary Results

4. EVALUATION

We have conducted a trace-based evaluation, where simulations were conducted in an event-driven C simulator. The trace was captured from a commercial website during a six-month-period from 2011 to 2012, which totaled 824,741 URL-requests sent by 420,324 users. We treated each URL as an information object. The simulation was run on the topology of CERNET2, which is the world's largest IPv6 network. We set each router connected by 1,000 APs. According to the trace, each AP received content requests with the intensity of 0.1 req/s under Poisson distribution. We compared ICA to pure AP caching and handset caching by running the simulation for 100 minutes, which was long enough to get a stable result. Qian et al. [4] performed a comprehensive network-wide study on handset caching and found that the Least Recently Used (LRU) algorithm was the most widely used replacement policy for commercial smartphones. As a result we set handsets use LRU. We also used LRU for traditional AP caching. We measured the system performance using *average system delay* which was calculated by Eq.(2).

We see that ICA significantly outperformed handset caching. When content number=6,000, ICA reduced the delay of single and cooperative handset caching by 60% and 33%, respectively. We also see that ICA achieves more delay reduction that pure AP caching. Throughout the simulation, ICA improved the delay reduction of pure AP caching by 58% on average.

$$\frac{\sum_{j=1}^{N}\sum_{i=1}^{M} delay\_for\_object_i\_at\_AP_j}{MN} \quad (2)$$

5. REFERENCES

[1] A. Eriksson, B. Ohlman, and K. A. Persson. What are the services of an information-centric network, and who provides them? In *IEEE AP2PS*, 2012.

[2] N. Golrezaei, G. A. Dimakis, and M. F. A. Wireless device-to-device communications with distributed caching. In *ISIT*, 2012.

[3] T. Koponen, B. Chawla, A. Emolinskiy, H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *ACM SIGCOMM*, 2007.

[4] F. Qian, K. S. Quah, J. Huang, J. Erman, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Web caching on smartphones: Ideal vs. reality. In *ACM MobiSys*, 2012.

[5] G. Tyson, S. Kaune, S. Miles, Y. El-khatib, A. Mauthe, and A. Taweel. Trace-driven analysis of caching in content-centric networks. In *IEEE ICCCN*, 2012.