Sense-Aid: A framework for enabling network as a service for participatory sensing

Heng Zhang, Nawanol Theera-Ampornpunt, He Wang, Saurabh Bagchi, Rajesh K. Panta (AT&T Labs)
School of Electrical and Computer Engineering, Purdue University

What is mobile crowdsensing (MCS)?
- Mobile crowdsensing is a new way to distribute computing and sensing workload to mobile devices.
- Participatory crowdsensing
  - Need user interaction with devices
  - Opportunistic crowdsensing
  - Automatic sensing, collecting and sharing

Why mobile crowdsensing is getting popular
- Ubiquitous use of smartphones
- Sophisticated sensors embedded
- Accuracy and real-time delivery
- Lower cost than many dedicated infrastructures
- Fast network support

Current MCS has problem encouraging more piggyback opportunity
- Predict user’s app usage
  - Some devices at the same location are large redundancy
- Energy costly and implement.

Intuitive and easy to remote servers.

User’s incentive
- Current MCS has problem encouraging more participants
  - One concern is energy drain
  - Second concern is privacy
- Most users would allow 25% energy drain for MCS activities
- But energy cost is higher in today’s MCS solutions for most MCS tasks

What We Built: A mobile crowdsensing framework

Problem Statement
- Current approaches to mobile crowdsensing activities need to be improved with respect to energy consumed and fairness
  - Mobile crowdsensing client: High energy consumption.
  - Mobile crowdsensing server: High overhead and low task distribution efficiency;
  - Need a general mobile crowdsensing framework to schedule mobile crowdsensing tasks as well as pick mobile crowdsensing participants.

Solution Approach
- Sense-Aid is a distributed software framework for mobile crowdsensing
  - Embedded at the edge of 4G LTE network
    - Fetch device information from LTE Core network
    - Offload mobile traffic to Sense-Aid Server
    - Schedule mobile crowdsensing tasks
  - Approach
    1. Orchestrate among many devices
    2. Piggyback mobile crowdsensing data packets
    3. Utilize cellular radio state opportunity

Our Contributions
- Propose a framework for developing crowdsensing applications
  - Saving device as well as energy for all mobile crowdsensing devices
  - Fairly choose crowdsensing devices
  - Deploy Sense-Aid server in 4G LTE network edge
  - Have a global view of crowdsensing devices’ locations and cellular radio state
  - Evaluated by a user study to compare with baseline periodic sensing and state-of-art piggyback crowdsensing

Cellular Radio state
- Cellular radio state transition is energy costly
  - State promotion is unnecessary to just send a few hundred bytes of crowdsensing data packet.
  - Cellular tail has no data transmission between device and cellular tower.
  - Tail timer is reset if client sends or receives data in tail time.

Design
- Crowdsensing Application Server
- Aid Server
- Crowd Sensing Client
- Task Scheduler
- Device Database
- Task Distributor

Energy: Fairness
- Sense-Aid Basic
  - Reset tail timer if data transmitted in tail time.
- Sense-Aid Complete
  - Not set tail timer

Periodic and Piggyback crowdsensing
- Periodically sample sensors and send sensor values to remote servers
- Intuitive and easy to implement.
- Energy costly and large redundancy (some devices at the same location are scheduled the same tasks)

Piggyback crowdsensing
- Piggyback crowdsensing data packet onto regular traffic
- Predict user’s app usage pattern to determine piggyback opportunity

Result: Energy Saving
- Sense-Aid out-performs Periodic and PCS in all scenarios
- Sense-Aid shows good scalability with respect to concurrent MCS tasks running on one device.
- Even with near perfect prediction of PCS, Sense-Aid still saves more energy than PCS.

Acknowledgments
This work was supported by NSF grant CNS-1409506 and AT&T. The views expressed represent those of the authors and do not necessarily reflect the views of the sponsoring agency.

Reference