Demo: *PAUL* - Proactive Automated mobile User-centric content deLivery*

Mohamed A. Abd ElMohsen[1,2], Omar K. Shoukry[1,2], Hesham El Gamal[3], Tamer ElBatt[1,2], Nayer M. Wanas[4], Mohamed Abdel Raouf[4], Mostafa A. Zakaria[4], Ahmed I. Abdelkader[4], Hakem M. Zaied[4]
{massem, oshoukry}@inmobly.com [1]Nile University [2]Cairo University [3]Ohio State University [4]Inmobly

Categories and Subject Descriptors
C.2.3 [**Network Operations**]: Public networks

Keywords
Content pre-fetching, behavioral models, scheduling, smart phone traces, traffic offloading

1. INTRODUCTION
Motivation The wide adoption of bandwidth intensive multimedia applications on smart phones has been driving the growth of Mobile Data Traffic. The predictable mobile user behavior can be leveraged to retrieve content *before* demand to relieve cellular congestion [1].

Objectives In this demo, we exhibit the novel concept of proactive content delivery which performs "temporal" offloading to reduce congestion in cellular networks. *PAUL* is an Android Mobile App that showcases the proposed concept and system.

System Overview *PAUL* solves the cellular congestion problem via leveraging the users' predictable behaviors, which have tremendous impact on the traffic load. Moreover, it significantly enhances the user experience via eliminated download delays and reduced cost. *PAUL* proceeds through three major steps towards retrieving content before demand. First, it logs the behavior of the mobile users throughout the day, namely content consumption, Wi-Fi data rate and battery usage. Second, the logged traces are used to build probabilistic user behavioral models (profiles). These user profiles capture the resource-side (Wi-Fi data rate and battery conditions) and the demand-side (content consumption). Third, we propose a greedy, low-complexity, proactive scheduling algorithm. It utilizes the built profiles to create a schedule for retrieving content before demand over Wi-Fi connectivity opportunities throughout the day, resides on the cloud and constitutes the centerpiece of PAUL.

2. PAUL DEMO
Equipment We use several smart phones with Android 4.0 (or higher) OS, Wi-Fi connectivity and a laptop for illustrating user behavioral modeling. Demo setup requires less than five minutes.

Description The *PAUL* demo is based on an Android application hosting the logging and content fetching software modules on the smart phone, in addition to Python scripts running on the cloud server. The server side hosts the user behavioral modeling and proactive scheduling modules. The system demonstrates the following software modules and technologies: i) Logging module: records traces of the battery, Wi-Fi and content consumption on popular apps, ii) Probabilistic behavioral modeling: generates user profiles based on the logged data and iii) Proactive Scheduling: leverages the profiles to build content download schedules that are then used to retrieve content before demand. The demo also shows the direct impact the user profile has on pre-fetching opportunities, as targeted by the system to increase the probability of the user finding the content item already in its cache, when needed.

Figure 1: PAUL Concept

The demo hosts three major research components: i) A novel system architecture for proactive content delivery to mobile users with the core modules (behavioral modeling and proactive scheduling) residing on the cloud, ii) User behavioral modeling translates the logged data into representative histograms, for all time slots of the day. Three models are generated: a) The content model capturing the probability of consuming one of the content categories in each time slot, b) The battery model capturing the probability of being in one of three battery states (Charging, High and Low) in each time slot and c) The data rate model capturing the probability of having the average Wi-Fi download data rate in a certain range, in different time slots and iii) The proactive greedy scheduling algorithm decides which content item to assign to which slot of the day, depending on the resource availability and the probability of demanding this content in the future. The algorithm proceeds in N Rounds, where N is the number of slots per day, whereby a content item is assigned to a slot in each round. Within a single round, the algorithm assigns a content item based on its future demand and resource availability.

3. REFERENCE
[1] H. El Gamal, J. Tadrous, A. Eryilmaz, Proactive Resource Allocation: Turning Predictable Behavior into Spectral Gain, 48th Annual Allerton Conference on Communication, Control and Computing, Sept. 2010.

*This work was funded in part by a research contract with the Egyptian National Telecom Regulatory Authority (NTRA) and the Information Technology Industry Development Agency (ITIDA) under Round 9 Call For Proposals.

MobiSys'13, June 25–28, 2013, Taipei, Taiwan.
ACM 978-1-4503-1672-9/13/06.