Enabling Offline Access to Facebook Streams on Mobile Devices

Ye Zhao
Dept. of Information and Computer Science
University of California, Irvine
yez@uci.edu

Ngoc Do
Dept. of Information and Computer Science
University of California, Irvine
nmdo@uci.edu

Shu-Ting Wang
Dept. of Computer Science
National Tsing Hua University, Taiwan
kelvinstwang@gmail.com

Cheng-Hsin Hsu
Dept. of Computer Science
National Tsing Hua University, Taiwan
chsu@cs.nthu.edu.tw

Nalini Venkatasubramanian
Dept. of Information and Computer Science
University of California, Irvine
nalini@ics.uci.edu

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Offline, Prefetching, Rich Content, Facebook

1. INTRODUCTION

The recent years have witnessed the wild success of social networks and social media sites such as Facebook, Twitter, LinkedIn, Google+, and Instagram – indeed, many people rely on these social networks to communicate with their friends, family and community on a day to day basis. Via these sites, users exchange rich media contents, such as video, audio, and images for an enhanced user experience. With the growing popularity of mobile communications, users connect to these social networks using mobile devices (e.g., smartphones, tablets) and expect to derive the same experience no matter how and where they connect. It is reported that 93% of Android smartphone users in India use social networks on their smartphones [6]. Another survey, done in Canada, shows that 61% of users access Facebook through mobile devices [2]. These market reports indicate that social media streams, carrying rich content, have become key aspects of how people view information and how society interacts today.

However, several challenges arise in enabling seamless rich social media experience on mobile devices, including: (a) wireless network availability is sporadic (accessibility of WiFi accesss points, unpredictable data rates in 3G networks) (b) the shared network bandwidth is limited in public locations, and (c) dataplans are becoming volume-driven and hence costly. All of the above result in degraded user experience and limit the exchanges of social media information on mobile devices.

Today, the social media mobile applications (apps), such as mobile Facebook [3], expect always-on connectivity. These apps synchronize with social networks when mobile users launch the apps and mobile devices are connected to the Internet. In the absence of connectivity at the time users wish to use the apps, users do not have access to up-to-date social media content. We believe that offering offline access for these apps by prefetching social media contents is necessary for mobile users in order to interact with social contents when Internet access is not available.

In this work, we design and prototype *OfflineFacebook* [5], a mobile application to enable seamless, personalized, and effective experience of social media access on mobile devices. More specifically, we implement a custom Facebook client on mobile devices to automatically prefetch user-generated contents from Facebook streams to provide mobile users uninterrupted access to social media. Our goal is to build a proof-of-concept platform and show the feasibility of this emerging feature. Although our app is Facebook specific, the feature is general for most social media and social network applications. Meanwhile, we seek to shed lights on its potential impact on the way mobile users interact with social media.

2. APPLICATION ARCHITECTURE

Fig. 1 depicts the system architecture of *OfflineFacebook* app. It is structured into two layers: *User Interface (UI)* layer and *Service* layer. UI layer consists of three components:

(i) Authenticator. The app begins with the user logging into his or her Facebook account through the Authenticator component. Proper authentication is required by Facebook for accessing the user's newsfeed stream (i.e. updates from his or her friends). This is achieved by obtaining the access token on behalf of the user via the login service.

(ii) Content Displayer. The app displays the user's Facebook stream through the Content Displayer component. It also provides interfaces for the user to interact with the con-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Middleware 2013 Posters and Demos Track, December 9-13, Beijing, China
Copyright 2013 ACM 978-1-4503-2549-3/13/12 ...$15.00.

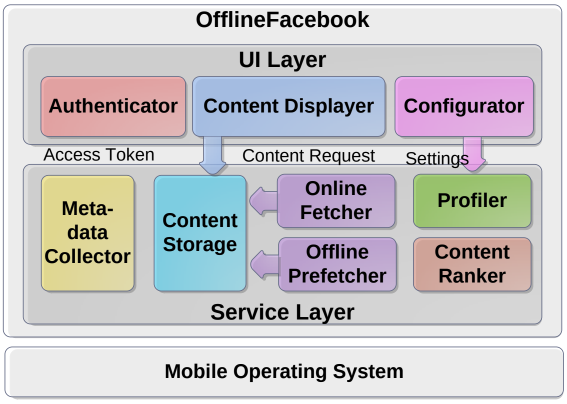
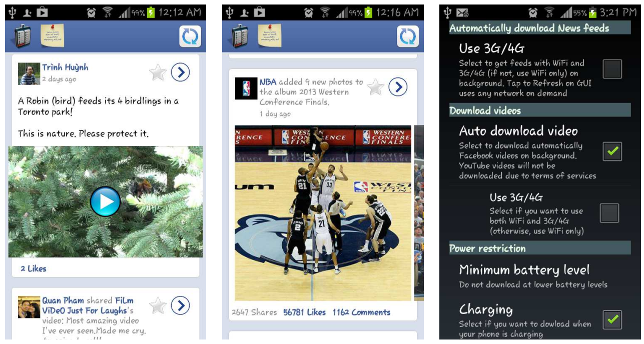


Figure 1: System architecture of OfflineFacebook.



(a) (b) (c)

Figure 2: Snapshots of the oFacebook app: (a) show a video content, (b) show an album content, and (c) set configuration parameters.

tents, e.g., play the embedded video, like or comment on updates, etc. Fig. 2a and 2b present snapshots of the interface.

(iii) Configurator. The app also provides interfaces to change system settings on prefetching conditions (e.g. network and battery). This is achieved through the Configurator component. Fig. 2c presents a snapshot of the interface.

Service layer works as the core of the application. It systematically detects rich contents from the mobile user's social media stream and intelligently prefetches contents in order to provide seamless access under intermittent Internet access. It consists of the following components:

(i) Profiler. The component monitors network and battery conditions on the device via various metrics, including network type, signal strength, and battery level. It also monitors the storage and size of total downloads to avoid filling up the storage space. The aggregated information is used to decide when to prefetch contents.

(ii) Meta-data Collector. The component periodically pulls the meta-data from the user's Facebook newsfeed stream to get updates of new rich contents since the last pull. This is done when the network is available and the remaining battery level is above an operational threshold. The meta-data is saved onboard for later processing.

(iii) Content Ranker. To provide intelligent prefetching, the component utilizes machine learning techniques to predict the probability the user would view a specific content. The ranking algorithm is based on social relations between entities in the social network, user interests and content popularity. The prediction is crucial for the prefetching service to invest the limited resources (network, energy, and storage) on only those contents that are most likely to be watched.

(iv) Offline Prefetcher. The component decides on which contents to prefetch and when to prefetch. This is achieved by carefully taking into account the variations in the sizes of the contents (from the Meta-data Collector), their expected probability of being watched (from the Content Ranker), the network and battery conditions (from the Profiler). Downloaded contents are saved in content storage and displayed to the user when requested.

(v) Online Fetcher. If a user requests a content that hasn't been prefetched, the component downloads the content on the fly. This requires the Internet connectivity at the request time, which resembles the behavior of traditional social media applications.

3. PROTOTYPE IMPLEMENTATION

The prototype of *OfflineFacebook* has been implemented on Android platform in Java. Facebook provides a SDK for Android [4] to facilitate the development of apps with Facebook integration. *OfflineFacebook* takes advantage of its library for Login with Facebook authentication, reading from and publishing to streams through Facebook APIs.

The service layer runs as a background service [1] on Android containing two main threads: one to collect the meta-data, which includes information on social media updates as well as social interactions (i.e. likes and comments for updates). The app saves the meta-data into its SQLite database and uses the information later for displaying the feeds and prefetching media contents.

The other thread is to prefetch media contents detected in the updates, which includes Facebook profile pictures for authors, likers and commenters of feeds, and photos or videos in feeds. The app stores the downloaded contents in local storage waiting to be requested for display. All saved contents will be purged after a specific period (one week by default) to save storage.

4. DEMONSTRATION

In this demonstration, we run the *OfflineFacebook* application along with the official Facebook application on an Android device and show how the *OfflineFacebook* app enables offline navigation to Facebook streams. In particular, during the demonstration, we disable the WiFi connection on the device and switch between both apps to compare the difference. The demonstration hence has minimum requirements in terms of equipments, space and setup time.

5. REFERENCES

[1] Android background service.
[2] Canada-new media trend watch long-haul. `http://tinyurl.com/bv6p7mp`.
[3] Facebook mobile app. `https://www.facebook.com/mobile/`.
[4] Facebook SDK for Android. `https://developers.facebook.com/docs/android/getting-started/facebook-sdk-for-android/`.
[5] OfflineFacebook. `http://www.ics.uci.edu/~dsm/oFacebook`.
[6] Smart and social: Android phones in india.