Optimizing Mobile Middleware for Coordinated Sensor Activations

Ting-Fang Hou

Motivation

- Increasingly more context-aware applications (apps) leverage the rich set of sensors on the smartphones.
- These applications directly control sensors which lead to redundant activations and energy waste

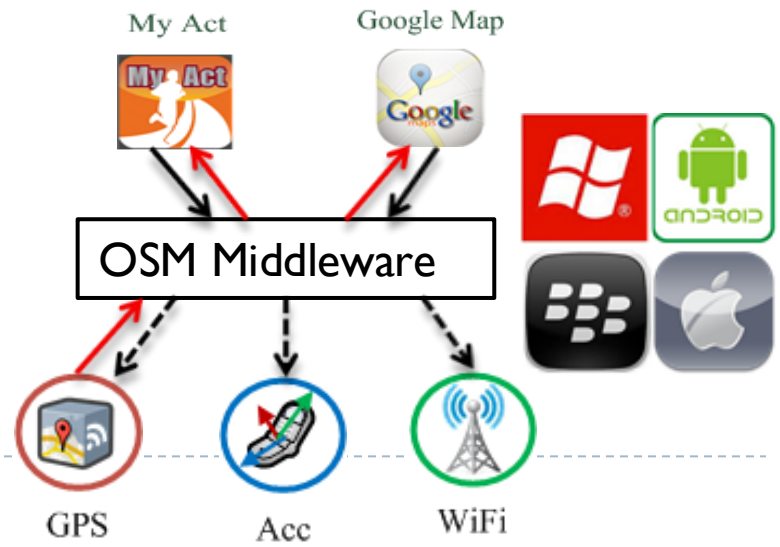
Introduction

- We can find some examples in our lives:
  - Case1: Different apps may require same contexts
  - Case2: Use different sensors to sense same contexts

➢ How to select the most efficient sensing strategy ?

– Satisfy all apps' requirements

– Minimize energy consumption



Contributions

- We achieve **coordinated** and **optimized** usages of sensors
- For a single smartphone[middleware14 under review]:
  - Consider the tradeoff between **energy** and **accuracy**
  - Present the **middleware** with scheduling algorithms
  - Propose four **optimal/efficient** algorithms
  - Rigorously solve the scheduling problem with sensory data caches
- For multiple smartphones[PIMRC14 under review]:
  - Design, implement, and evaluate a **crowdsensing system**
  - Consider the tradeoff between **carbon footprint (cost of traveling and sensing)** and **complete ratio of tasks**
  - Propose two **optimal/efficient** algorithms

Previous work

- Chun-Lin Lin's work [1]
  - Solved a similar but simpler problem
  1. Proposed two heuristic algorithms
  2. Implemented the algorithms on smartphones
- Improvement
  1. Mathematically formulate two scheduling problems
  2. Improve and extend his algorithms
  3. Leverage the relationship between the frequency of requests and sensor sampling rates
  4. Develop a simulator to do larger experiments
  5. Apply the sensor scheduling algorithms to multiple smartphones

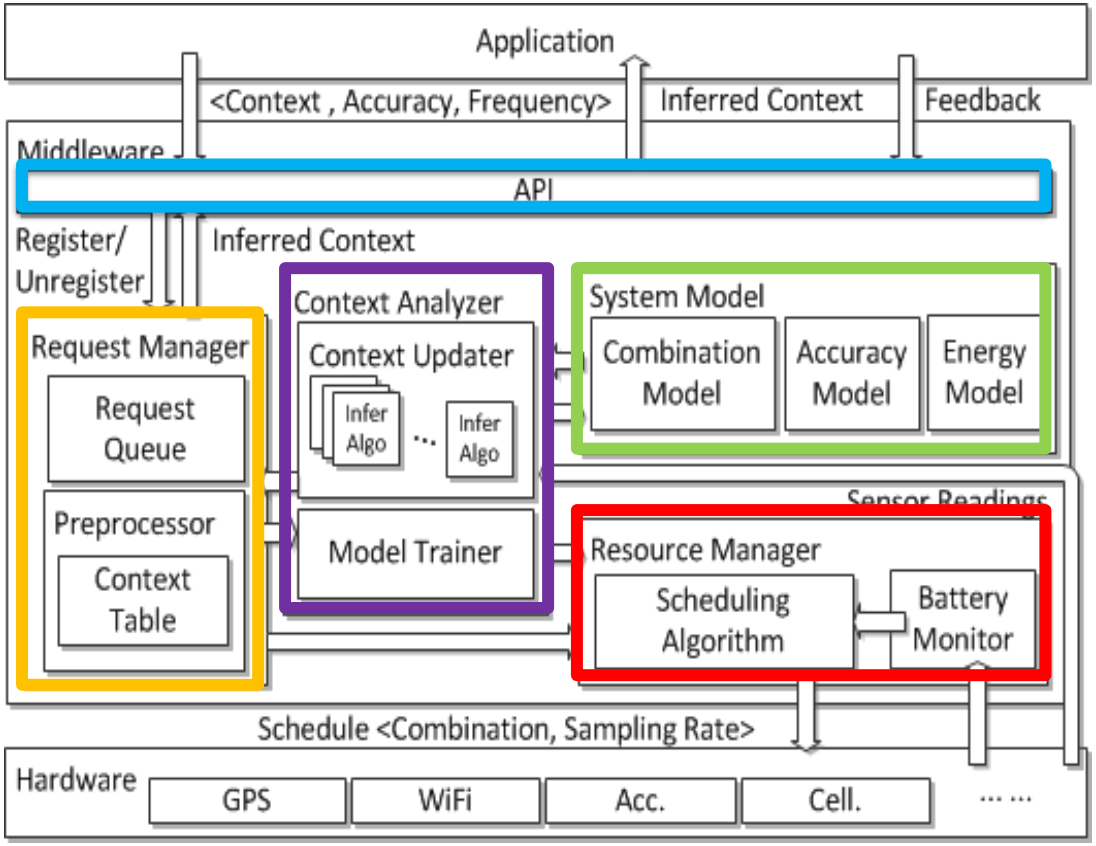
[1] C.-L. Lin. An Energy/Accuracy-Optimized Framework for Context Sensing on Smartphones. Master's thesis, National Tsing Hua University, Taiwan, 2013

System Overview

- We proposed an Optimal Sensor Management (OSM) middleware
- The OSM middleware sits between apps and the hardware
- OSM middleware :
  - Provides API to connect apps
  - Maintains a database of active requests
  - Determines which sensors should be activated

System Architecture

Joint with C. Lin



API:
1. Register()/Unregister()
2. Feedback()

Request Manager
1. Manages a Request Queue
2. Preprocess the requests

Context Analyzer
1. Context Updater
2. Model Trainer

Sensory Data
Infer Algo
Context

System Model
Combination/Accuracy/Energy

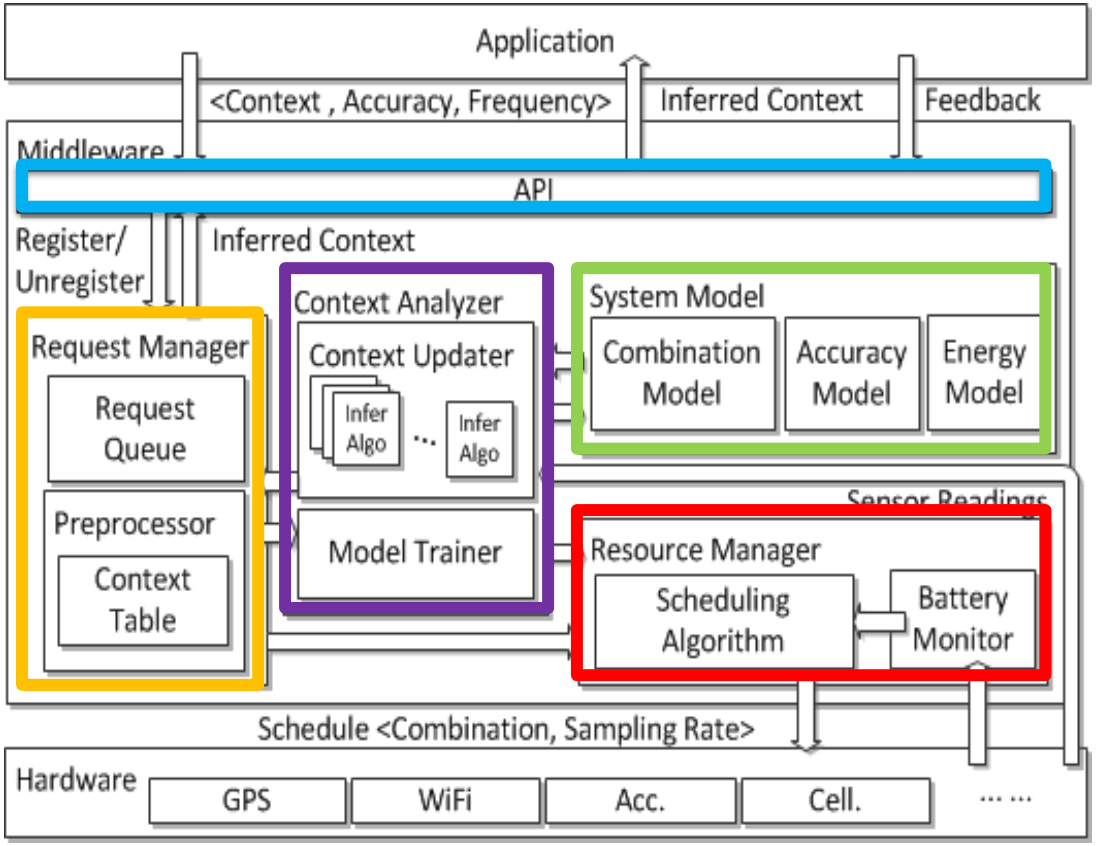
Resource Manager
1. Battery Monitor
2. **Scheduling Algorithm**

System Model

- **Combination model**
  - The sensor combination of each inference algorithm
- **Accuracy model**
  - The precision (accuracy) of each inference algorithm
- **Energy model**
  - The energy consumption of each sensor
- Example:
  - IsMeeting { <Acc. Mic. Wifi.> < 80% > <265mW> }

System Architecture

Joint with C. Lin



API:
1. Register()/Unregister()
2. Feedback()

Request Manager
1. Manages a Request Queue
2. Preprocess the requests

Context Analyzer
1. Context Updater
2. Model Trainer

System Model
Combination/Accuracy/Energy

Resource Manager
1. Battery Monitor
2. **Scheduling Algorithm**

*Coordinated and efficient sensor usage !

*Avoid redundant energy waste !

Scheduling Problem

Joint with C. Lin

- Support two optimization criteria:
  - **Energy Minimization (EM):**

    Minimize the energy consumption

    Satisfy all the apps' requirements
  - **Accuracy Maximization (AM):**

    Maximize the overall accuracy within an energy budget

Problem Formulation

- Decision variable: $x_s \in \{0, 1\}$
  x_s indicates whether the sensor s should be activated
- Energy Minimization :

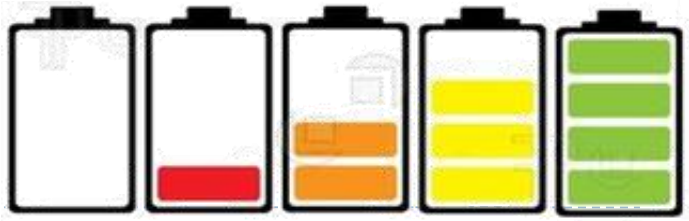
$$\min \sum_{s=1}^{S} e_s x_s$$

Minimize energy

$$s.t. \max_{c=1}^{C} \left\{ \left\lfloor \frac{\sum_{s=1}^{S} m_{c,s} x_s}{\sum_{s=1}^{S} m_{c,s}} \right\rfloor a_{c,r} \right\} \geq y_r, \forall r = 1, 2, \ldots, R;$$

$$x_s \in \{0, 1\}, \forall s = 1, 2, \ldots, S.$$

Satisfy all requirements



Problem Formulation

- Decision variable: $x_s \in \{0, 1\}$
  x_s indicates whether the sensor s should be activated
- Accuracy Maximization :

$$\max \sum_{r=1}^{R} \max_{c=1}^{C} \left\{ \left\lfloor \frac{\sum_{s=1}^{S} m_{c,s} x_s}{\sum_{s=1}^{S} m_{c,s}} \right\rfloor a_{c,r} \right\}$$

Maximize accuracy

$$s.t. \sum_{s=1}^{S} e_s x_s \leq E;$$

Energy budget

$$x_s \in \{0, 1\}, \forall s = 1, 2, \ldots, S.$$

Proposed Scheduling Algorithms

- Optimal algorithms :
  - The formulations are “Integer Programming Problem”
  - We use commercial optimization CPLEX solver which is developed by IBM

  Energy Minimization Algorithm (EMA)
  Accuracy Maximization Algorithm (AMA)

  * Good performance * Small scale case

- Efficient algorithms :

  Efficient Energy Minimization Algorithm (EEMA)
  Efficient Accuracy Maximization Algorithm (EAMA)

  * Less running time * Designed for smartphones

Efficient Energy Minimization Algorithm (EEMA)

Joint with C. Lin

- EEMA is inspired by the Set Cover Problem
- Define the utility function g_c :

  => The combination c has higher g_c means it satisfies more requests and less costs

---

1: // Input: A, M, $\hat{\mathrm{R}}$; Output: X
2: let $\mathrm{X} = \varnothing$
3: while $\hat{\mathrm{R}} \neq \varnothing$, $\max_{1 \le c \le C} g_c(\mathrm{A}, \mathrm{M}, \mathrm{X}, \hat{\mathrm{R}}) > 0$ do
4: for each $c = 1, 2, \ldots, C$ do
5: compute $g_c(\mathrm{A}, \mathrm{M}, \mathrm{X}, \hat{\mathrm{R}})$ using Eq. (13)
6: select $c^* = \operatorname{argmax}_{c=1,2,\ldots,C} g_c(\mathrm{A}, \mathrm{M}, \mathrm{X}, \hat{\mathrm{R}})$
7: $\mathrm{X} \leftarrow \mathrm{X} \cup \{s | m_{c^*,s} = 1\}$
8: $\hat{\mathrm{R}} \leftarrow \hat{\mathrm{R}} - \{r | y_r \le a_{c^*,r}\}$

Calculate and Update the g_c

Choose the most efficiency combination

Efficient Accuracy Maximization Algorithm (EAMA)

Joint with C. Lin

- EAMA is inspired by the 0/1 knapsack Problem
- Redefine the utility function : g'_c
  => The combination c has higher g'_c means it provides the higher accuracy and less costs

Calculate and Update the g'_c

Choose the most efficiency combination

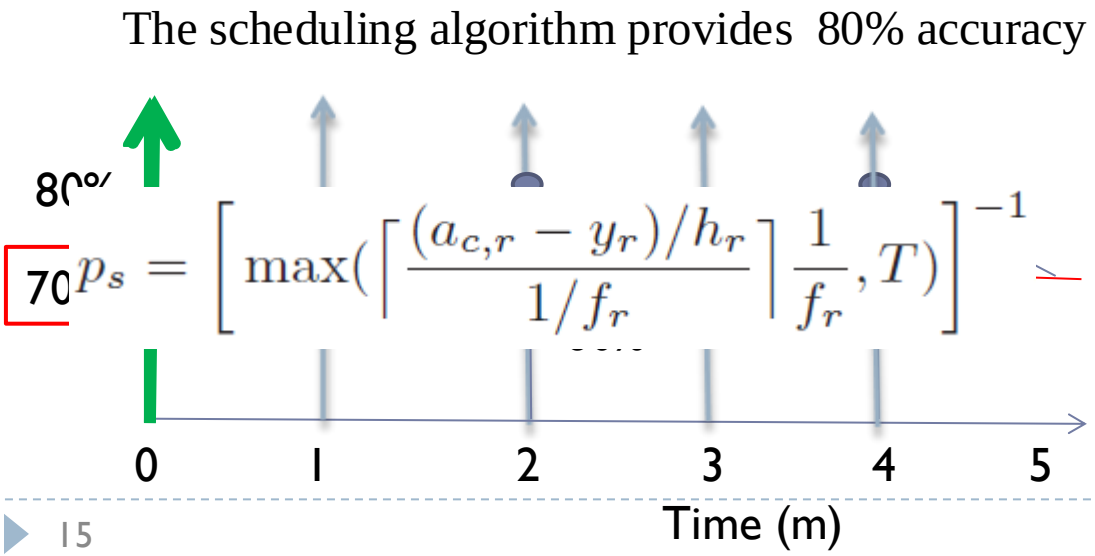
Check the energy budget

```
1: // Input: A, M; Output: X
2: let e_X = 0, W = {1, 2, ..., C}, Ŷ = 0
3: while W ≠ ∅ do
4:     for each c = 1, 2, ..., C do
5:         compute g'_c(A, M, X, Ŷ) with Eq. (15)
6:     select c* = argmax_{c∈W} g'_c(A, M, X, Ŷ)
7:     W ← W − {c*}
8:     if e_X + w_{c*}(M, X) < E then
9:         X ← X ∪ {s | m_{c*,s} = 1}
10:        e_X = Σ_{s=1}^{S} e_s x_s
11:        for each r = 1, 2, ..., R do
12:            if ŷ_r ≤ a_{c*,r} then
13:                ŷ_r = a_{c*,r}
```

Heterogeneous Frequencies/Sampling Rates

- Extend the EEMA and the EAMA with heterogeneous frequencies/sampling rates as EEMA* and EAMA*
- h_r : the accuracy degradation rate of the context r

Ex: The context r requires 70 % accuracy in every minute
, and the accuracy which decreases 10% per minute



*Efficiently cache the sensory data !

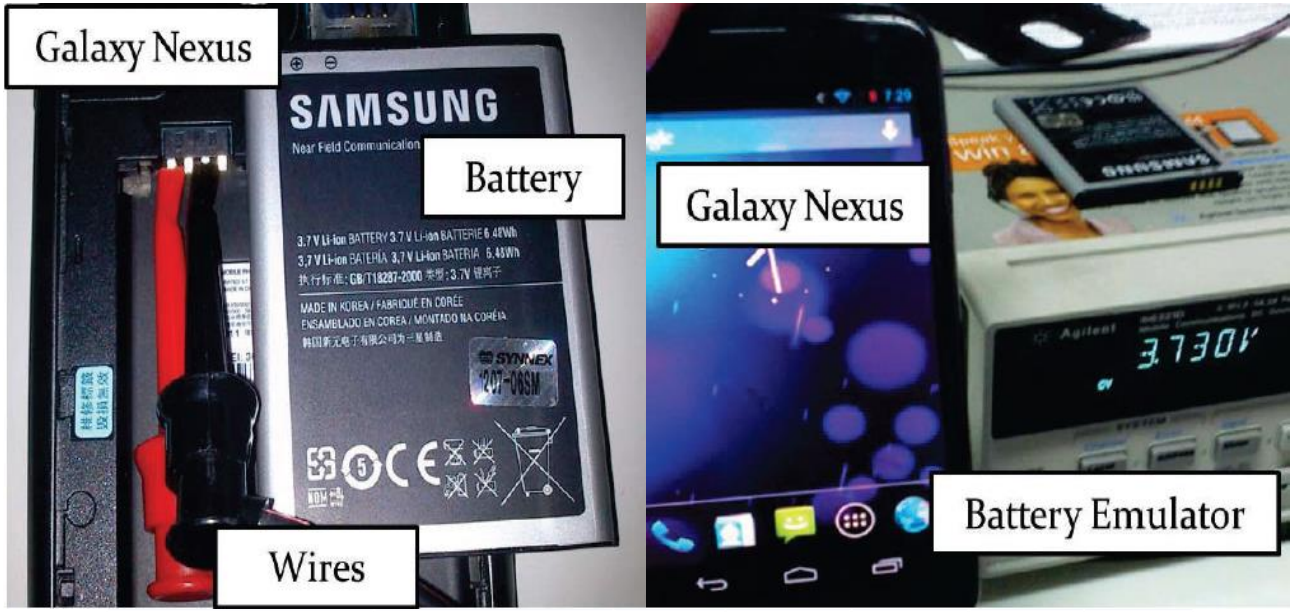
*Reduce energy consumption !

Simulation

- Developed an event-driven simulator in Java
- Baseline algorithm :
  - Selects sensors with the highest accuracy for each context
- Compare scheduling algorithms :
  - Optimal : EMA / AMA
  - Efficient : EEMA / EAMA
  - With frequencies/sampling rates : EEMA* / EAMA*
  - Baseline

Data Collection

1. Collect active apps in the Android activity stack from 5 users in three weeks
2. Collect inference algorithms (from existing papers)
3. Create the energy model



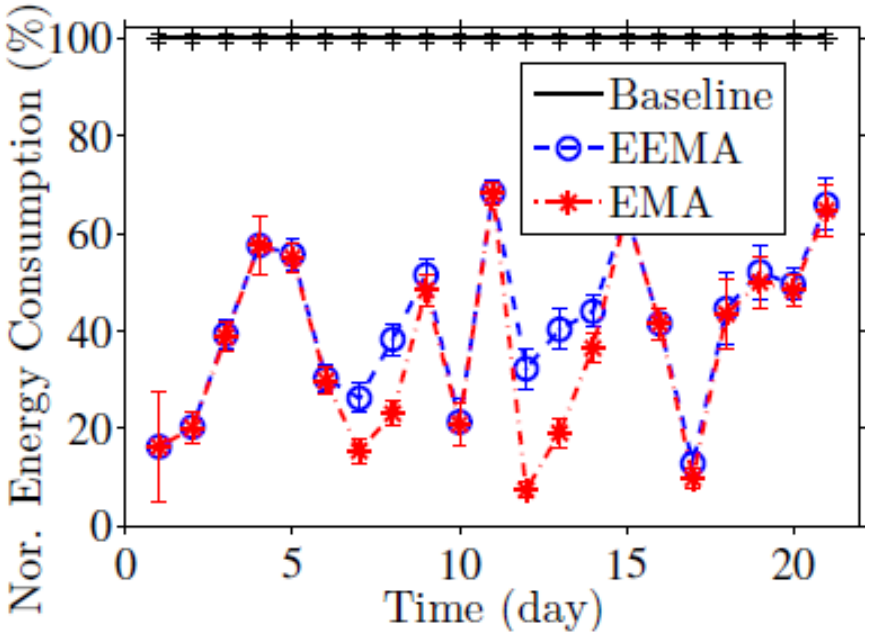
Energy	(mW)
Acc.	187
Blue.	195
WiFi	16
Mic.	62
GPS	546
Cell.	20

Parameter

- Set a fixed scheduling time T = 5min
- Set E as the **Energy budget** in a scheduling time
  - E= {45, 52.5, 60, 67.5, 75}J

Energy Saving in EM

- The average energy consumption with 5 users in 21 days



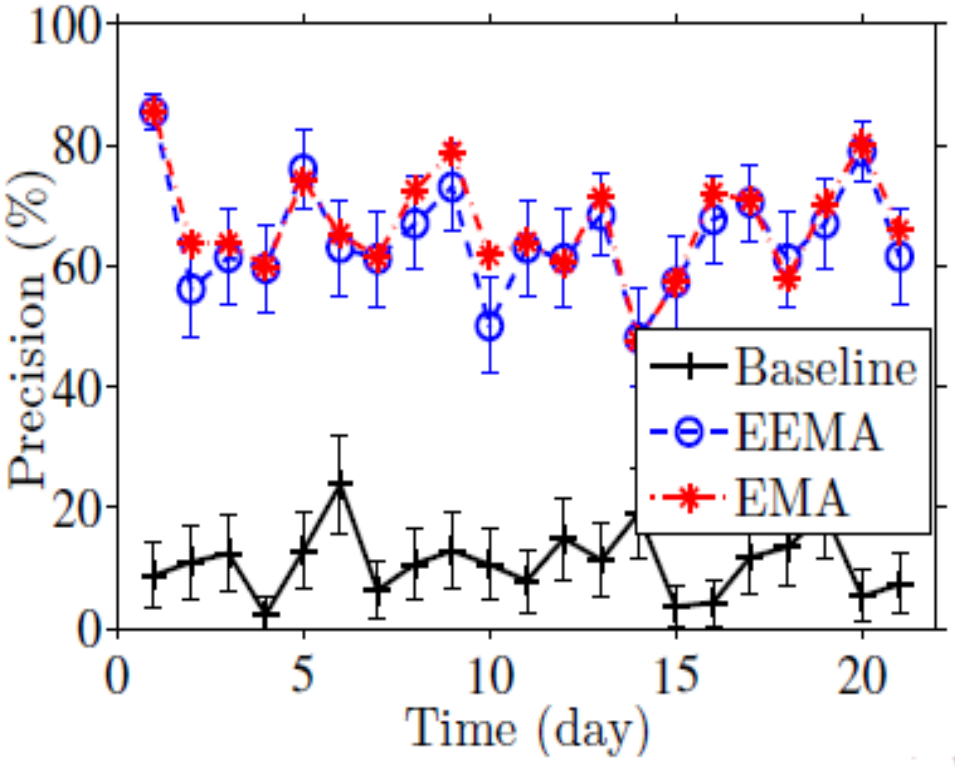
* Saves at least 30%
* EEMA achieves a small gap with EMA

* EMA terminates in 50ms and EEMA terminates in 1ms

Improvement of Accuracy in EM

- The precision with 5 users in 21 days

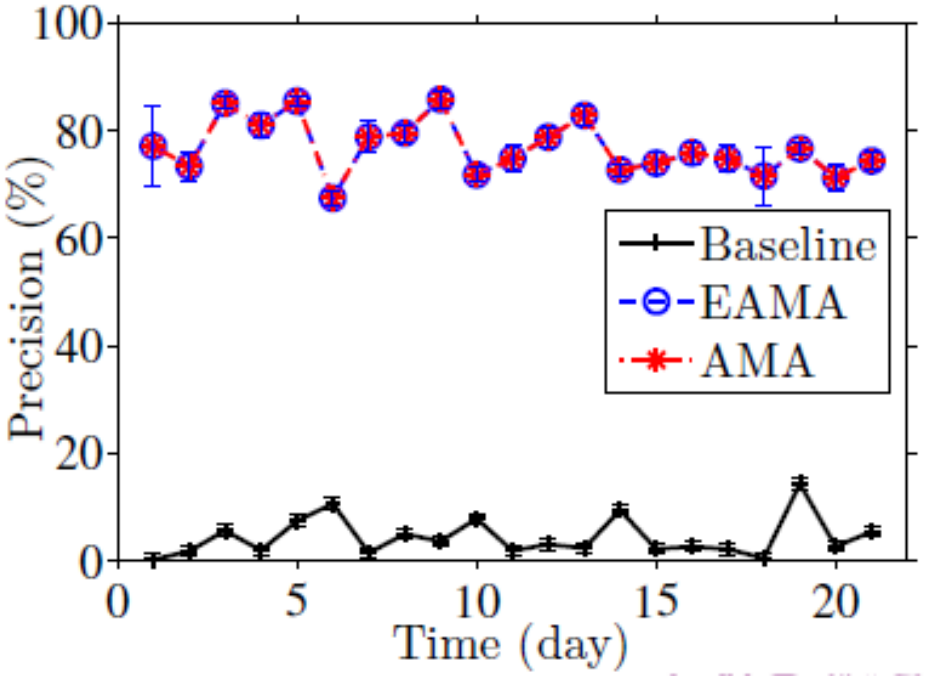
* EEMA achieves at least 30% higher accuracy
* EEMA achieves a small gap with EMA



Accuracy Improvement in AM

- The average accuracy with 5 users in 21 days
  E= 52.5J

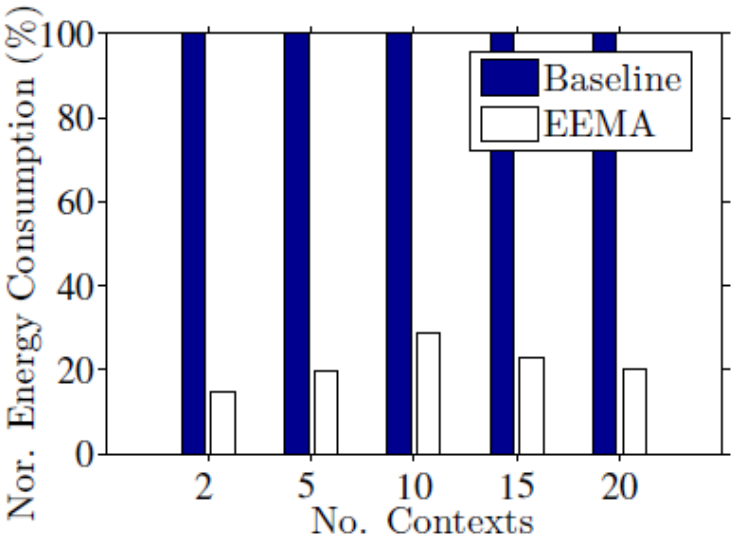
* Accuracy is 72.38% higher than the baseline
* EAMA achieves a ~0.1% gap with AMA

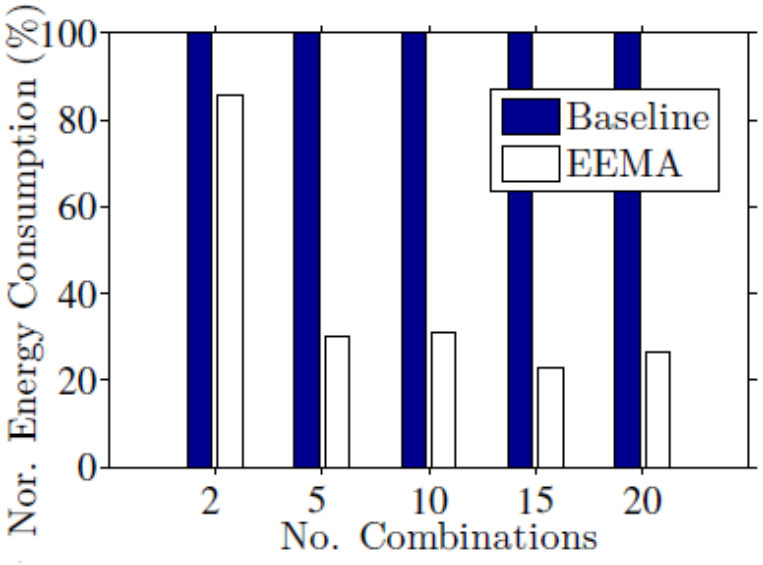


* AMA terminates in 5000ms and EAMA terminates in 1ms

Energy Saving with Variant No. Contexts/Combinations in EM

- Comparison of the average energy saving between EEMA and the baseline
- E = 60 J



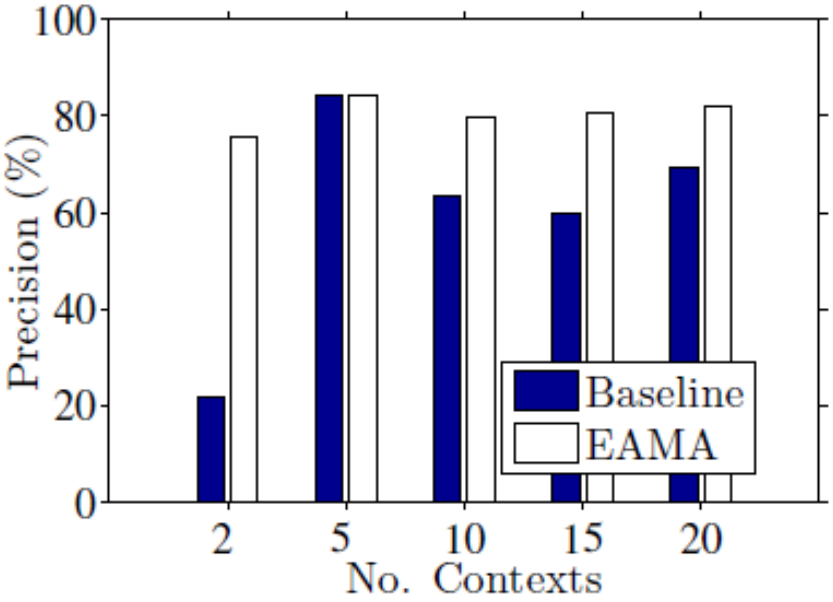


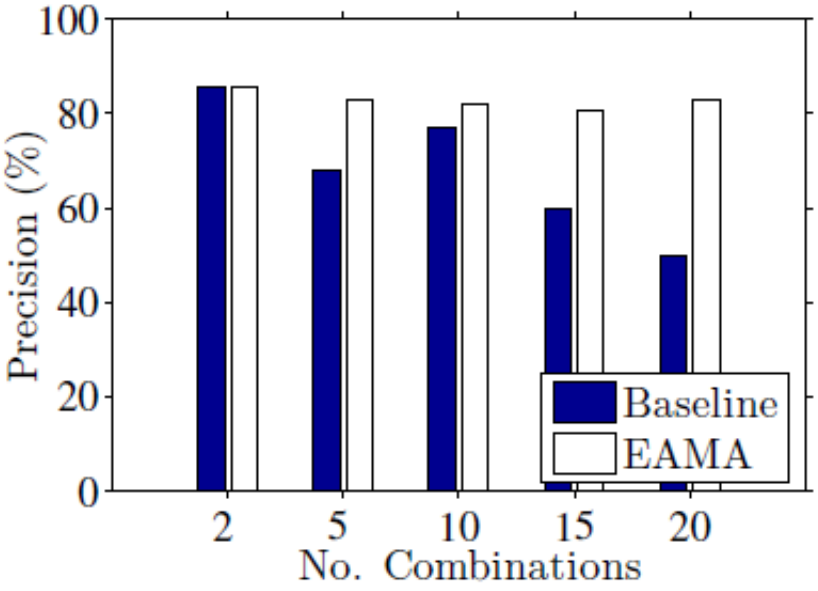
* EEMA saves at least 71.03% in energy

* EEMA saves more energy when combinations increase

Accuracy Improvement with Variant No. Contexts/Combinations in AM

- Comparison of the average energy saving between EAMA and the baseline
- E = 60 J





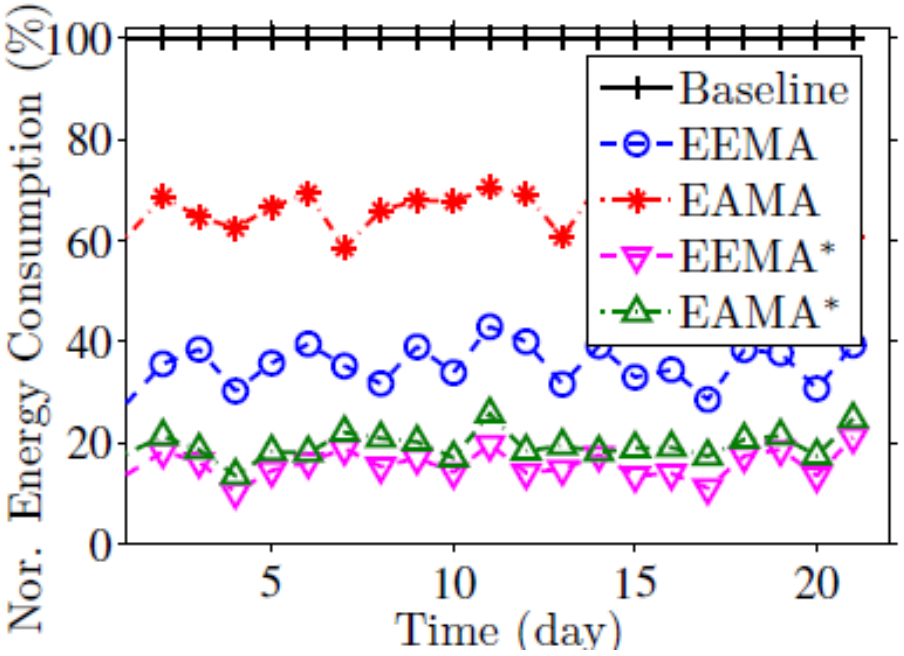
* EAMA achieves at most 53.71% in the precision

* EAMA achieves less when the problem is small

Energy Saving with Heterogeneous Frequencies/Sampling Rates

* EEMA saves 64.58%
  EEMA* saves 84.66%

* EAMA saves 33.68%
  EAMA* saves 80.48%



* Higher performance in EAMA*
  => high precision which is achieved by EAMA

Implementation

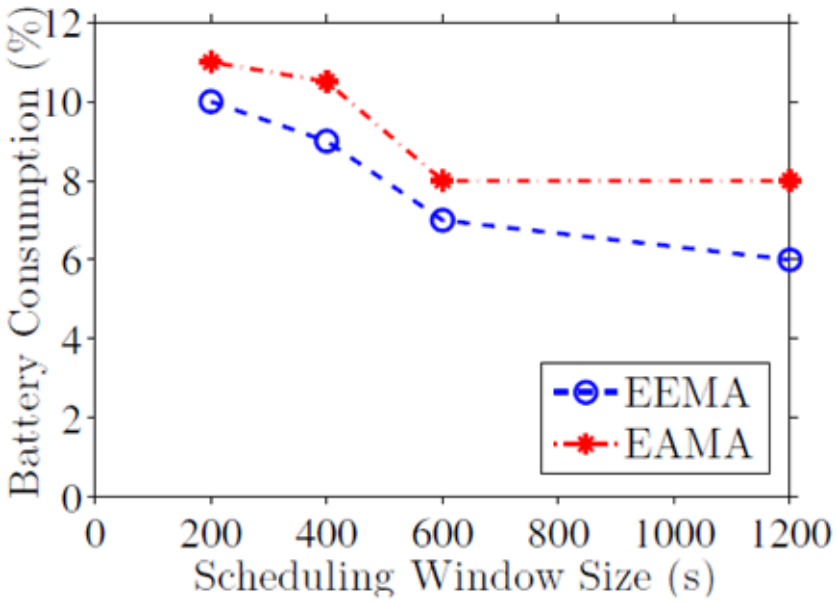
Joint with C. Lin

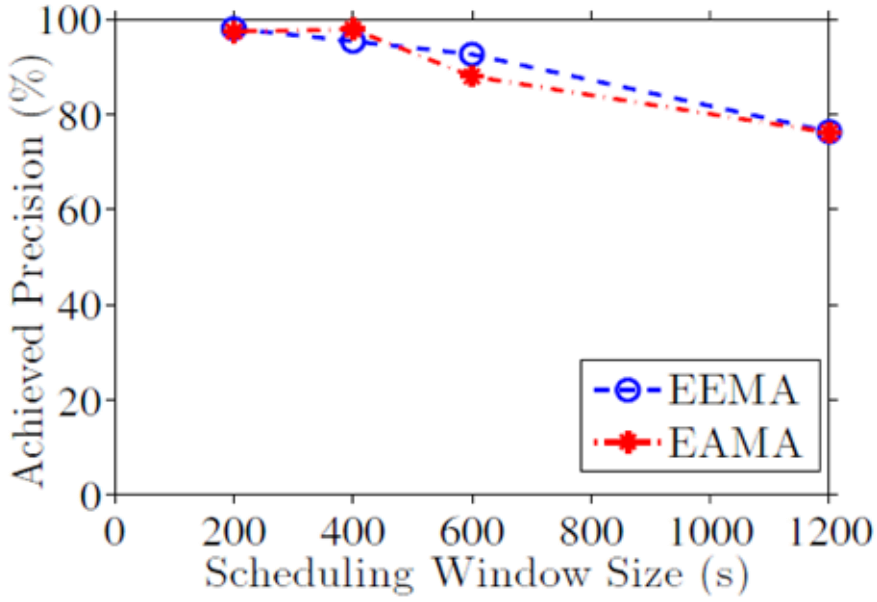
- Implement OSM with two efficient algorithms on the Android system

* EEMA :
  Prolongs battery life two times
  Achieves accuracy : 93.94%
* EAMA:
  Prolongs battery life 1.5 times
  Achieves accuracy : 94.85%

* Execution times :
  EEMA and EAMA at most 306 ms and 503 ms

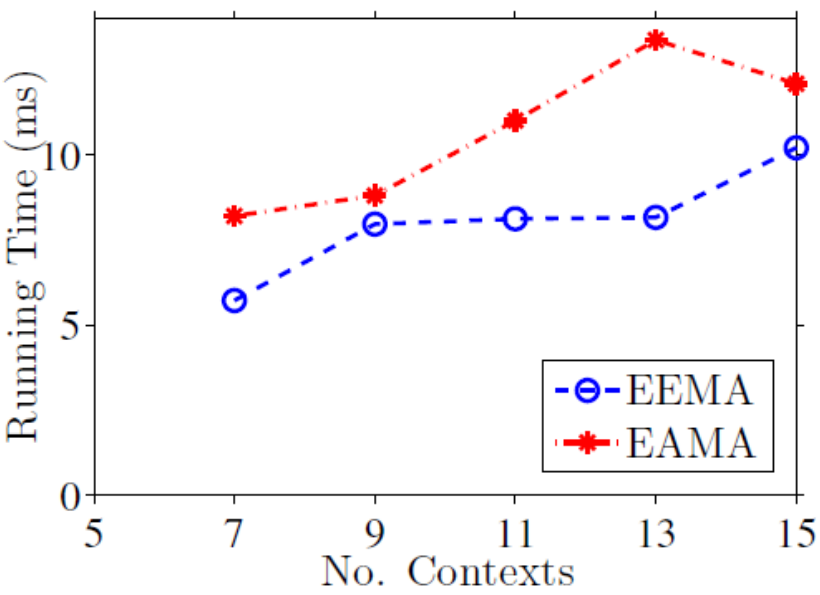
Benefit of Variant Scheduling Window Sizes

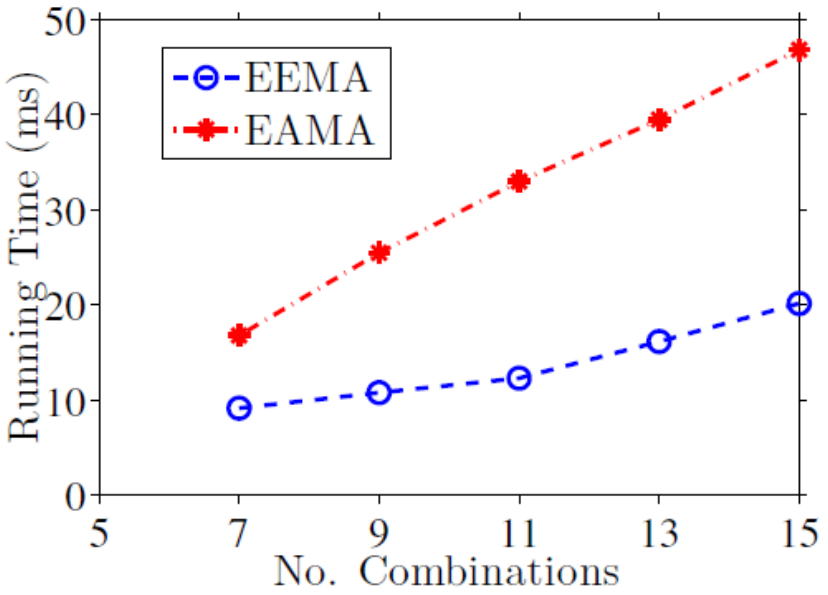




* When the size increases, EEMA and EAMA save more energy but expense of lower precision
* EEMA constantly saves more energy compared to EAMA
* EEMA and EAMA achieve very high precision

Benefit of Running Time with Variant No. Contexts/Combinations





* While the contexts and combinations increase
  1. The running time increases
  2. EEMA and EAMA algorithms are always efficient

Sensor Scheduling for Multiple Devices

- Smartphones are widespread and equipped with rich sensors

⇒ **Extend the sensor scheduling for multiple smartphones**

- Apply to the "**Crowdsensing system**"
  - Utilize the strength of smartphone users
  - Cooperate sensors on multiple smartphones and infrastructure sensors

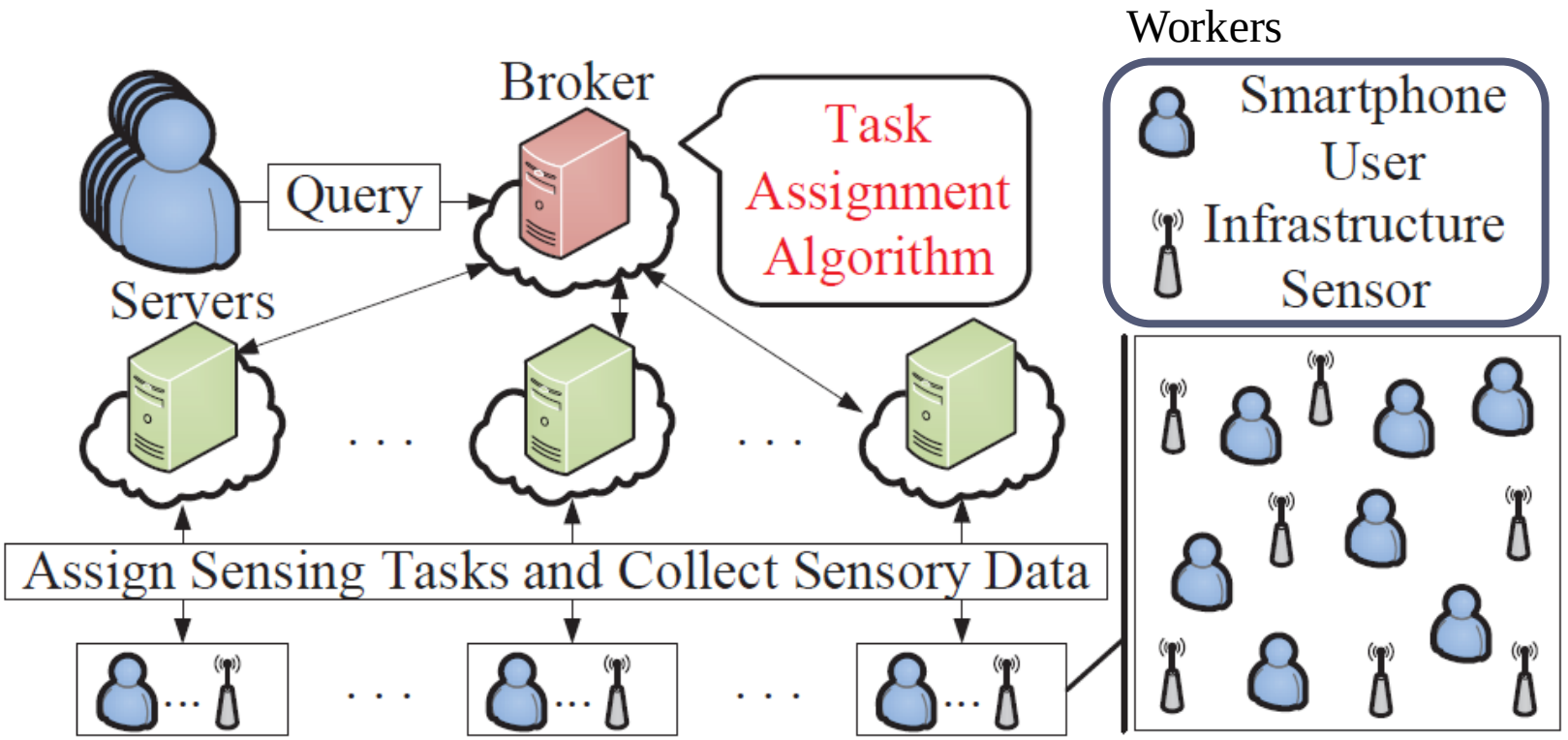
Crowdsensing System

- Scenario:
  Task: Which one facility is the most popular in the amusement park?
  Context : Is the facility Crowded ?

but it may be hard finished by a worker who is far away from the amusement park or whose smartphone is lack of energy

- How to assign tasks in the most efficient strategy?
  - Satisfy all tasks' requirements
  - Minimize the cost of carbon footprints

Crowdsensing System Overview



Task Assignment Problem

- Task Assignment Problem has two parts:
  (i) Worker selection problem
  (ii) Sensor scheduling problem

- Consider workers' locations and abilities (ex. The battery level of smartphones)
  1. Schedule tasks to workers with working paths
  2. Achieve tasks' requirement
  3. Minimize the cost of carbon footprints
     -> which consists of sensing and traveling costs

Problem Formulation (1/2)

- Decision variables: $x_{w,l,s} \in Z,\ E_{i,j}^{w} \in \{0, 1\}$

$x_{w,l,s}$ indicates how many times that worker w turns on the sensor s at the location l

$E_{i,j}^{w}$ indicates whether the worker w moves from the location i to the location j

- Objective :

Sensing energy

Traveling cost (distance)

$$min\{\alpha \sum_{w \in \mathbf{W}} \sum_{l \in \mathbf{L}} \sum_{s \in \mathbf{S}} e_{w,s} x_{w,l,s} T_s + \beta \sum_{w \in \mathbf{W}^*} \sum_{l,l' \in \mathbf{L}} d_{l,l'} E_{l,l'}^{w}\}$$

Minimize the total carbon footprints

- Task Constraints :

$$Q(\tau_r, a_{r_c}) \geq f_r, \forall r \in \mathbf{R}$$

Ensure tasks are satisfied

$$F_w - \sum_{l \in \mathbf{L}} \sum_{s \in \mathbf{S}} e_{w,s} x_{w,l,s} \geq \theta_w, \forall w \in \mathbf{W}^*$$

Energy threshold

*Carbon footprint transform coefficients : α, β

Problem Formulation (2/2)

- Path Constraints:
  - Start location:

In-degree: $$\sum_{j \in \mathbf{L}} E^{w}_{j,A_w} = 0, \forall w \in \mathbf{W}^*$$

Out-degree: $$\sum_{j \in \mathbf{L}} E^{w}_{A_w,j} = \left\lceil \frac{\sum_{j \in \mathbf{L}} \sum_{s \in \mathbf{S}} \left\lceil \frac{x_{w,j,s}}{x_{w,j,s}+1} \right\rceil}{|\mathbf{L}||\mathbf{S}|+1} \right\rceil, \forall w \in \mathbf{W}^*$$

  - Other locations: Visit locations which have sensing tasks

In-degree: $$\sum_{j \in \mathbf{L} \cup A_w, j \neq i} E^{w}_{j,i} = \left\lceil \frac{\sum_{s \in \mathbf{S}} \left\lceil \frac{x_{w,i,s}}{x_{w,i,s}+1} \right\rceil}{|\mathbf{S}|+1} \right\rceil, \forall i \in \mathbf{L}, \forall w \in \mathbf{W}^*$$

Out-degree: $$\sum_{j \in \mathbf{L}, j \neq i} E^{w}_{i,j} \leq \left\lceil \frac{\sum_{s \in \mathbf{S}} \left\lceil \frac{x_{w,i,s}}{x_{w,i,s}+1} \right\rceil}{|\mathbf{S}|+1} \right\rceil, \forall i \in \mathbf{L}, \forall w \in \mathbf{W}^*$$

Efficient Task Assignment Algorithm (ETA)

- ETA is inspired by the Set Cover Problem
- Define the task ratio $\lambda_{w,l}$:

  => The worker w has higher $\lambda_{w,l}$ at the location l means it satisfies more tasks and less costs

Calculate and update the $\lambda_{w,l}$

Choose the most efficient worker

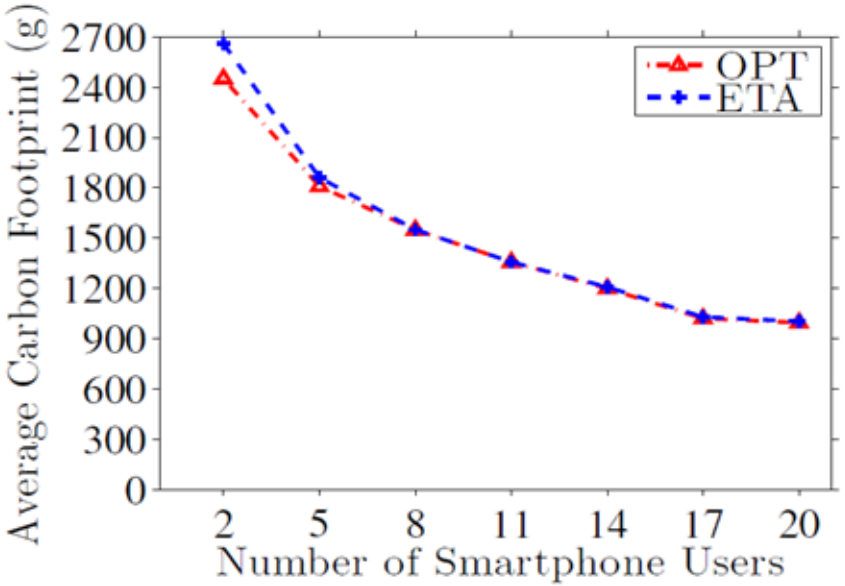
1: Input: $\Phi =< \mathbf{L}, \mathbf{R}, \mathbf{W}, \mu, \widehat{\mathbf{R}} >$
2: Output: $\mathbf{X}, \mathbf{E}$
3: **while** $\widehat{\mathbf{R}} \neq \emptyset$ **do**
4: $\mathbf{z} =$ Sensor_scheduling(Φ)
5: Compute($\mathbf{\Lambda}$) using Eq. (11)
6: Sort($\mathbf{\Lambda}$)
7: $\lambda_{w,l}$ = pop($\mathbf{\Lambda}$)
8: **for** $s \in \mathbf{S}$ **do**
9: $x_{w,l,s} = x_{w,l,s} + z_{w,l,s}$
10: $E^{w}_{A_w,l} = 1$
11: update($\mathbf{E}$, $\mathbf{W}$, μ)
12: **for** $r \in \mathbf{R}$ **do**
13: **if** $Q(\mu_{r_p,r}, a_{r_c}) \geq f_r$ **then**
14: remove r from $\widehat{\mathbf{R}}$
15: **return** $\mathbf{X}, \mathbf{E}$

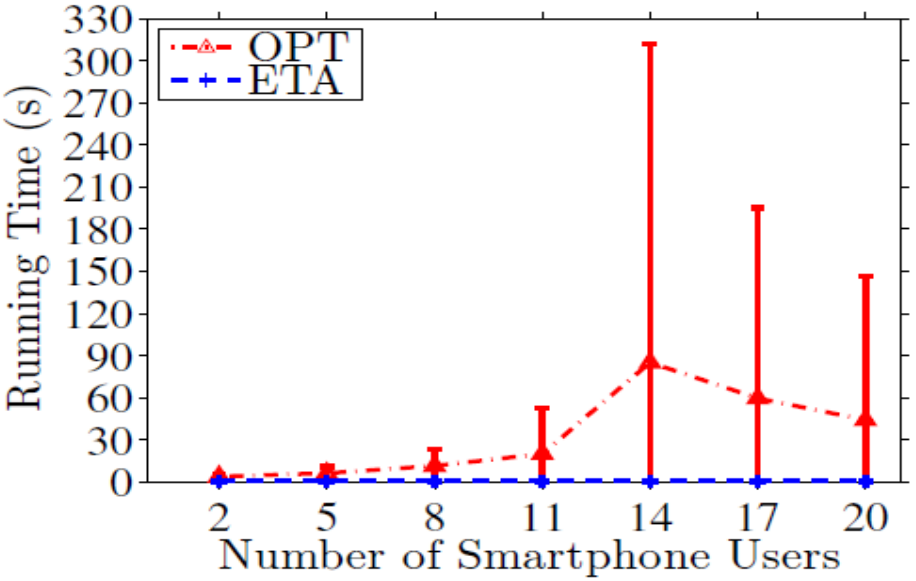
Simulation

- Develop an event-driven simulator in Java
- Baseline
  1. Infrastructure sensors only (IS)
  2. Opportunistic sensing (ISOS):
     - Consider infrastructure' and mobile' sensors
     - Workers move by the random waypoint
- Optimal algorithm (OPT):
  - Developed by the IBM CPLEX solver
- Efficient Task Assignment Algorithm (ETA)

Optimization Gap in Carbon Footprints between ETA and OPT

- Variant numbers of users with 10 queries



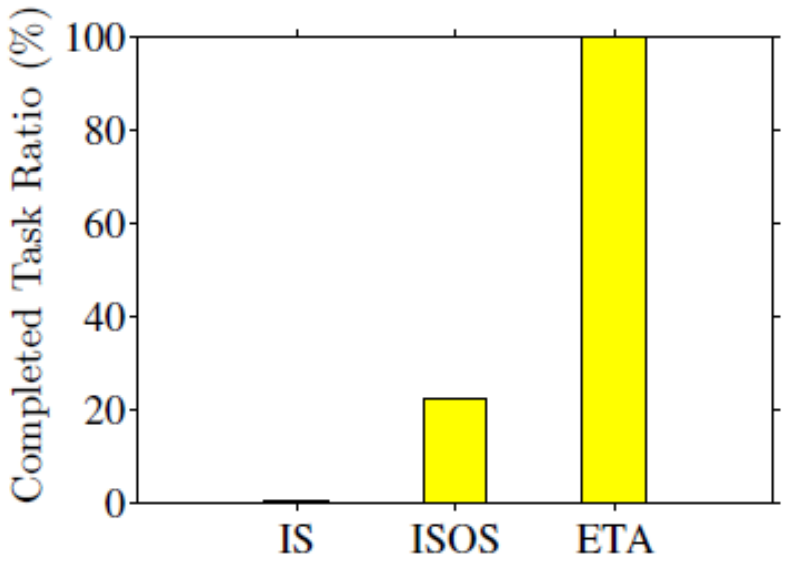


* ETA achieves a small gap of ~2% with OPT

* The running time of ETA is 1333 times faster than OPT

Improve Completed Task Ratio by ETA

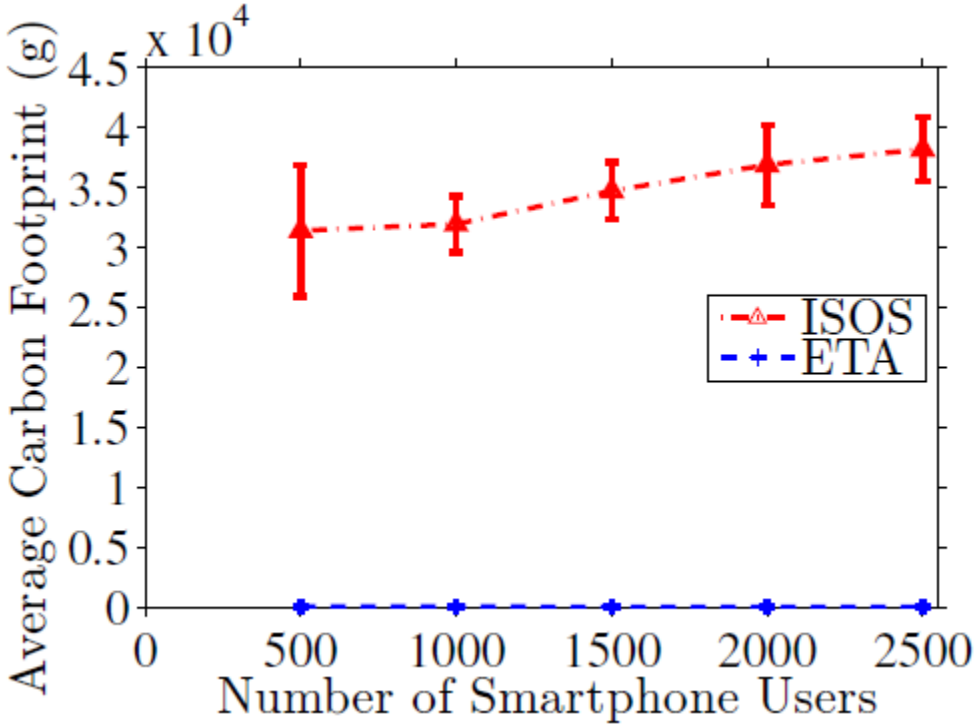
- Comparison of the completed task ratio between ETA, IS, ISOS



* ETA achieves higher completed task ratio than IS and ISOS

Save Carbon Footprints by ETA

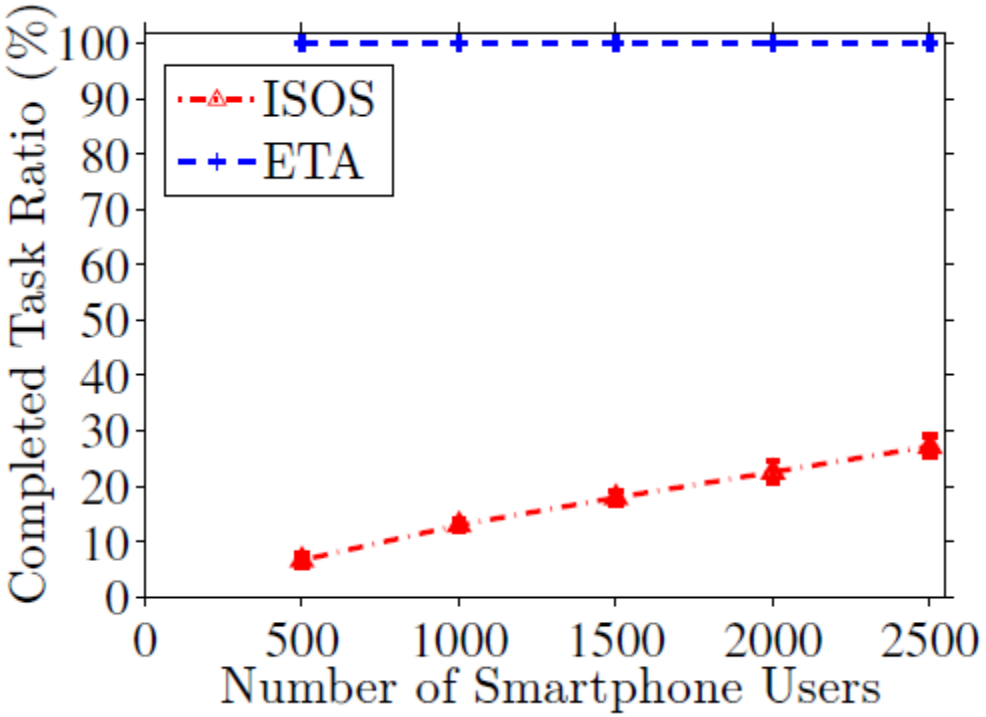
- Comparison between ETA and ISOS in larger scale



* ETA saves 364 times costs more than ISOS

Improve Completed Task Ratio by ETA

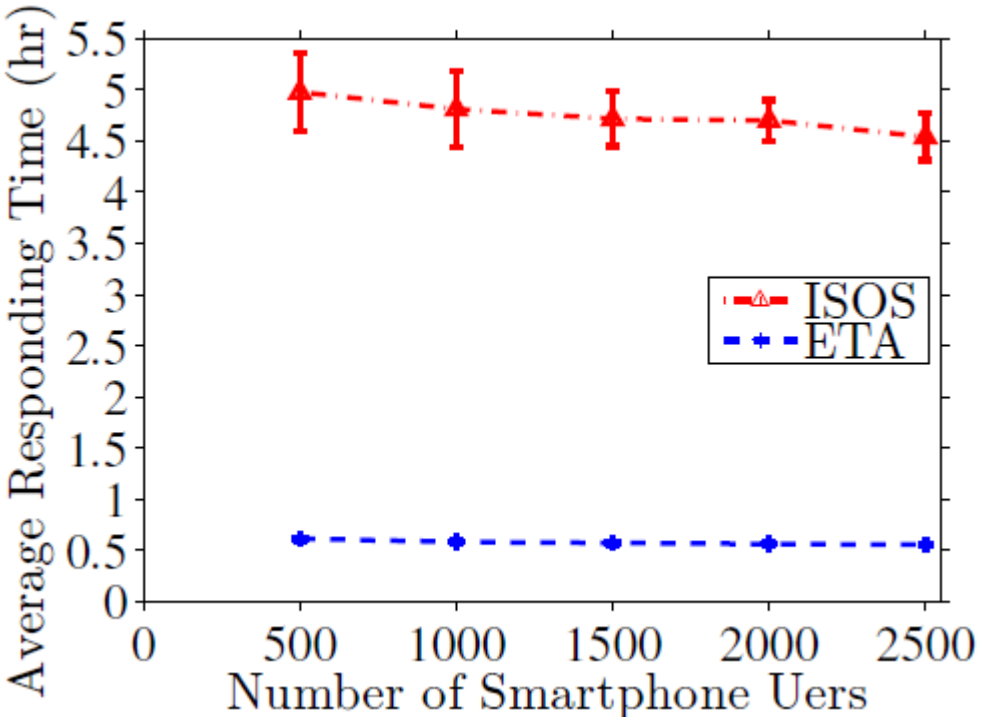
- Comparison between ETA and ISOS



* ETA achieves 5 times ratio higher than ISOS

Improve Responding Time by ETA

- Comparison between ETA and ISOS



* The responding time of ETA is 8 times faster than ISOS

Conclusions

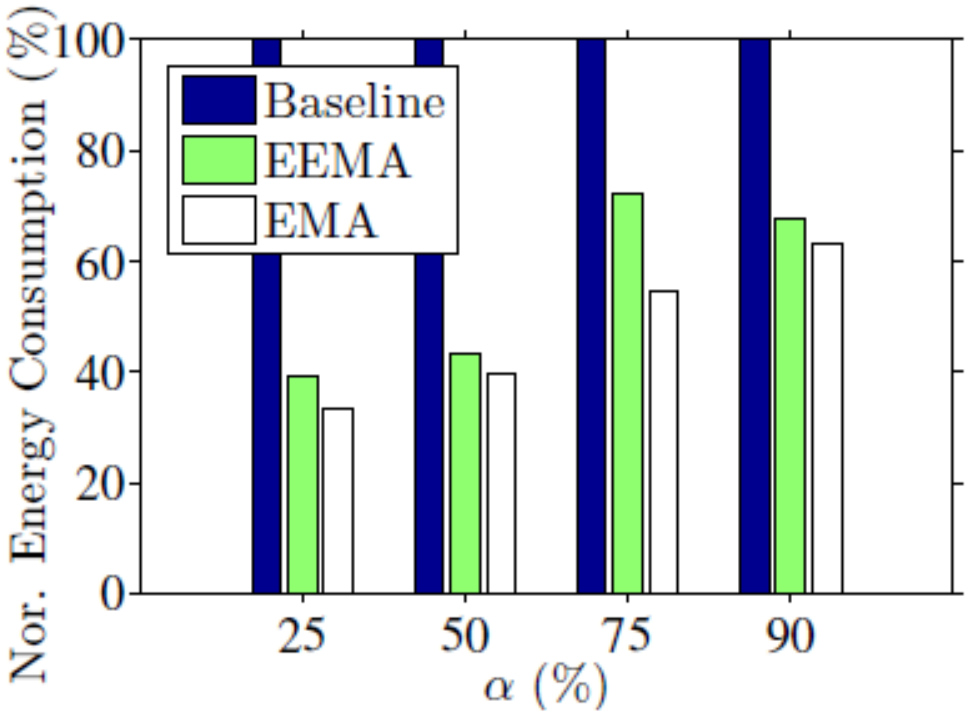
- For sensor scheduling problem :
  - EEMA/ EAMA have a gap as small as ~3% with EMA/AMA
  - EEMA/ EAMA run in real-time
  - EEMA/ EAMA lead to a better performance than the baseline

- For task assignment problem :
  - ETA has a gap as small as ~2% with OPT
  - ETA runs faster than OPT
  - ETA leads to a better performance than ISOS

Backup

Energy Saving with Variant α in EM

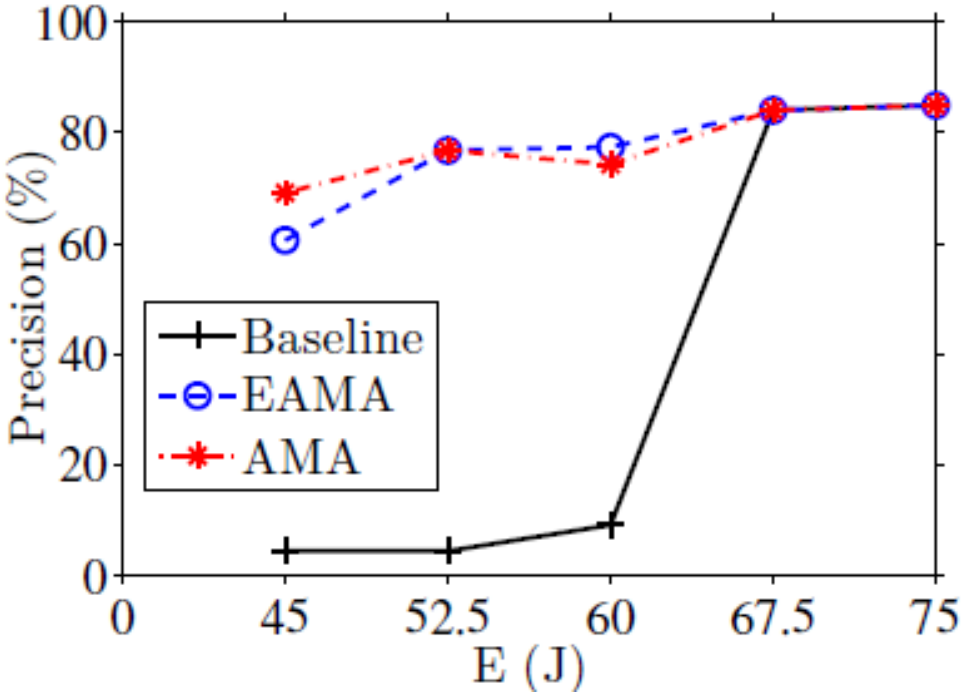
- The average energy consumption which is normalized to the baseline with 5 users in variant α



* When α increases, the gap with the baseline will decrease

Accuracy Improvement with Variant E in AM

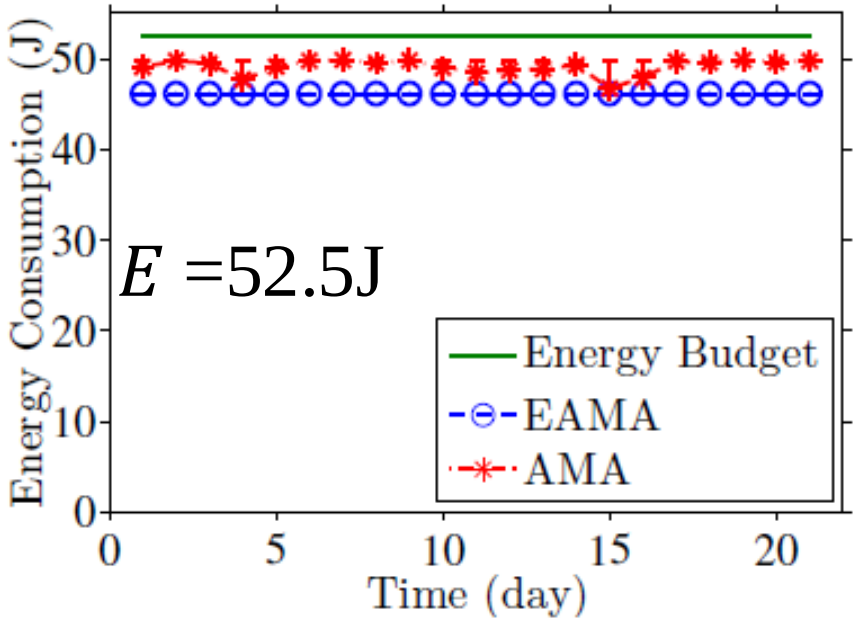
- Consider the average energy consumption with 5 users



* AMA always higher Than the baseline
* EAMA performs well when the energy budget is lower

Energy Budget in AM

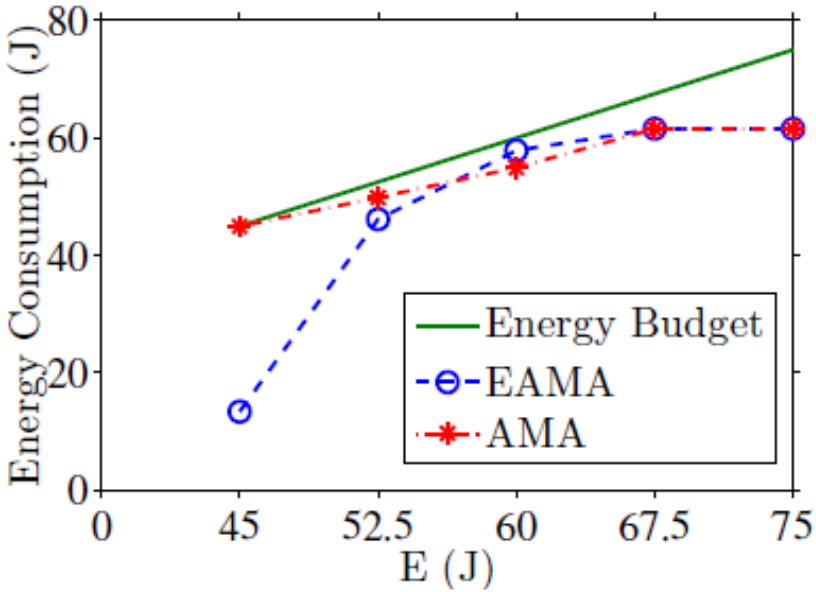
- The average energy consumption with 5 users in 21 days
  E= 52.5J



* EAMA is always within the energy budget

Variant Energy Budget in AM

- The average energy consumption with 5 users in 21 days
  E= 52.5J



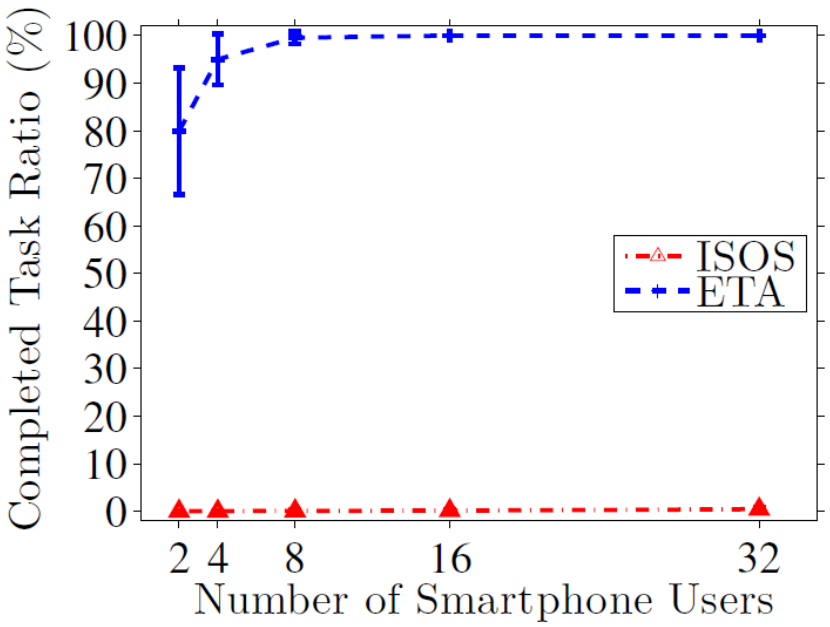
* EAMA saves more energy when E decreases

Data Collection

1. Collect posts from BBS as our queries
    1. The time of post => the time of query
    2. The IP address of post => the location of query
2. Convert the IP address into location using IPInfoDB

Improvement Completed Task Ratio by ETA

- Comparison between ETA and ISOS



* ETA always completes more tasks than ISOS