Demo: Crowd-Cache – Popular Content for Free

Kanchana Thilakarathna
UNSW & NICTA, Australia
kanchana.thilakarathna
@nicta.com.au

Fangzhou Jiang
USYD & NICTA, Australia
fangzhou.jiang
@nicta.com.au

Sirine Mrabet
NICTA Australia
sirine.mrabet
@nicta.com.au

Mohamed Ali Kaafar
INRIA France, NICTA Australia
dali.kaafar
@nicta.com.au

Aruna Seneviratne
UNSW & NICTA, Australia
aruna.seneviratne
@nicta.com.au

Prasant Mohapatra
UC Davis, USA
pmohapatra
@ucdavis.edu

ABSTRACT

Crowd-Cache is a novel crowd-sourced content caching system which provides cheap and convenient content access for mobile users. Our system exploits both transient colocation of devices and the spatial temporal correlation of content popularity, where users in a particular location and at specific times would be likely interested in similar content. We demonstrate the feasibility of Crowd-Cache system through a prototype implementation on Android smartphones.

Categories and Subject Descriptors

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

Keywords

Distributed Caching; Crowd Sourcing; Mobile Content Distribution

1. INTRODUCTION

The emergence of smart mobile devices have led to an explosion in mobile data traffic [2]. This traffic is being generated by the user consuming rich media, especially video via the over the top services such as catch-up TV and YouTube. It has been reported that the current cellular network operators are struggling to cope with the increasing demand [1] and as a result, today a majority of cellular operators only provide capped data plans.

Content caching and peer-to-peer dissemination are two of the common approaches to address the needs of growing mobile data traffic. Traditional in-network caching exploits the locality of interest by storing the content as close to the location of the interest. However in mobile networks, as the delay and cost are incurred on the last hop wireless link, the traditional in-network caching helps in reducing the latency and bandwidth only in the backhaul links.

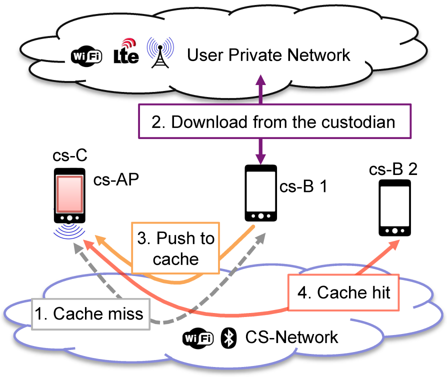

Figure 1: Operations of the Crowd-Cache System

We propose and demonstrate a novel content caching scheme which combines the concepts of traditional in-network caching, crowd-sourcing and peer-to-peer data delivery and harness the advanced capabilities of mobile devices rather than thin clients as traditional systems.

2. CROWD-CACHE: AN OVERVIEW

Our system exploits both transient colocation of devices and the epidemic nature of content popularity, where users in a particular location and at specific times would be likely interested in the same set of content. The key idea of the Crowd-Cache system is to leverage on the possibility to gather such content of interest into a localised cache so that further requests would not require any expensive bandwidth cost.

The architecture of the proposed system is illustrated in Figure 1. We consider two modes of operation: a browser-only mode, referred as *cc-B*, and a cache-mode, referred as *cc-C* for crowd-source caching. *cc-C* users will act as Crowd-Cache network access-points, denoted *(cc-APs)*. In practice, we envision several reward-based schemes to gratify *cc-C* users, which provides sufficient incentives for becoming a *cc-C*. The *cc-C* devices advertise their availability via specific SSIDs broadcast (e.g. cc-SSID), which is operated by simply activating the WiFi Hotspot mode. All mobile devices running a *cc-B* can then associate with the *cc-AP*. Associations to the *cc-APs* are handled by the *cc-B* app and is secured using e.g. existing standard WLAN security mechanisms such as WPA authentication mechanisms.

MobiSys'14, June 16–19, 2014, Bretton Woods, New Hampshire, USA.
ACM 978-1-4503-2793-0/14/06.
http://dx.doi.org/10.1145/2594368.2601464.

When a user requests a content via *cc-B*, the device associates with available *cc-AP* and then the corresponding *cc-C* app checks the cache for the requested content. In case of a cache hit, the content is delivered from the *cc-C* to the *cc-B* locally. In case of a miss, *cc-B* is informed which triggers a switch of the network connection from *cc-AP* to the user *private network* (e.g. LTE, 3G network). Then the *cc-B* downloads the content from the external content custodian as shown in step 2 in Figure 1.

Contribution to the crowd cache happens whenever a user downloads content via his *private network connection* due to a cache miss. When first connecting to a *cc-AP*, *cc-B* pushes the content that lies in the device local cache to *cc-C* of the *cc-AP* (step 3 in Figure 1). The *cc-C* makes a local decision as to whether it is useful or not to update its cache with the new content depending on the content replacement strategy adopted by the *cc-C*. Then, whenever another *cc-B* user requests the same content, a cache hit happens as illustrated in step 4 in Figure 1.

We model the crowd-cache system based on real-world video access patterns of mobile users. The preliminary results of real data-driven simulations show that it is possible to achieve up to approximately 90% cache hit rate, while more than 80% of users saving approximately 80% of cellular bandwidth without significant impact on the device energy consumption.

2.1 Application Scenario

We envisage several deployment scenarios of the Crowd-Cache system. In the following, we describe a real-life deployment which considers a public transport scenario (e.g. Bus), where users commute to work. Let's assume for simplicity, that the bus driver is a registered *cc-C*. Thus, the bus driver's mobile device will act as a *cc-AP* within the bus. Initially, the cache of the bus driver's *cc-C* only contains data that has been downloaded via the driver's own (private) network connection (for example for his/her own use). When a passenger who has subscribed to the Crowd-Cache service gets on the bus, the *cc-B* discovers the availability of a *cc-AP* (via the active discovery mode searching for specific AP beacons).

First, *cc-B* pushes locally cached content (e.g. videos downloaded whilst at home or at the bus stop via user own network connections) to the bus driver's *cc-C*. When a newcomer (a passenger in our scenario), sends a requests via the Crowd-Cache system to access some internet content (e.g. YouTube videoclips, hot trending news, etc.), *cc-B* first checks the bus driver's *cc-C* to verify whether the content is available. In case of a cache hit, the content is obtained through the bus driver's *cc-C*. Otherwise, the user request is served through his own network connection if available (the user is notified that the connection is redirected to outside the local *cc-AP*). More generally, each passenger who gets on the bus and has subscribed to the Crowd-Cache system, transfers its own cached data to the bus driver's *cc-C* cache. In such a semi transient environment, after a while the cache of the *cc-C* will most likely contain the popular videos, thus increasing the cache hit rates.

2.2 Incentives for users

The main incentive to use the *cc-B* is mainly driven by free access to popular content, while contributing from time to time to the cache. When users encounter a crowd cache miss, there is no additional cost, except from contributing via a cheap connection (e.g. WiFi) to the crowd cache so that potential further requests for the same content from other users can be served from the cache. In fact, the crowd cache here is constituted by a collaborative effort from multiple users gathered in the same area for a particular lapse of time. In return, users could benefit from each others effort to save on downloads costs exploiting existing content in the "crowd-sourced" cache.

From the *cc-C*'s perspective, the incentives for participating in the system are twofold. First, *cc-C* inherits all incentives from *cc-B*'s as they still benefit from the content downloaded to the *cc-AP*. Second, we envision several ads-based reward schemes to gratify *cc-Cs* who contribute to the system. Whenever a *cc-C* serves for a cache hit, it also displays an advertisement. Since these ads are locally served from a *cc-C*, such data will not be accounted on the users private network quotas. The *cc-Cs* and *cc-Bs* will upload their impression counts whenever they have access to an Internet connection via a *private network connection*, e.g when they have access to a low cost network at home. *cc-Cs* with a high number of ads impressions get a monetary reward, as a portion of the ads monetary revenue generated by the ads clicks and/or impressions and affiliate ad campaigns. In the above model, the more a *cc-C* supplies content, the more likely it will get rewarded. This represents a clear incentives for *cc-Cs* to participate and operate as a cache contributor whenever they can.

Another important aspect to consider is the potential drain of battery from the content pushing process to the *cc-C*. However, the extra energy consumption of uploading will be offset by the reduction in energy consumption when *cc-B* downloads large volumes of data from the *cc-C*[1]. In the case of *cc-C*, we consider the device is connected to a power source during the hosting of *cc-AP*, e.g. bus driver's *cc-C* device needs to be connected to a power source.

3. DEMONSTRATION

We demonstrate the feasibility of Crowd-Cache system using a prototype Android app. Several Android smartphones are used to demonstrate the two operation scenarios of cache hit and cache miss, while another Android device is used as *cc-C*. We will ask interested conference participants to query and watch their preferred video clips on YouTube using Crowd-Cache app. Then, we revel among other metrics the actual cache hit rate achieved during the demonstration period based on the participants' transient co-location of content preference.

4. REFERENCES

[1] ByteMobile. Mobile analytics report. In *http://www.bytemobile.com*, oct 2011.

[2] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2011–2016. In *http://www.cisco.com.*

[3] H. Petander. Energy-aware network selection using traffic estimation. In *Proc. of the 1st ACM workshop MICNET '09*, pages 55–60, Beijing, China, 2009.

---

[1] It has been shown that the energy consumed by WiFi is significantly lower than those of cellular networks mainly due to the heavy traffic load in cellular networks leading to lower speeds and thus taking longer time to complete data transfers [3].