Poster: A Virtual Sensing Framework for Mobile Phones

Jon C. Hammer, Tingxin Yan
Dept. of Computer Science and Engineering, University of Arkansas
{jhammer, tingxiny}@uark.edu

Categories and Subject Descriptors

C.5.3 [**Computer System Implementation**]: Microcomputers—*Portable devices*

Keywords

Mobile devices; Virtual Sensor; Sensing Framework; Behavior Inference

1. VIRTUAL SENSING FOR MOBILE PHONES

Much research has been done regarding the use of hardware sensors in smartphones, leading to the emergence of mobile sensing systems such as activity recognition, localization, and proximity sensing [2]. Although hardware sensors provide informative contextual information, continuous sensing and inference could incur prohibitive energy costs.

We argue that besides hardware sensors, there exist many *virtual* sensors on mobile phones that are continuously available and inexpensive to query from the operating system and services. Examples include screen status, mobile app usage, app notifications, and responses to notifications. Our core observation is that events triggered by the operating system or services provide continuous data streams that indicate user contexts. For example, the frequency and duration of screen sessions indicates how busy a mobile user is, and the frequency of application notification arrivals and departures indicates how mobile users communicate with others.

This work presents a virtual sensing framework by exploiting operating system events for energy efficient context inference. Specifically, we investigate features that can be extracted from virtual sensors and used to infer the *logical status* of mobile users, such as `isWorking`, `isSocial`, and `isStressful`. The preliminary results indicate promising performance and suggest several potential applications.

2. DESIGN AND IMPLEMENTATION

Our framework acts as a middleware between the OS and applications, as shown in Figure 1. The framework provides three core functionalities with negligible energy cost: (1) It continuously logs operating system events and abstracts them as *data probes*, and provides APIs to serve applications. (2) It extracts descriptive features from data probes to enable context inference. (3) It enables various applications by using the context inference results.

MobiSys'14, June 16–19, 2014, Bretton Woods, New Hampshire, USA.
ACM 978-1-4503-2793-0/14/06.
http://dx.doi.org/10.1145/2594368.2601457.

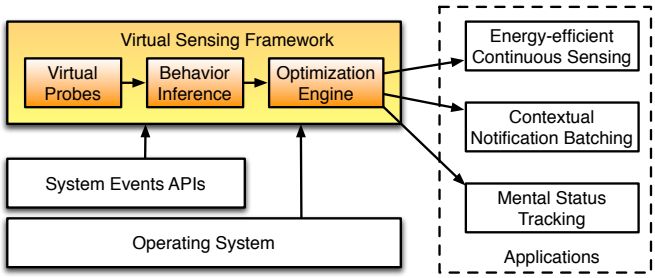

Figure 1: The system architecture of the proposed virtual sensing framework.

As a preliminary work, we present four virtual probes (summarized in Figure 2) for logical status inference. Our observations indicate strong correlation between many OS events and the logical status of mobile users, such as screen status and `isWorking`. Inspired by such observations, we designed a set of descriptive features for logical status inference. For example, queueing parameters, such as inter-arrival time and service rate, are often good features to classify `isWorking` and `isSocial`. Furthermore, connected network information, such as BSSID, would be useful for augmenting the `isWorking` prediction, as location is likely to be an important factor in such a classification.

Virtual Probes	Features	Logical Status
App Notification	Arrival/Departure Rate, # Responses, App Owners	IsWorking, IsSocial, IsStressful
Screen State	Duration, Frequency, App Sequence	IsWorking, IsStressful
Foreground Activity	App category, App Sequence, Trigger App, Trigger Method.	IsWorking, IsSocial, IsHappy
Network Connection	Connection ID, Session Duration	Logical Location

Figure 2: Virtual probes and their features in the proposed virtual sensing framework.

We implemented our virtual sensing framework for unmodified Android devices, inspired by the Funf Open Sensing Framework [1]. Our measurement results indicate that the continuous usage of our framework consumes 0.5% of total battery, an imperceptible amount for mobile users. Several applications can be enabled using our framework, including controlling hardware sensors for energy-efficient continuous sensing, mobile system performance optimization such as notification batching, and efficient logical status tracking such as `isWorking`, `isSocial`, and `isStressful`.

3. REFERENCES

[1] N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland. The social fMRI: measuring, understanding, and designing social mechanisms in the real world. In *Ubicomp*, 2011.

[2] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 2010.