Academia Sinica, Taipei, Taiwan

SNHCC: Mobile Social Networks

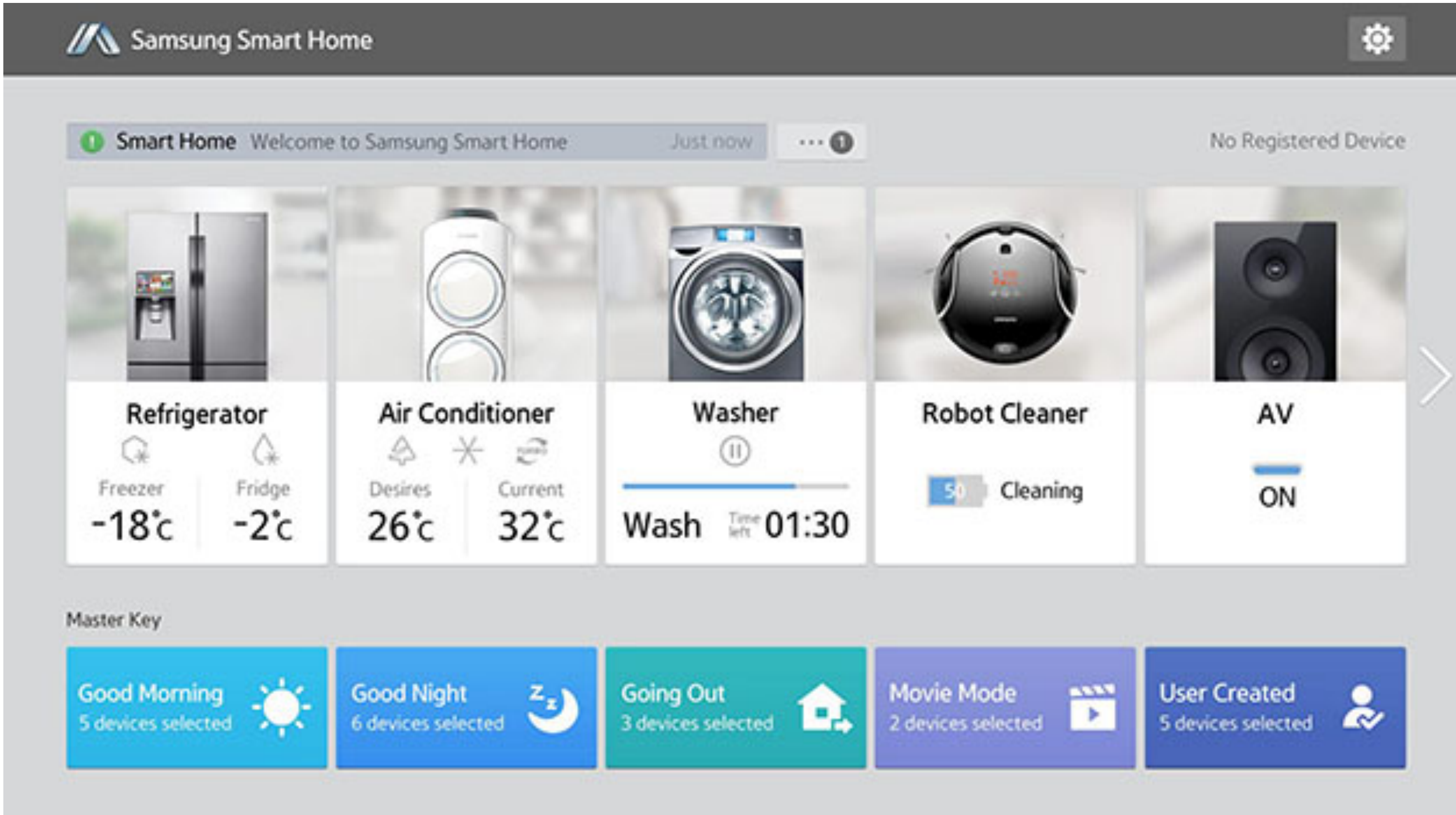
Socially Aware Computing: Concepts, Technologies, and Practices

Instructor:
Cheng-Hsin Hsu (NTHU)

Ubiquitous Computing

- Research topic for more than 2 decades
  - Instead of desktop/laptop computing, ubiquitous computing is made on any device, at any location, and in any form
- A.k.a. pervasive computing
- Integrated (distributed) systems with capabilities of
  - Sensing
  - Computations
  - Communications
- Affecting our daily life!
  - Targeted advertisements

Example: Smart Home

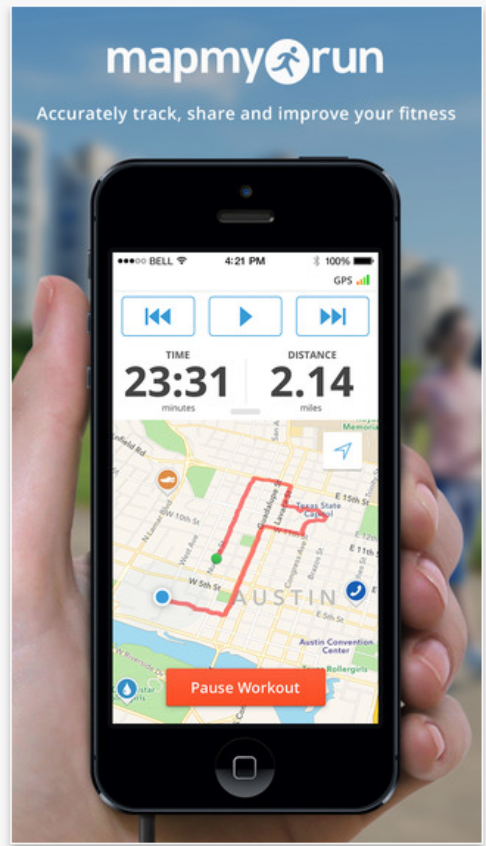


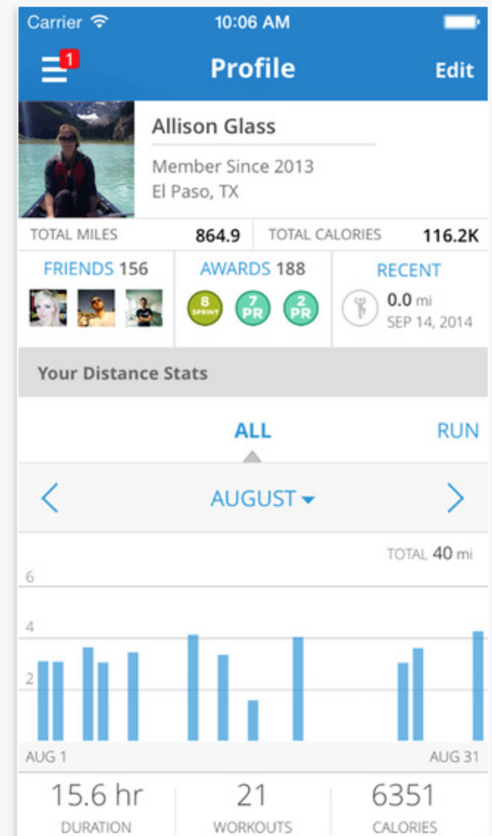
Socially Aware Computing

- To capture, quantify, and visualize social context to enhance human social interaction [Pentland 2005]
  - Tone
  - Gesture
  - posture
- Use massive data from the real world to understand [David Lazer et al. 2009]
  - Individuals
  - Organizations
  - Communities
  - Society

New Opportunities for Socially Aware Computing: Smartphones

- Smartphones can be used for socially aware computing because they are
  - quipped with many sensors
  - programmable
- Possible applications of smartphones:
  - Activity recognition: Calories calculator
  - Large-scale sensing: Noise level detection
  - Mobile social network: Figure out social relations using mobile phone data



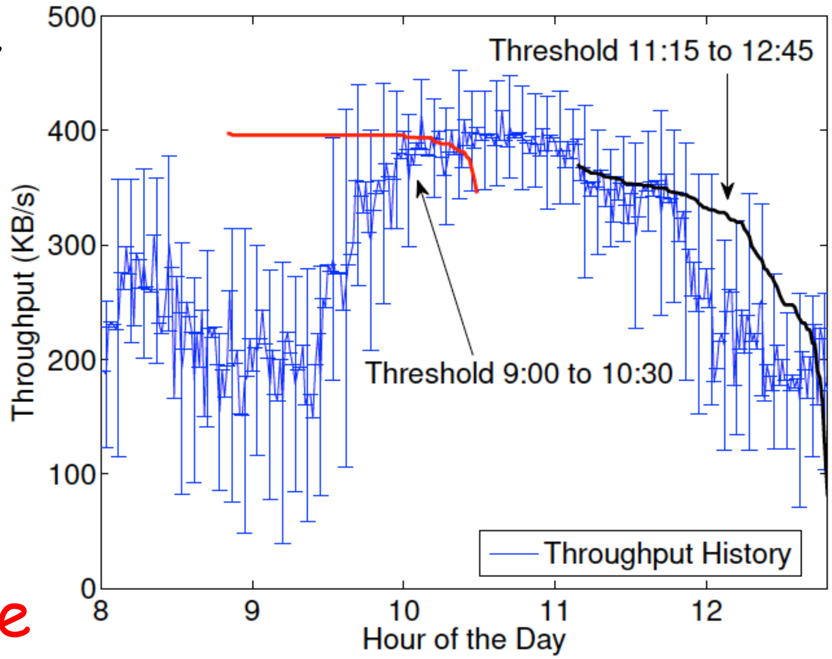


Problem with Traditional Way to Understand Humans

- Traditional way to under the behavior and interaction of humans (in the physical world) is via self-reported questionnaires, but
  - Subjective and could be biased
  - Static, cannot reflect dynamics
  - Small scale, due to high cost
- With the introduce of the Internet and WWW, computers can be used to analyze the human behavior and interaction in the virtual world (cyberspace)
- But the physical world analyses are still done using (online) questionnaires

Solution: Using Smartphones

- Use the smartphone sensors to sense the real world environment for sociological studies
  - No self-reported questionnaire
- Digital footprints of users
- This solves all 3 limitations
  - Objective outcomes
  - Spans over long time periods
  - Logs from many mobile users
- Smartphones enable socially aware computing, which is the combination of ubiquitous computing and social networks



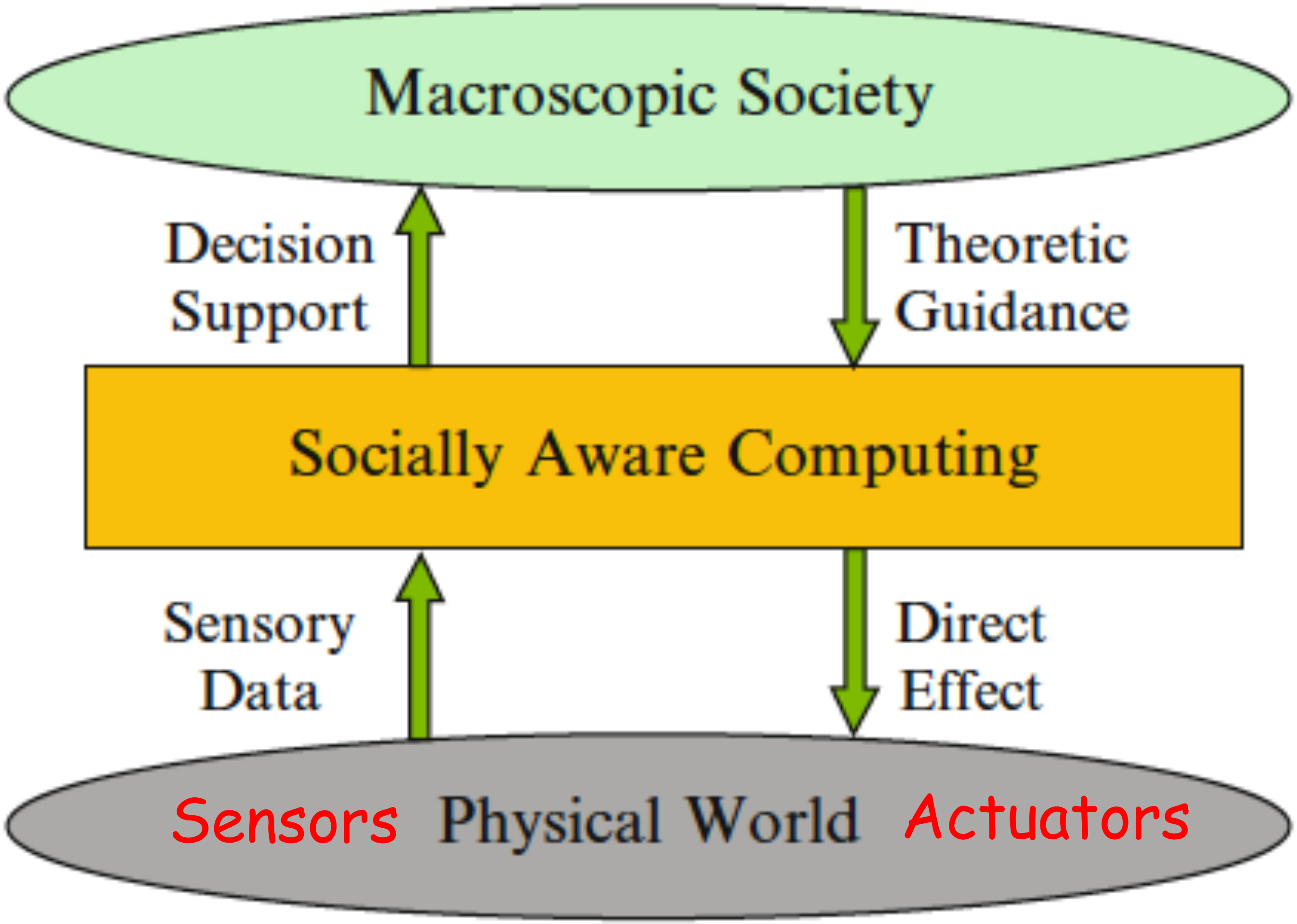
Socially Aware Computing

- Definition: Leverages large-scale, dynamic, continuous, and real-time sensory data to recognize individual behaviors, discover group interaction patterns, and support human communications and collaborations

- Sensors include
  - Ubiquitous (in-situ, infrastructure) sensors
  - Smartphone sensors
  - Internet data (emails, Web, call logs)

- Directly affect the physical world

High-Level Picture



Research Issues in Socially Aware Computing

This chapter covers the following 5 sample issues

1. Large-scale pervasive sensing
2. Activity and interaction inference
3. Social interaction support
4. Software framework and methodology
5. Applications

....Many more

Issue 1: Large-scale Pervasive Sensing

- Three sources of sensory data: mobile sensors, social Web, and infrastructure sensors
- Mobile sensors are attached to moving objects
  - Cars (GPS loggers and in-vehicle cameras) and human (smartphones, smart watches, and smart rings)
- Social Webs are online social networks
  - Twitter, Facebook, and others ← Web 2.0 users are consider as citizen sensors [Sheth 2009]
- Infrastructure sensors are fixed sensors
  - Surveillance cameras, environmental sensors, and positioning sensors

Issue 1: Large-scale Pervasive Sensing (cont.)

Challenge #1: Multimodal Data Processing

- Sensory data can be video, image, audio, text, and other continuous or discrete values
  - Continuous (analog) or discrete (digital) in various domains
- Different sensors achieve different accuracy levels and consume different energy amount
- How to extract the main features from the sensory data from each sensor?
- There may be some correlation among contexts
  - I am home => I'm not driving

Issue 1: Large-scale Pervasive Sensing (cont.)

Challenge #2: Semantic Representation

- The features (and maybe raw data) need to be represented in a unified way

Challenge #3: Large-Scale Sensing Data Fusion

- To infer different contexts, different sets of sensors can be used ← but for different accuracy/energy tradeoff

Challenge #4: Large-Scale Sensing Data Storage

- How to store, add, lookup multimodal data

Issue 2: Activity and Interaction Analysis

- Using sensory data, it is possible to recognize
  - Individual activity
  - Group interaction
- To infer individual activity, there are two approaches
  - Monitoring the human body using the sensors on the body ← such as walking, running, and exercising
  - Monitor how a human interacts with objects (sensors or tags are on objects) ← such as phoning, cooking, and washing hands

Issue 2: Activity and Interaction Analysis (cont.)

- The performance of individual activity depends on the learning model
  - Supervised learning ← classification problems ← static or temporal classifiers
  - Unsupervised learning ← clustering problems

- Analyzing individual activities is still hard
  - Multi-goal activities? ← I'm reading and eating at the same time
  - Multiple user perform a cooperative activity
  - Detecting abnormal activities ← unbalanced data problem ← we have little historical data on abnormal activities

Issue 2: Activity and Interaction Analysis (cont.)

- Various types of group interaction analysis
  - Group relationship reasoning ← e.g., friendship based on proximity (Bluetooth scans)
  - Interaction pattern discovery ← e.g., human interaction patterns on head gestures, attention, speech tone, and speak time
  - Community structure detection ← e.g., analyzing call records to implicitly build social networks
  - Evolution analysis ← e.g., dynamics of co-authorship

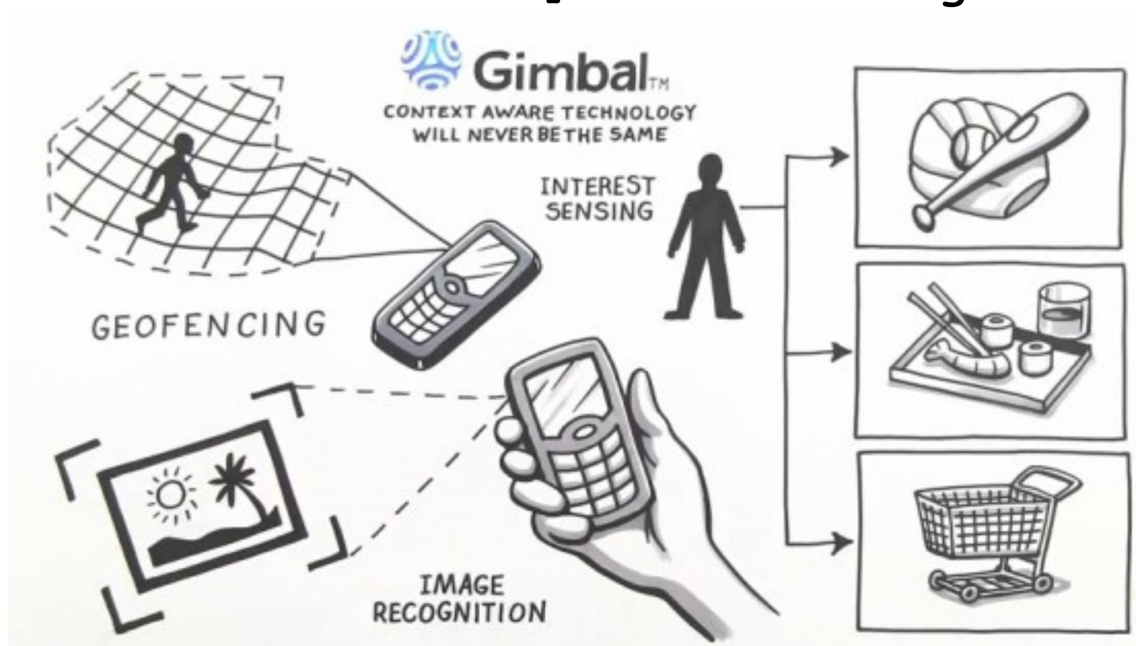
Many other possibilities given the tremendous amount of data from sensors!

Issue 3: Social Interaction Support

- How analysis results affect how human beings interact with system?
  - Personalized recommendation ← e.g., sync up two users' smartphone contacts if they share at least one friend
  - Social status visualization ← e.g., help the meeting organizer to under whether all participants reaches a consensus
  - Group collaboration ← e.g., show the co-authorship on a big display at a conference to introduce participants to each other
  - Smart decision-making ← e.g., targeted advertisements, the Macdonald example

Issue 4: Software Framework and Methodology

- Several projects aim to develop a framework for socially aware computing
  - WearCom is a framework proposed for wearable devices [Kortuem and Segall 2003] ← much earlier than wearable devices become hot recently
  - Sharing real-time context information among users in the same group [Raento and Oulasvirta 2008] ← several design decisions are presented
- There is a huge room for framework development

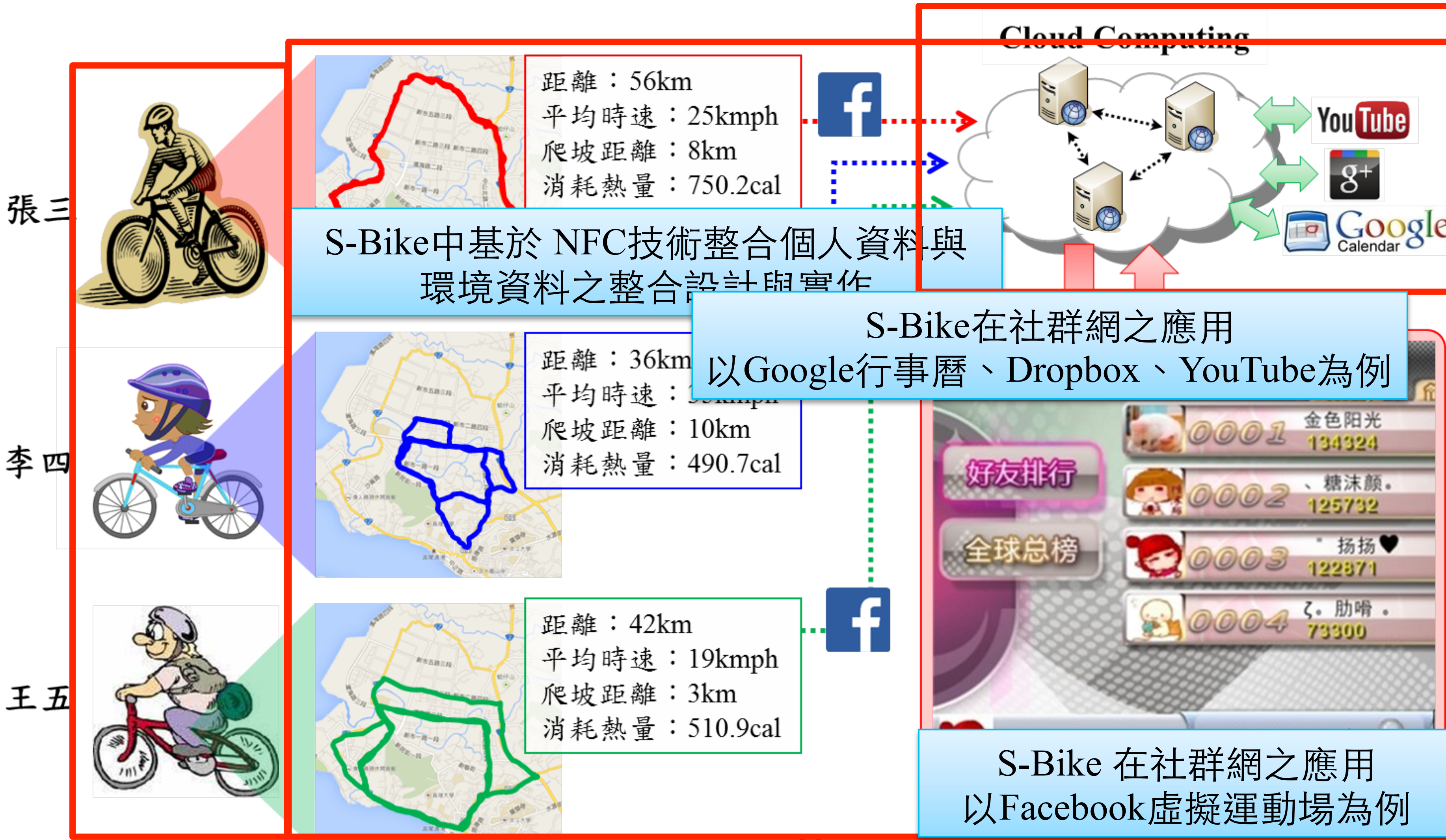


Issue 5: Applications

- Different socially aware computing applications have been proposed
  - Public health ← based on friendship relation, give people shots to control the outbreak
  - Public safety ← detecting the disasters, say warning system for earthquakes
  - Urban planning (computing) ← put sensors on public bikes, so that bikers can use their smartphone to find the shortest routes based on the road conditions and congestion levels

Issue 5: Applications (cont.)

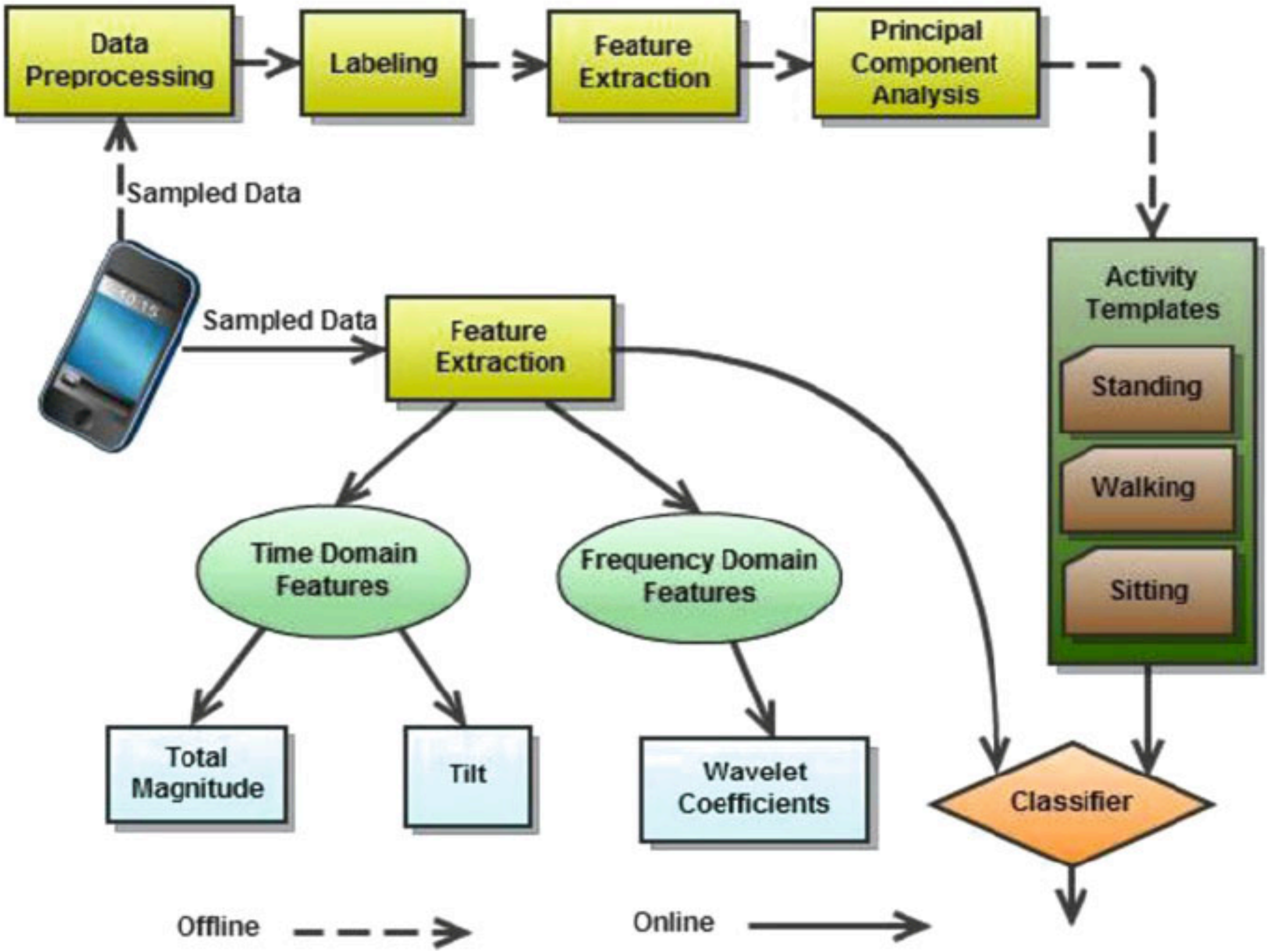
Virtual biking tournaments: An ongoing project at NTHU



Case Study #1: Activity Recognition

- Detecting activities using 3-axis accelerometers on smartphones
- Two types of activities
  - Static activities: such as standing, sitting, and driving
  - Repetitive activities: such as walking, running, cycling, and jumping
- Divided into two steps
  - Offline data training → activity templates
  - Online classification → feature extraction and classifier

Case Study #1: Activity Recognition (cont.)



Case Study #1: Activity Recognition (cont.)

- Two takeaways
  - Good overall recognition rate
  - Static activities only need time-domain feature ← lightweight

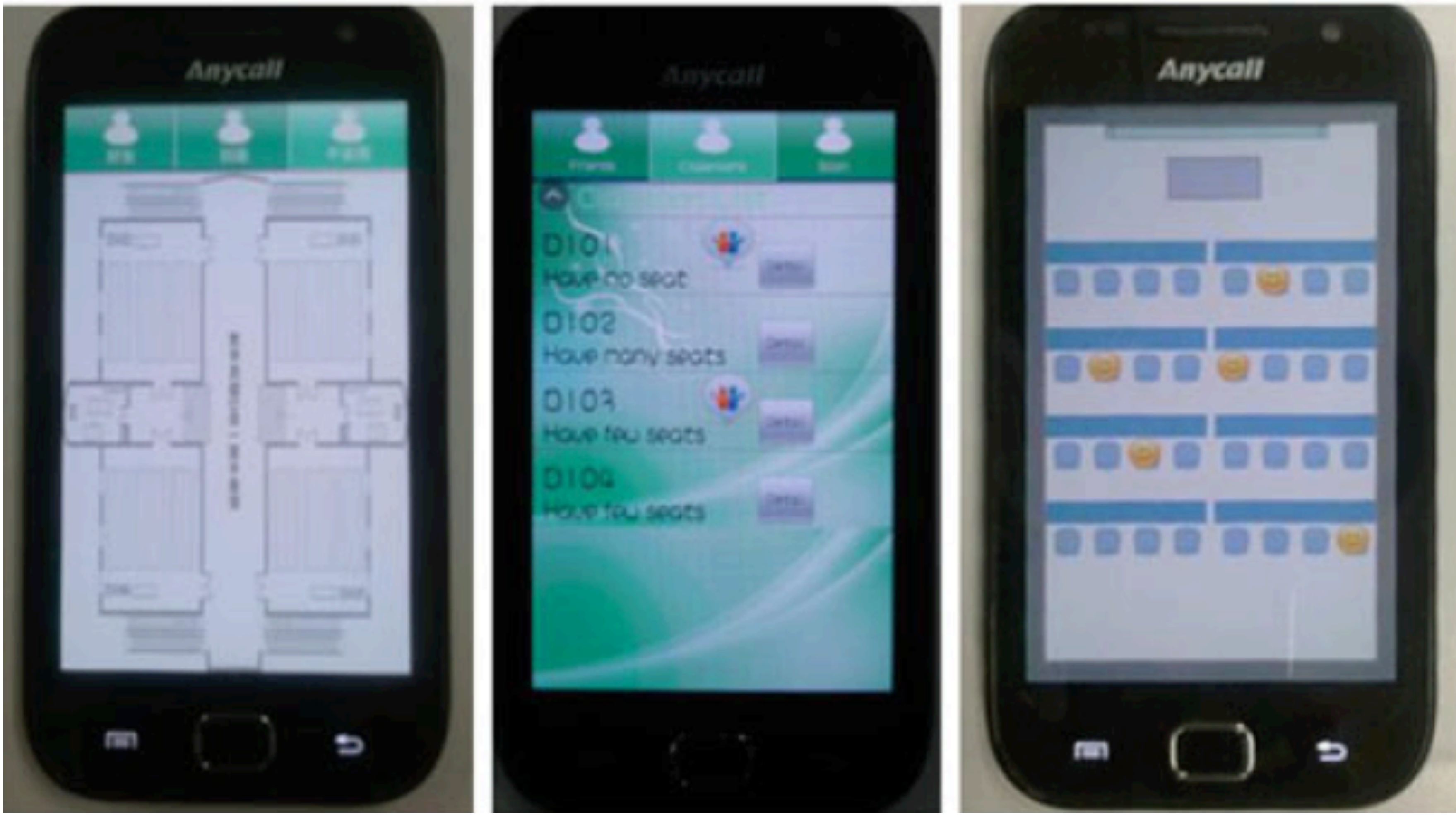
Activity	Percentage of records correctly recognized: Time-domain features (%)	Frequency-domain features (%)	Recognition rate (%)
Standing	98.98	1.02	98
Sitting	100	0	100
Lying (prone)	100	0	100
Lying (supine)	99.28	0.72	100
Driving	37.69	62.31	80
Walking	0	100	80
Running	56.76	43.24	86
Ascending	0	100	88
Descending	0	100	82
Cycling	97.50	2.50	84
Jumping	0	100	82
Average	53.66	46.34	89.1

Case Study #2: Enhancing Social Interaction

- A complete platform to enhance campus life, answering questions like:
  - Is the study lounge available?
  - What classrooms are my friends in?
  - Is the tennis court crowded?
- Three apps:
  - Where2study ← find the room and location to study
  - I-sensing ← campus information-sharing system, asking others to sense for us ← we have a similar project called SAIS ← Will present later
  - BlueShare ← share multimedia content (videos) over short range Bluetooth ← we have a similar project called CCDN ← will present later

Case Study #2: Enhancing Social Interaction (cont.)

- WheretoStudy ← how to get the vacant seat???



Case Study #3: Mining Mobile Phone Data

- Dataset: from MIT Reality Mining project [Eagle et al. 2009]
- Features: proximity, calls, and messages
- First, use SVM to predict friendship
- Second, analyze the social relation evolution

- Not too much details, there might be some more recent details ← may present them in the class

Conclusion

- We define the present socially aware computing
- The research are sensing-based, data-driven, and field-study-based

- Many potential (cool) problems to solve
- Many open questions will be answered throughout this semester

- I will present two projects done at NTHU if time permits

Questions?

Contact me at chsu@cs.nthu.edu.tw anytime

SAIS: Smartphone Augmented Infrastructure Sensing for Public Safety and Sustainability [EMASC'14]

Chen-Chih Liao[1], Ting-Fang Hou[1], Ting-Yi Lin[1], Yi-Jun Cheng[1], Aiman Erbad[2], Cheng-Hsin Hsu[1], and Nalini Venkatasubramanian[3]

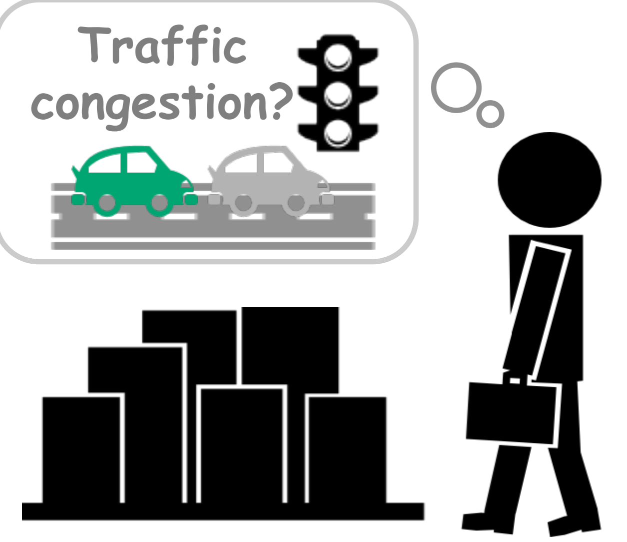
[1]Department of Computer Science, National Tsing Hua University, Taiwan
[2]Department of Computer Science and Engineering, Qatar University, Qatar
[3]Department of Computer Science, University of California Irvine, USA

Outline

- Overview
- Motivations
- Crowdsourcing System
- Task Assignment Problem
- Optimization Algorithm
- Efficient Task Assignment Algorithm
- Simulation Results
- Conclusion
- Future Work

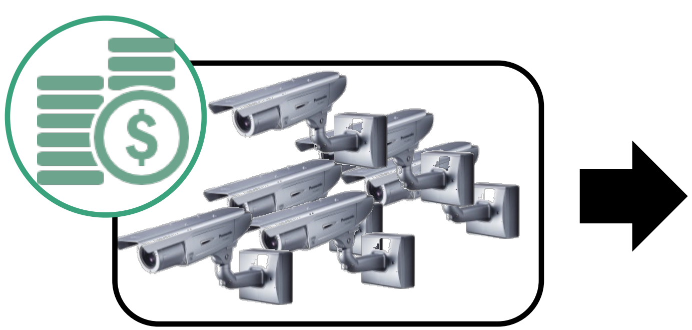
Sensing in Smart City

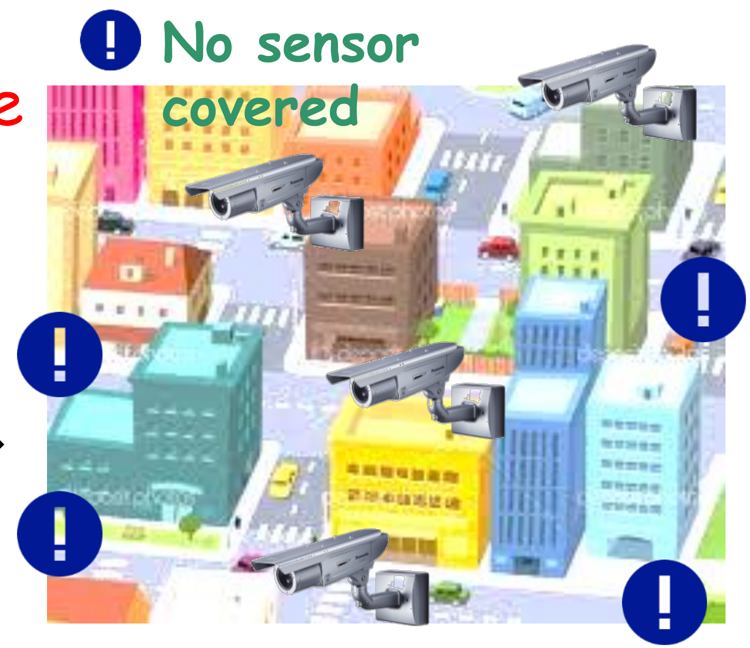
- Sensing platforms improve situation awareness by allowing individuals to query the environment for interested events
- For instance, we may want to know whether traffic jam occurs on the way we are going to take when we are in a hurry or whether shops are crowded



Infrastructure Sensing

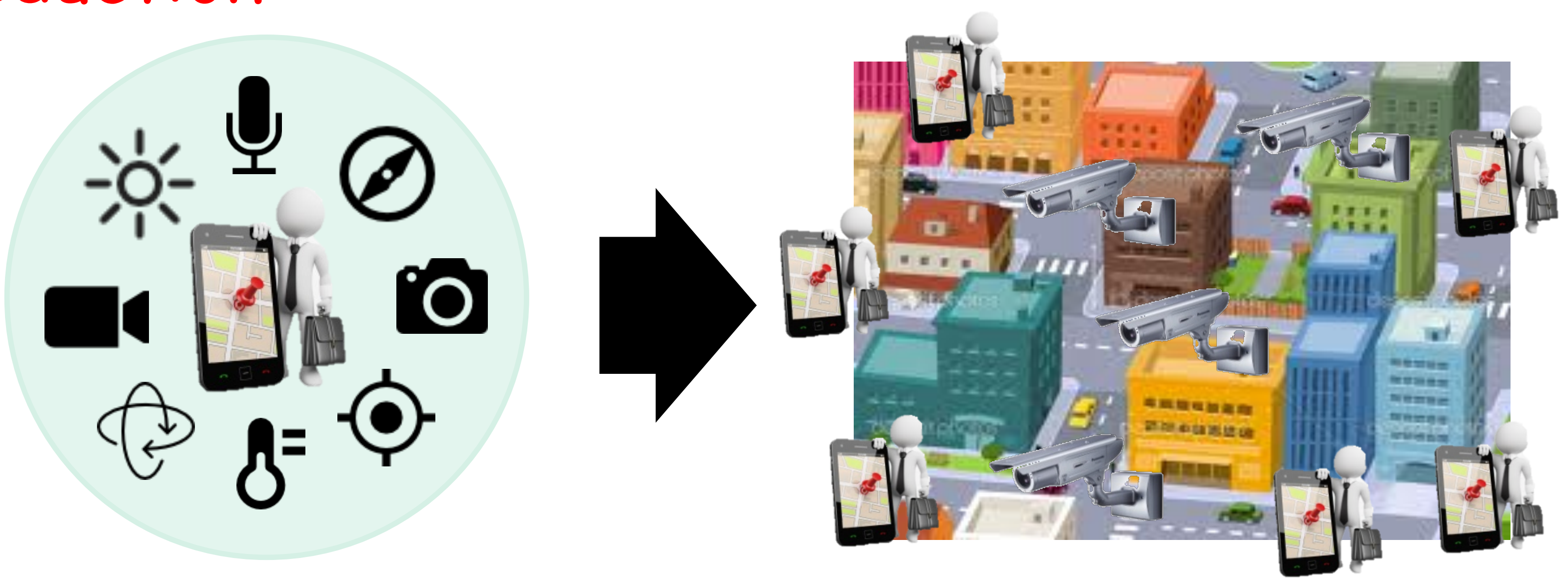
- Infrastructure sensing helps for providing city information
- But, deploying, managing, and maintaining in-situ sensors is actually an expensive, labor intensive, and error-prone process
- We may suffer from incomplete or inaccurate sensory data due to limited resources
- Our solution is to leverage the power of crowds





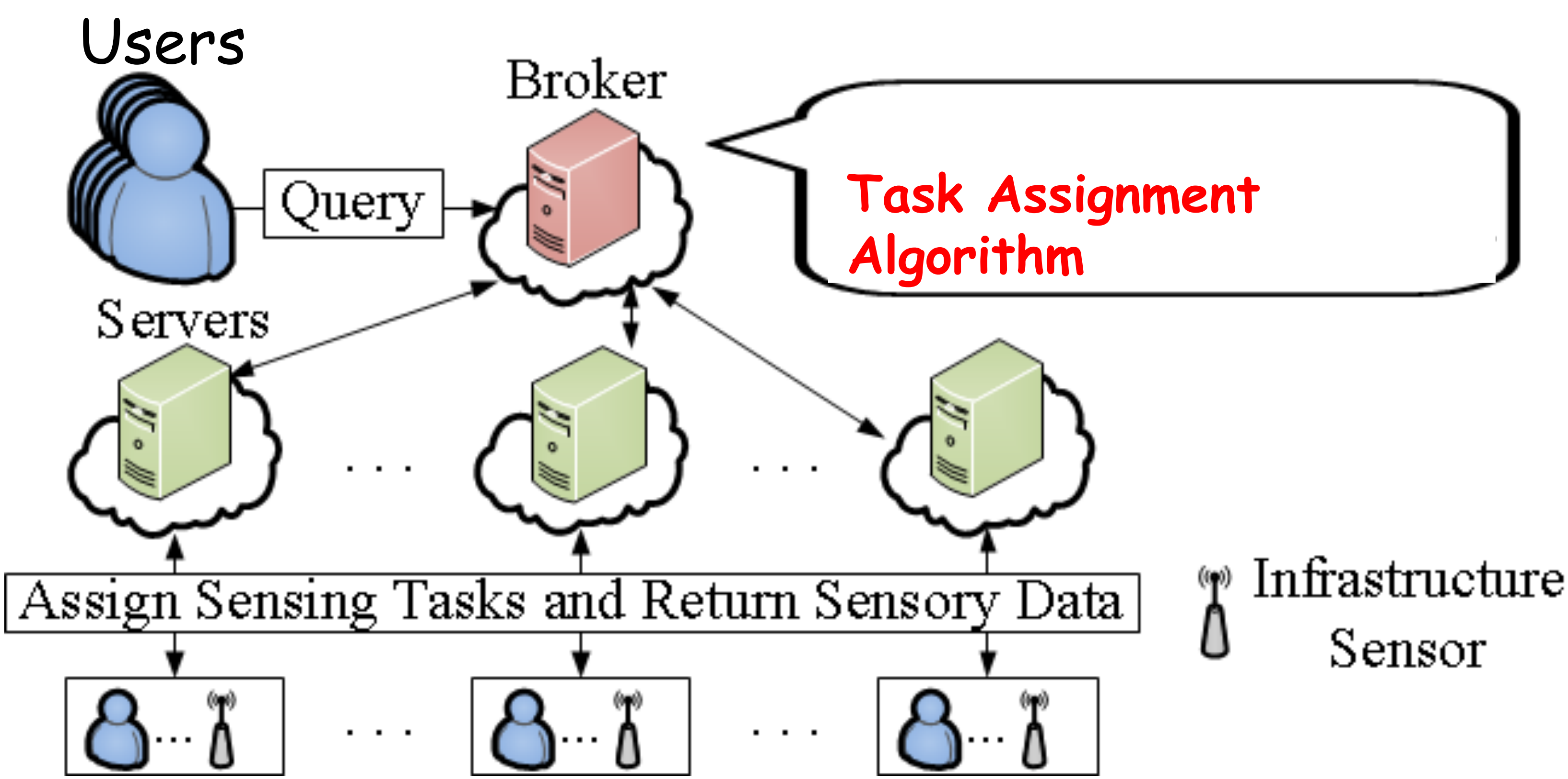
Smartphone-Augmented Sensing

- Smartphones are equipped with various kinds of sensors, such GPS, gyroscopes, microphones, cameras, and etc.
- Using smartphones to augment infrastructure sensing by incorporating crowdsensing for cost reduction



Our Proposed SAIS Platform

- We develop a platform called Smartphone Augmented Infrastructure Sensing (SAIS)
- SAIS consists of *brokers*, *servers*, and *users*



How SAIS Works?

- ***Users*** submit queries to the ***broker*** for interested events, such as degree of crowdedness at a location
- Both smartphone users and in-situ sensors are workers that perform sensing tasks for each query
- The **broker** runs a task assignment algorithm that jointly determines: (i) the assignment of sensing tasks and (ii) the dispatch of smartphone users
- ***Severs*** collect sensory data, derive (compute) answers and return the answers to **users**

Prototype System Implementation

- We have implemented the proposed SAIS system on Linux (server) and Android (client)
- GUIs for: (a) submitting queries, (b) choosing a sensing task , and (c)performing the sensing task.

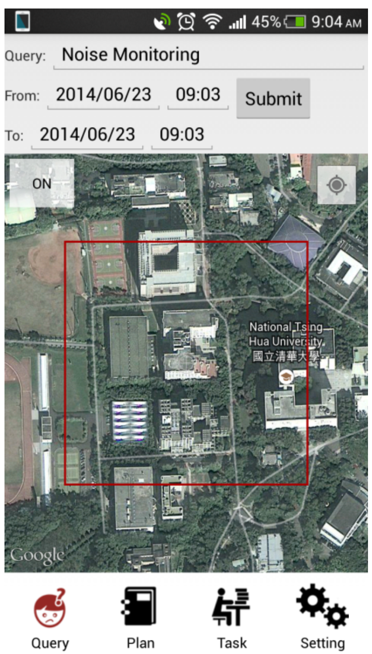


Fig (a)

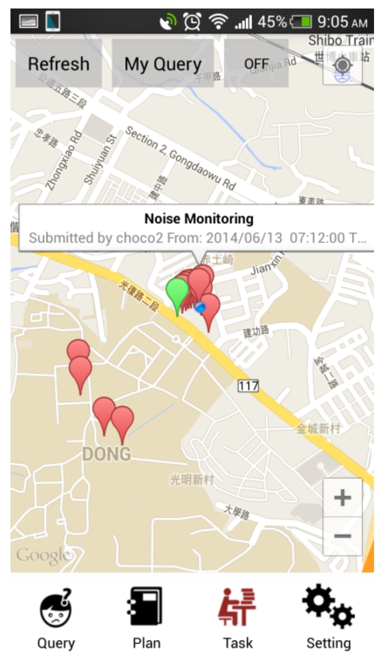


Fig (b)

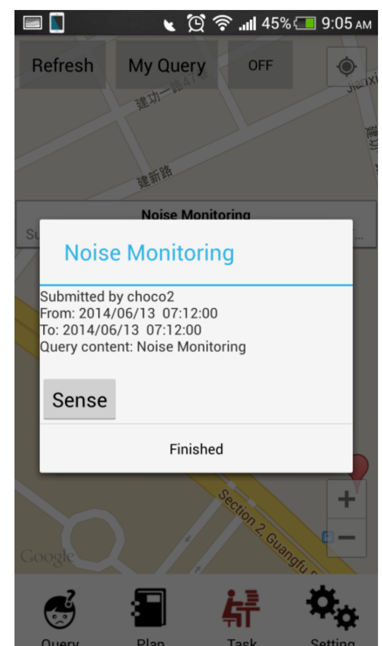
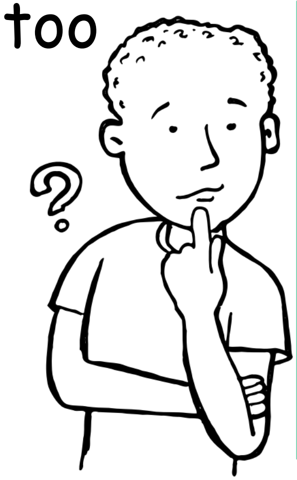
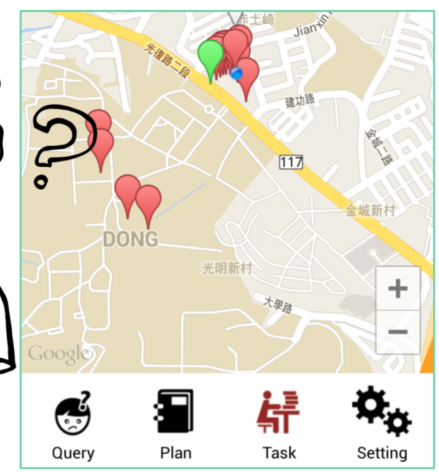


Fig (c)

User Study

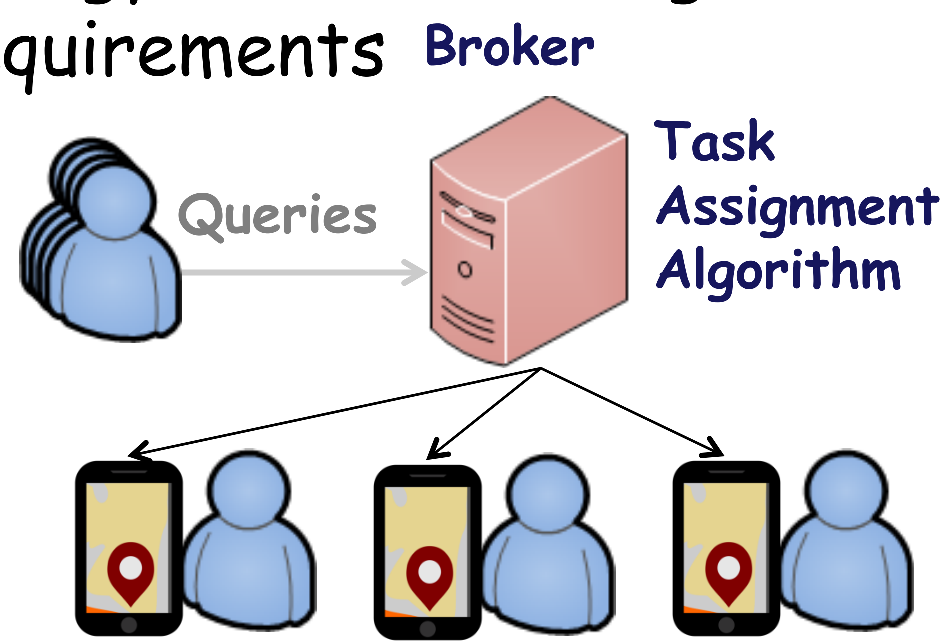
- 7 students for a 7-day experiment
- Most users thought the system helps their daily lives
- However, there are many sensing tasks submitted by users and all shown on the map
  - Users feel confused on which sensing tasks to choose and just randomly pick any of them
  - Tasks may be either performed too many times or left unanswered
- Inefficient use of resources of crowds





Task Assignment Problem

- An efficient task assignment algorithm is needed for our system that takes responsibility for the best leverage of resources from smartphone users
- The goal is to achieve sustainability of the system through energy-efficient design while satisfying requirements of each query



Problem Formulation

- We formulate the problem to minimize the overall energy consumption including both sensing and worker movement energy cost
- Our objective function

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

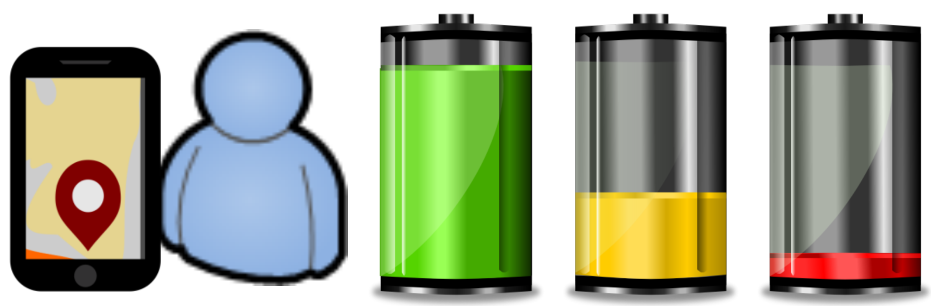
- Decision variables are defined to determine the assignment of sensing tasks to workers and the traveling routes of workers

Constraints

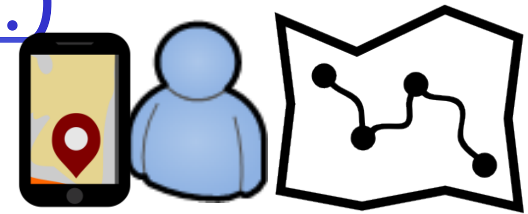
- Eq. (2): Query required accuracy levels are satisfied
- Eq. (3): Energy budget of each worker is not violated

$$Q_{r,s}(W'_{s,r_p}) \leq f_s, \forall r \in \mathbf{R}, \forall s \in \mathbf{r_c}; \quad (2)$$

$$F_w - \sum_{l \in \mathbf{L}} \sum_{s \in \mathbf{S}} e_{w,s} x_{w,l,s} \geq \theta_w, \forall w \in \mathbf{W}^*; \quad (3)$$



Constraints (cont.)



- Computing the paths for workers to travel along the query locations

$$\sum_{j \in \mathbf{L}} E^{w}_{j,A_w} = 0, \forall w \in \mathbf{W}^*; \quad (4)$$

$$\sum_{j \in \mathbf{L}} E^{w}_{A_w,j} = \left\lceil \frac{\sum_{j \in \mathbf{L}-A_w} \sum_{s \in \mathbf{S}} x_{w,j,s}}{|\mathbf{L}||\mathbf{S}|+1} \right\rceil, \forall w \in \mathbf{W}^*; \quad (5)$$

$$\sum_{j \in \mathbf{L}, j \neq i} E^{w}_{j,i} = \left\lceil \frac{\sum_{s \in \mathbf{S}} x_{w,i,s}}{|\mathbf{S}|+1} \right\rceil, \forall i \in \mathbf{L} - A_w, \forall w \in \mathbf{W}^*; \quad (6)$$

$$\sum_{j \in \mathbf{L}, j \neq i} E^{w}_{i,j} \leq \left\lceil \frac{\sum_{s \in \mathbf{S}} x_{w,i,s}}{|\mathbf{S}|+1} \right\rceil, \forall i \in \mathbf{L} - A_w, \forall w \in \mathbf{W}^*. \quad (7)$$

Optimization Algorithm

- The problem can be solved by generic optimization solvers, such as CPLEX and GLPK
    - We develop an optimal algorithm using CPLEX
- However, it leads to long running time even for small scale problems
- It may be less practical to employ the optimal algorithm in real deployment

Efficient Task Assignment Algorithm (ETA)

- We develop a greedy algorithm
- $T_{w,l}$ denotes the number of sensing tasks can be performed by worker *w* at location *l*
- $Y_{w,l}$ denotes the corresponding energy consumption
- The ratio between them is the utility!

$$T_{w,l} = \sum_{r \in \mathbf{R}} \sum_{s \in \mathbf{r_c}} z_{w,l,s} G^{w}_{s,l,r_p} w_{r,s}, \forall w \in \mathbf{W}, l \in \mathbf{L}. \qquad (8)$$

$$Y_{w,l} = \alpha \sum_{s \in \mathbf{S}} e_{w,s} \left[\frac{\sum_{r \in \mathbf{R}} n_{r,s} z_{w,l,s} G^{w}_{s,l,r_p} w_{r,s}}{|\mathbf{R}|} \right] + \beta d_{A_w,l}, \forall w \in \mathbf{W}, l \in \mathbf{L}. \qquad (9)$$

Simulation Setup

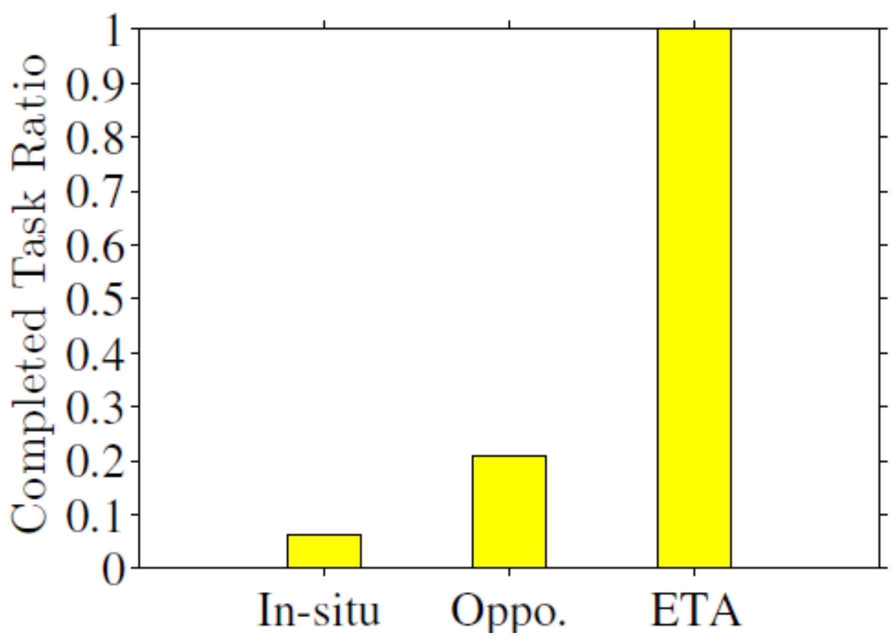
- We implement our and 3 baseline algorithms
  - Only *in-situ sensors* to provide sensory data
  - In-situ sensors with opportunistic sensing, where smartphone movement follows the random way point model
  - Optimal algorithm
- We use trace data collected from a popular Bulletin Board System (BBS) in Taiwan, PTT, to derive:
  - Poster locations (IPs)
  - Asked queries (keywords)
- In particular, 5700 posts in 10 days are collected and used to drive our simulator

Performance Metrics

- Several simulations are conducted to study the following performances
  - Completed task ratio
  - Energy consumption
  - Responding time, referred as the time difference between the broker receives the query and the query is answered
- Each simulation is repeated 5 times with mean, minimum, maximum results reported

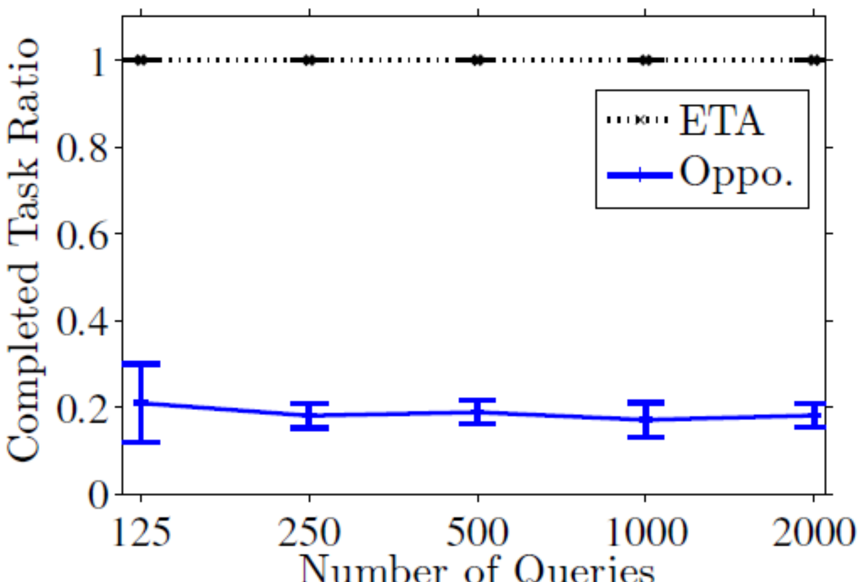
Overall Completed Task Ratio

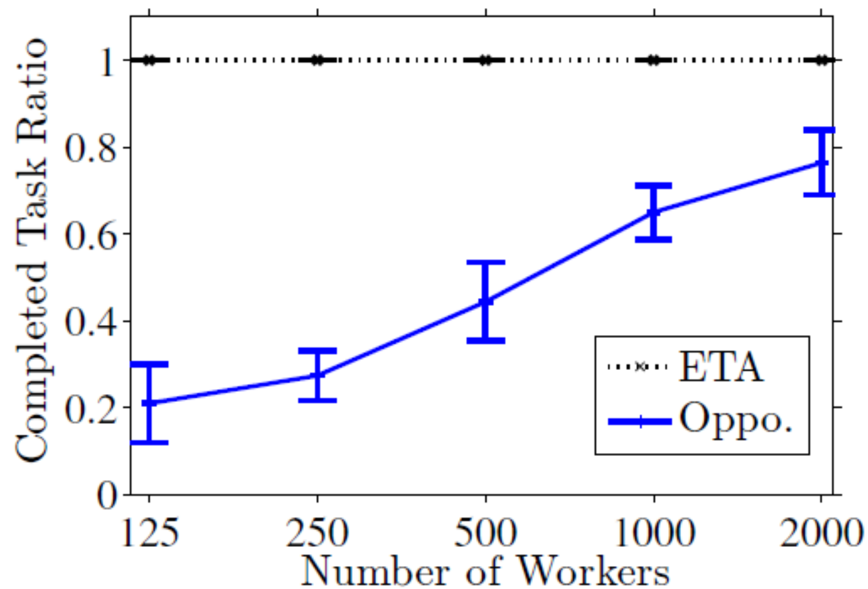
- Lower ratios significantly degrade the user experience
- 90+% improvement compared to in-situ (only) sensing
- 80% improvement compared to (in-situ sensing with) opportunistic sensing



Impact of Number of Workers and Queries

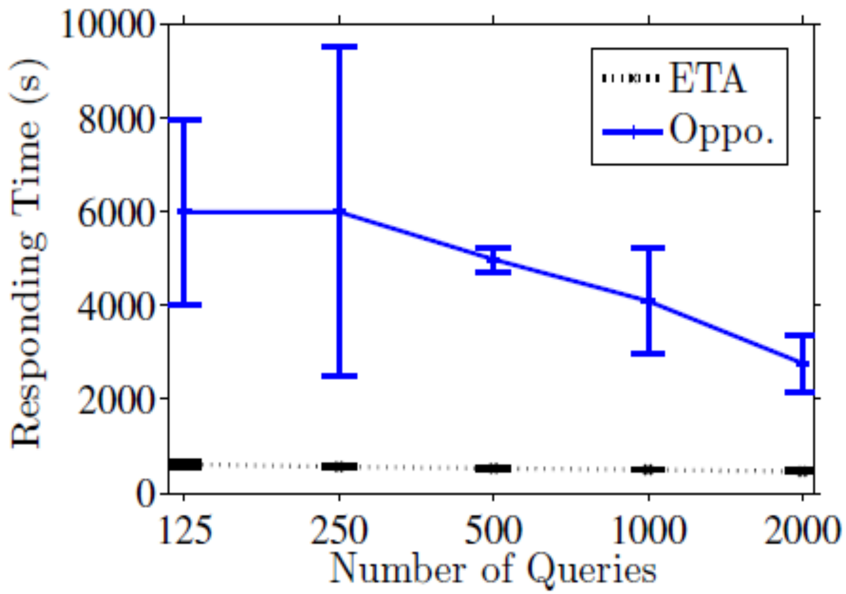
- ETA achieves high completed ratio even with many queries ← Left figure
- In-situ sensors with opportunistic sensing requires large number (two thousands, 16 times higher than ETA) of workers to achieve high completed ratio ← Right figure

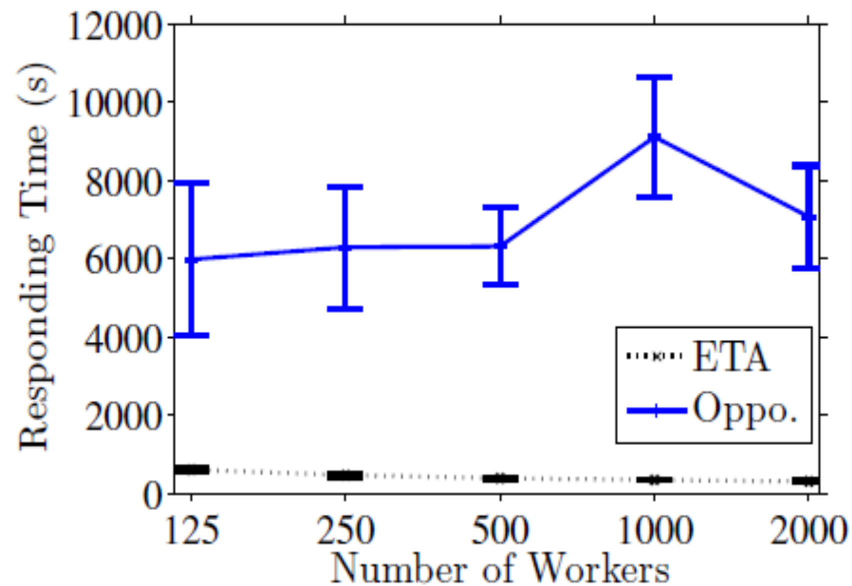




Impact of Number of Workers and Queries (cont.)

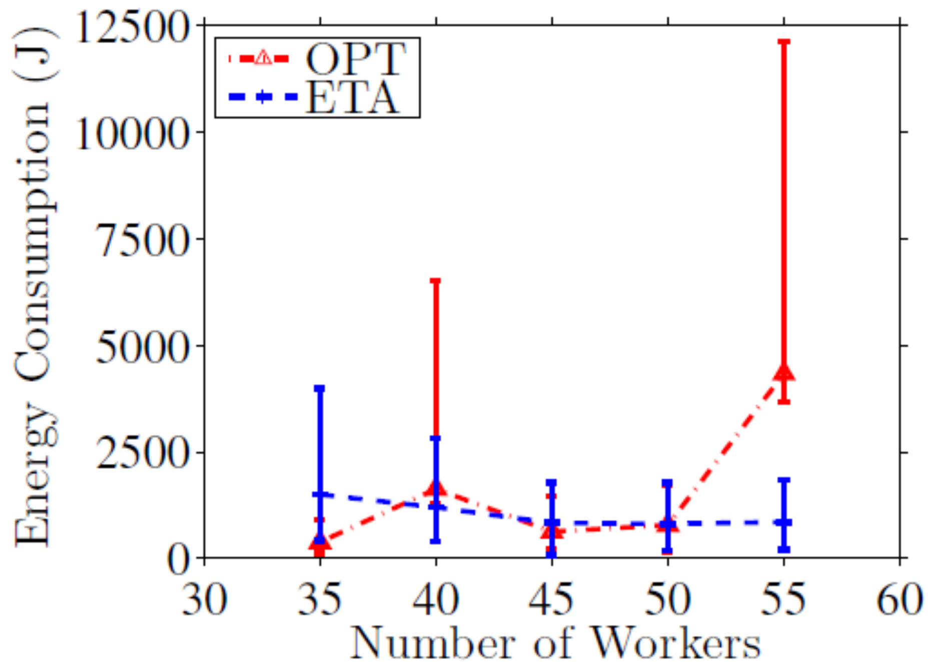
- The responding time of ETA is always less than 1000 seconds because ETA instructs workers to the required locations
- Opportunistic sensing cannot guide workers and thus suffer from up to 6 times higher responding time





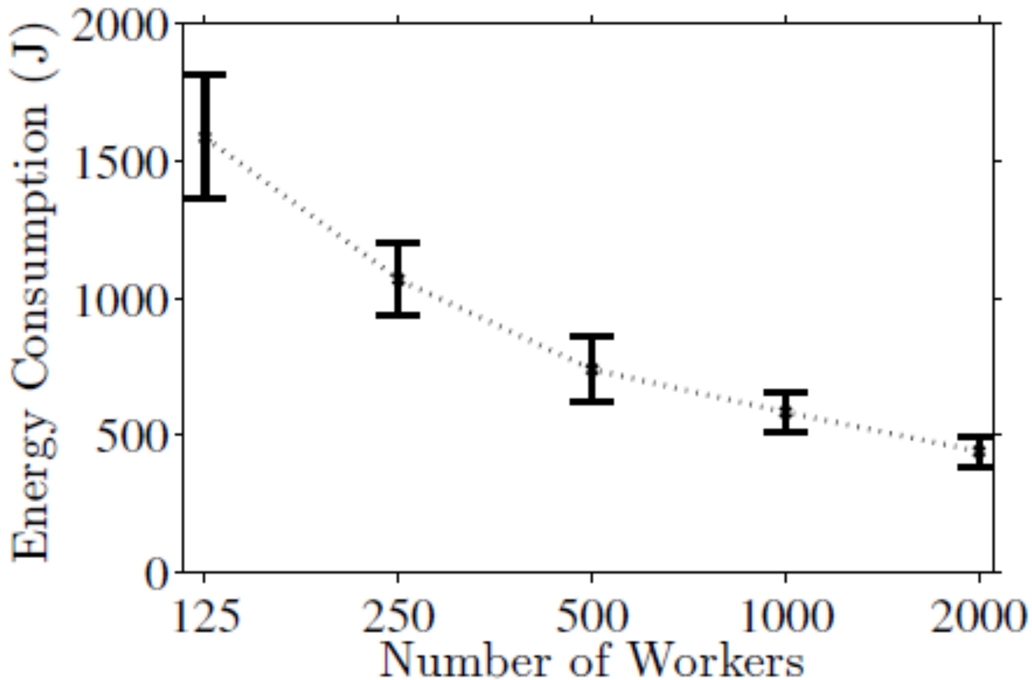
Optimality of ETA in Practical Settings

- The optimal algorithm is implemented in CPLEX with 1-min running time limit (realistic settings)
- ETA outperforms OPT once the number of workers exceeds 50



Benefits of More Workers

- ETA leads to lower energy consumption with more workers participation
- Because more workers allow ETA to choose better workers

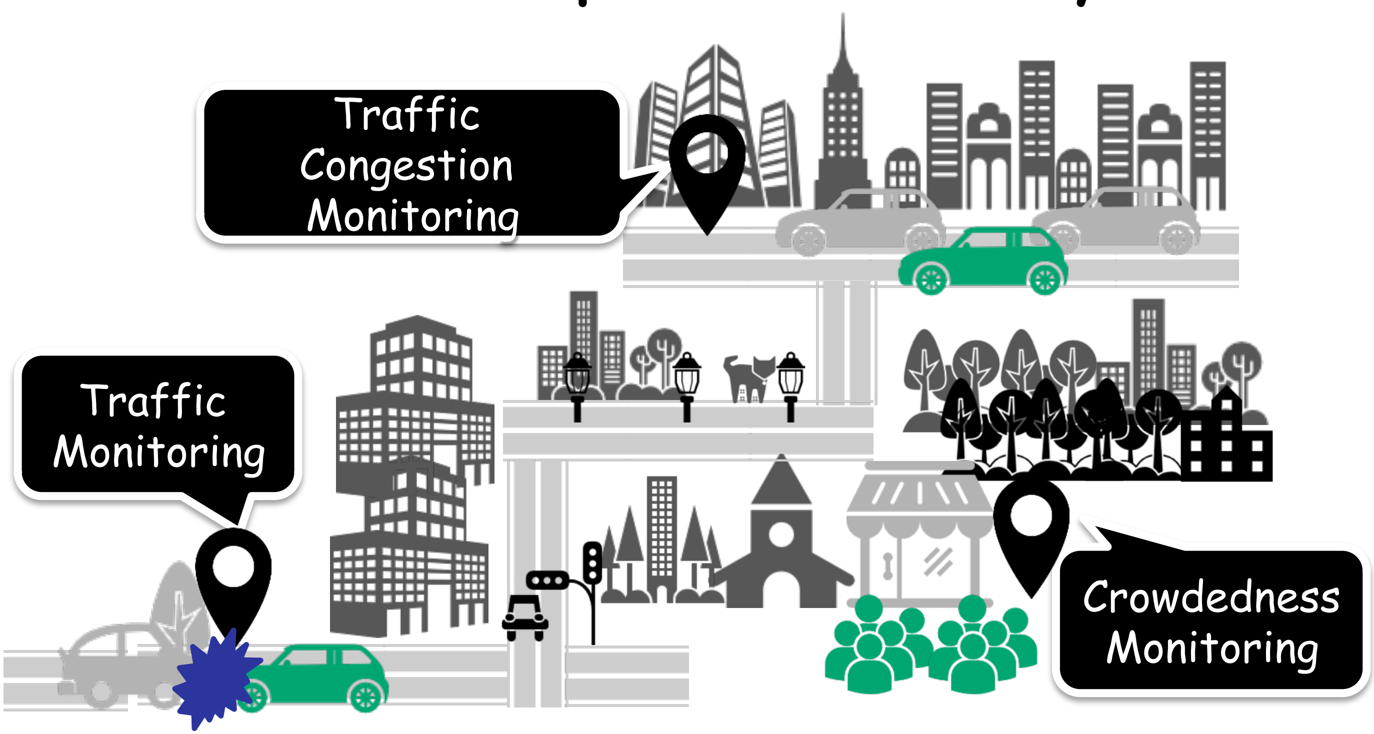


Conclusion and Future Work

- We present the SAIS platform of smartphone-augmented in-situ sensing
- We develop a task assignment algorithm (ETA) to leverage resources of smartphone users
- In our simulation results, we show
  - In-situ sensing achieves extremely low coverage
  - ETA completes 100% tasks with only 12.5% of workers compared to in-situ sensors with opportunistic sensing
  - ETA results in up to 6 times of responding time reduction
  - ETA performs better with more workers

Future Work

- A large-scale (Urban Computing) middleware design in order to ease the overhead of the deployment
- More research problems may surface



Questions?

Contact me at chsu@cs.nthu.edu.tw anytime

Distribution of Multimedia Content over Challenged Networks [Work-in-Progress]

Under weak network infrastructure, people cannot access any multimedia content.

In some popular places, they are still possible to have network access.

We propose a system called CCDN to distribute multimedia content

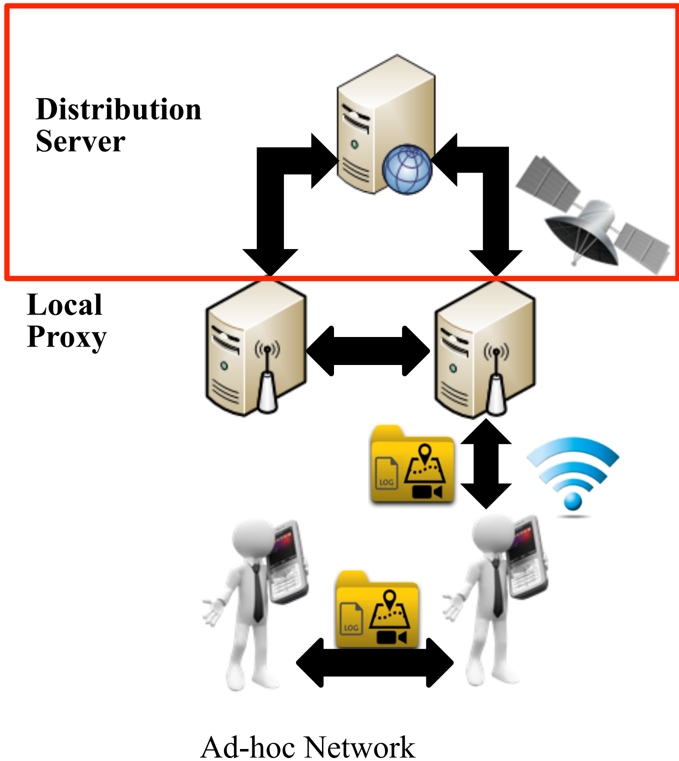
- ✓Deploy local proxies (LP) in the popular places to do distribution
- ✓Propose an efficient algorithm to decide that downloading which multimedia content from which LP (or user) can have highest user quality experience.

Usage Scenarios

- Local Proxies are located in some popular places:
  - Bus stations
  - City halls
  - Stores ← Starbucks, MacDonald's, and so on
- Mobile Users utilizes their WiFi to download multimedia content whenever they run into local proxies

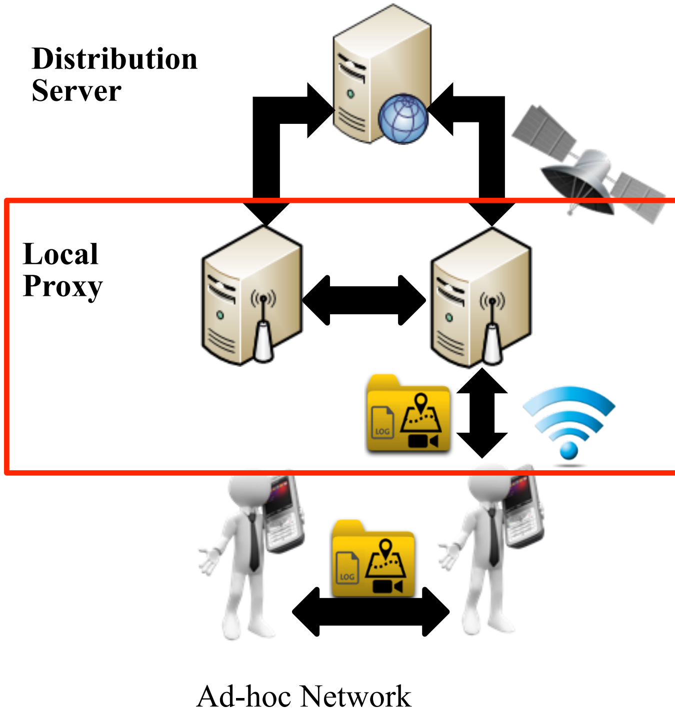
System Overview

- Distribution server:
  - Collect social profiles of users from local proxies ← What are social profiles?
  - Create distribution plan
  - Push multimedia content to local proxies



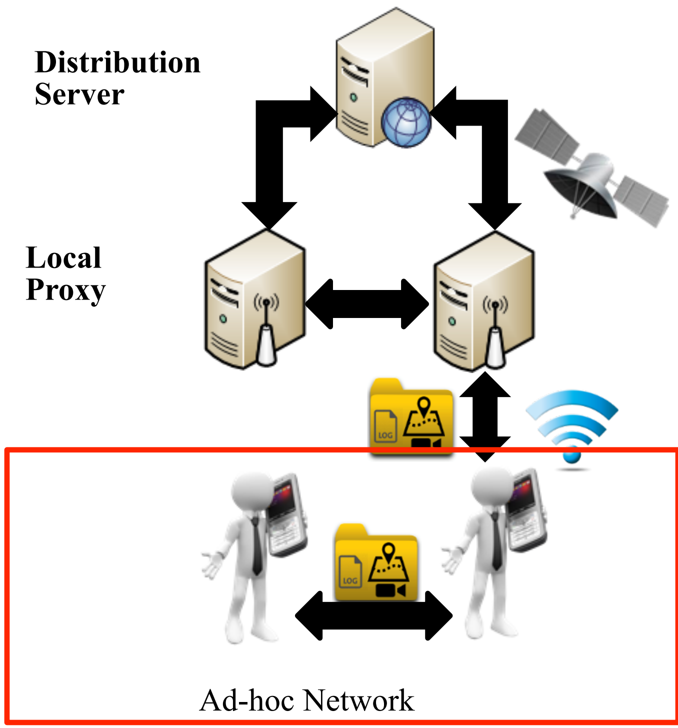
System Overview (cont.)

- Local Proxy:
  - Collect social profiles from users and push them to distribution server
  - Forward the distribution plan to mobile users whenever there is a contact
  - Deployed at popular places and cache the multimedia content for mobile users



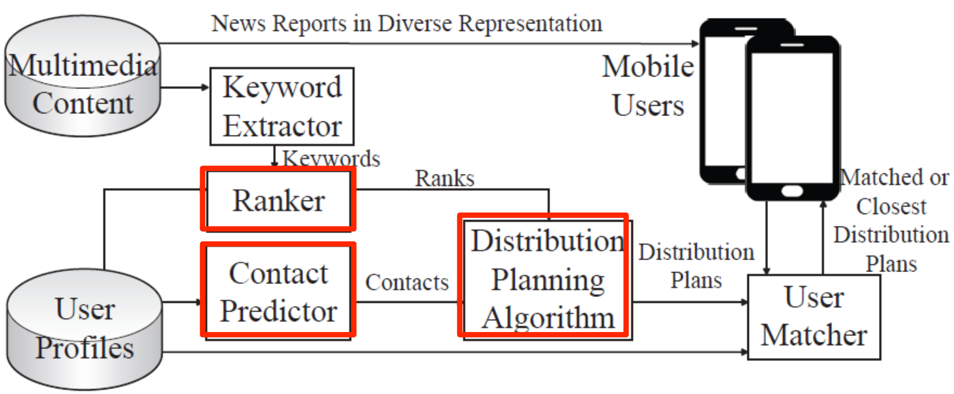
System Overview (cont.)

- Mobile Users:
  - Send their own social profiles, such as encounters, watched multimedia content, contact duration, throughput, etc to local proxies and then distribution server
  - Follow the plan to download corresponding multimedia content for later consumption



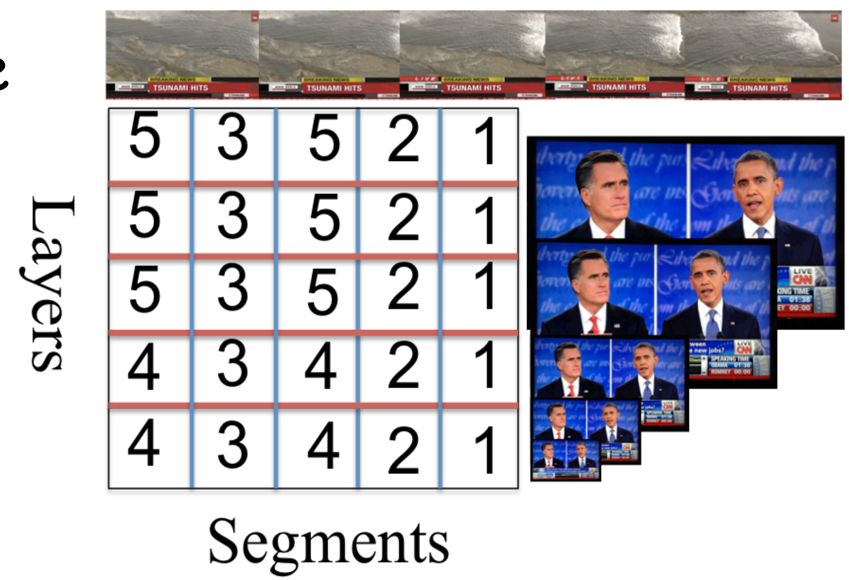
How to Create Distribution Plan?

- Ranker
  - Predict the viewing probability of each multimedia content based on historical social profiles.
- Contact Predictor
  - Predict the encounters of each user based on historical user trajectory
- Distribution Planning Algorithm
  - If we know the encounters, importance of multimedia content, and some information of multimedia content, we can design an algorithm to disseminate useful content to mobile users



Multi Layer Multimedia Content

- In our system, the contacts may too short to transfer the whole multimedia contents
- We can reduce the size of multimedia content
  - Temporal: segments
  - Special: transcoding
- We give different layer different user experience improvements



Problem Formulation - Objective

$$\max \sum_{u=1}^{U} \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{c=1}^{C} x_{u,n,l,c} \; \rho_{nL+l} \; \psi_{u,n} \tag{1a}$$

- The first step is to define the objective function
  - Maximize the total user experience improvement
- Decision variable
  - X is the 0-1 decision variable to decide which unit should be downloaded from which contact

Problem Formulation - Constraints

$$st: \psi_{p_{u',c',n'}} \geq \bar{\psi} x_{u',n',l',c'} \; \forall 1 \leq u' \leq U, 1 \leq n' \leq N, 1 \leq l' \leq L,$$
$$1 \leq c' \leq C \tag{1b}$$

Viewing Probability

$$\sum_{n=1}^{N} \sum_{l=1}^{L} b_{nL+l} x_{u',n,l,c'} \leq r_{u',c'} \kappa_{u',c'} \forall 1 \leq c' \leq C, 1 \leq u' \leq U \tag{1c}$$

Bandwidth Budget

$$\sum_{c=1}^{C} x_{u',n',l',c} \geq \sum_{c=1}^{C} x_{u',n',l'',c} \; \forall 1 \leq u' \leq U, 1 \leq n' \leq N,$$
$$1 \leq l' < l'' \leq L \tag{1d}$$

Layer Dependency

$$\sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{c=1}^{C} b_{nL+l} x_{u',n,l,c} \leq d_{u'} \; \forall 1 \leq u' \leq U \tag{1e}$$

Disk Budget

$$\sum_{u=1}^{U} \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{c=1}^{C} \{1 - \min[1, \max(p_{u,c}, u') - \min(p_{u,c}, u')]\} b_{nL+l}$$
$$x_{u,n,l,c} \hat{e}_{u',c} + \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{c=1}^{C} b_{nL+l} x_{u',n,l,c} \check{e}_{u',c} \leq q_{u'} \; \forall 1 \leq u' \leq U$$
$$\tag{1f}$$

Energy Budget

$$\sum_{c=1}^{C} x_{u',n',l',c} \leq 1 \; \forall 1 \leq u' \leq U, 1 \leq n' \leq N, l \leq l' \leq L \tag{1g}$$

$$x_{u,n,l,c} \in \{0, 1\} \; \forall u, n, l, c. \tag{1h}$$

Distribution Planing Problem

$$\max \sum_{u=1}^{U} \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{c=1}^{C} x_{u,n,l,c}\ \rho_{nL+l}\ \psi_{u,n}$$

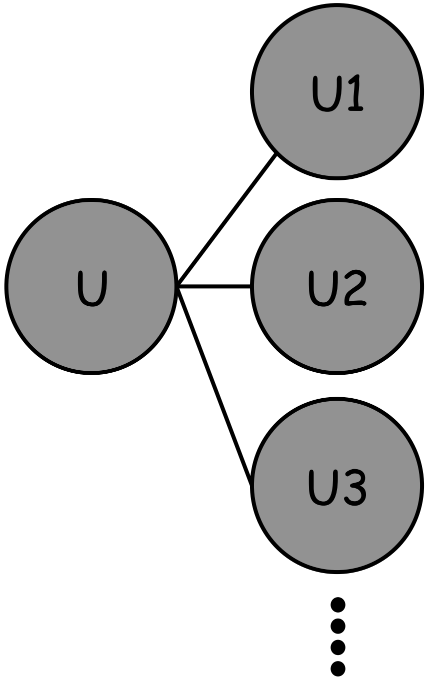
- Objective: maximize total user experience improvement
- Decision Variable: which layer of which news should we download from which contact
- Constraints:
  - Disk
  - Energy
  - Bandwidth
  - ...

Intuitions of Contact-Driven Round Robin (CDRR) Algorithm

- Get higher user experience improvement using less resources
- Allocate more resources to users who can make more contributions in the future
- Get news from the user who can make more limited contributions

Get Higher User Experience Using Less resources

- Unit = layers of news
- Each user will run into multiple users. We put the units that can be downloaded from those users in a queue Q_u={unit1, unit2, ….}
- We then sort Q_u by user experience improvement/size
  - resources = disk, energy, and bandwidth, all are proportional to the unit size
- Finally we know which news to get with the highest benefit/overhead ratio

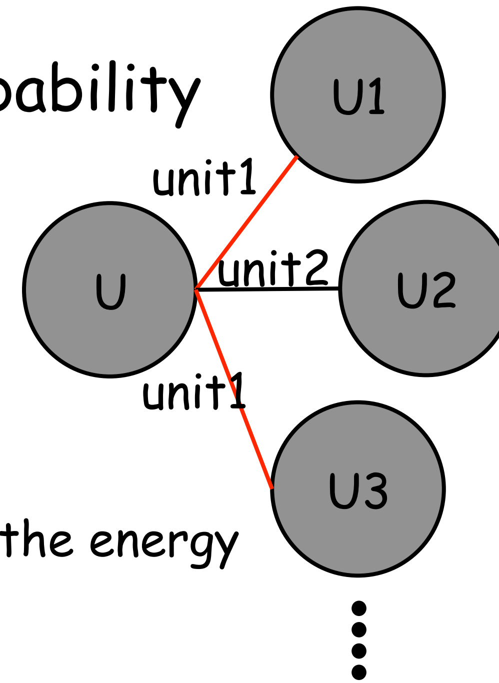


Allocate More Resources to Users Who Can Make More Contributions

- When I get a unit, the energy of contacted user (sender) will be consumed.
- Hard decisions: If I get all units in a queue, it will consume too much energy of the sender and affect other users.
- Moreover, it may not be the best decision based on our first intuition.
- Therefore,
  - for fairness and get more user experience, we do round robin to make all the users (receivers) get a unit from Q_u (sender) until all resources are consumed or all units are distributed
  - for maximize total user-experience, we allow a user who has higher probability to distribute news later to get more unit at each round ← probability is proportional to number of contact * contact duration

Get News from the User Who Can Only Make More Limited Contributions

- Now, we know which units should we get and how many units should I get at each round
  - but we have not decided where (which contact) should I get the units
- Get a unit from the user who has lowest probability to distribute news in the future
  - probability is proportional to number of contact * contact duration
- For example
  - contact durations are the same
  - number of contact of U1 = 1, U3 = 2
  - answer: I will get the unit from U1 because consuming the energy of U1 does not affect other user



Existing Algorithms for Comparisons

- Epidemic Routing [1]
  - Flood the message
- CSI-Dissemination algorithm [2]
  - Relay the message to the contact users with dissimilar mobility
  - Reduce the overhead (transmission, storage…)

[1] A. Vahdat and D. Becker. “Epidemic routing for partially connected ad hoc networks.” Technical Report CS-200006, Duke University, 2000.

[2] W. Hsu, D. Dutta, and A. Helmy. "CSI: A paradigm for behavior-oriented profile-cast services in mobile networks." Ad Hoc Networks, 2012.

Differences with Our Work

- Prefetching: we do not have specific receiver of each multimedia content
- Resource constraints: we consider resource limitations of mobile devices

Simulation Setup

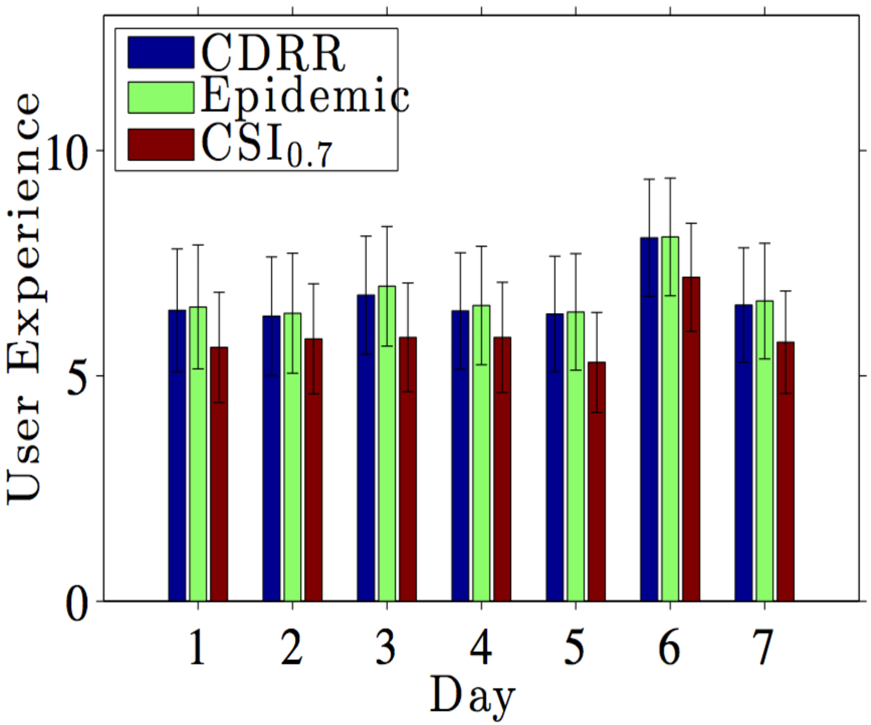
- We implement the two baseline algorithms and our algorithm in the simulator
  - Epidemic
  - CSI
  - CDRR
- We use a trace to simulate real scenarios
  - Collected in Beijing
  - 4-year traces
  - 178 users
  - GPS trajectory
- We preprocess the traces to drive our simulator

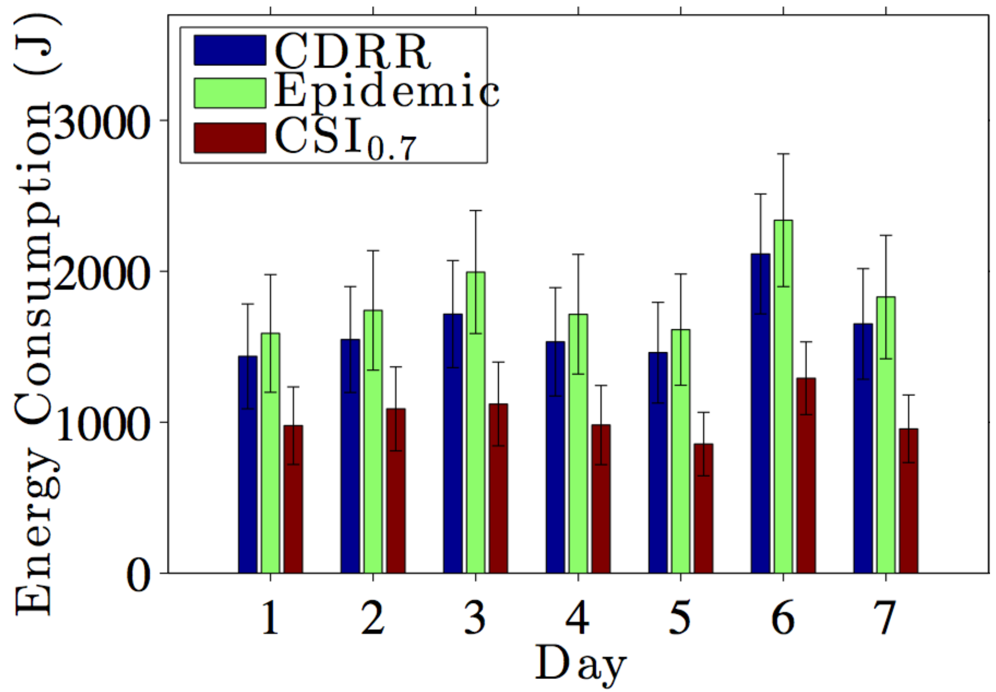
Performance metrics

- User experience: the average user experience of all the watched news reports
- Energy consumption: the average energy consumption of mobile devices
- Disk efficiency: the ratio of user experience and disk consumption
- Used disk space: the amount of used disk space
- Missed ratio: the fraction of unavailable news reports among all the user demanded ones
- Watched unit: the number of watched news reports among all the downloaded ones
- Received unit: the number of received news reports

Simulation Results - Unlimited Resources

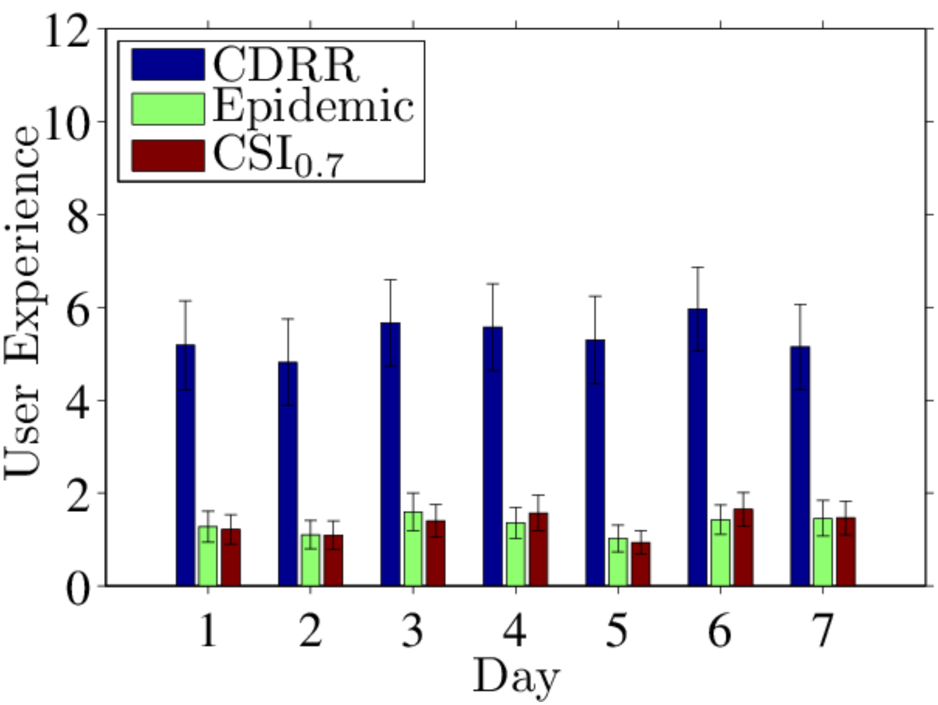
- Our CDRR algorithm can achieve 99% approximation factor, while epidemic giving highest user experience
- But unlimited resources are unrealistic!

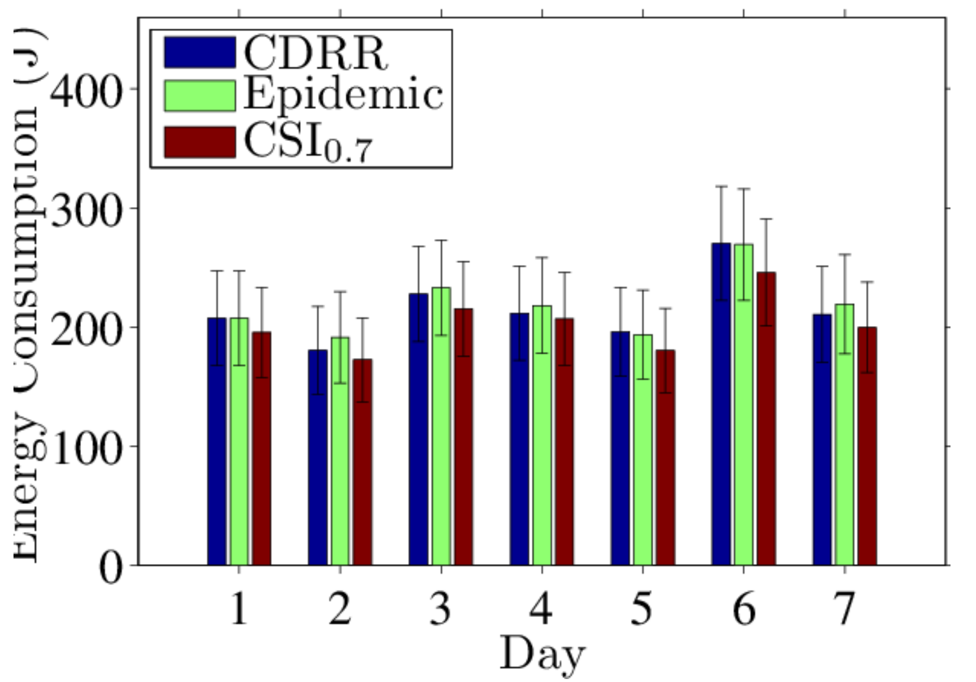




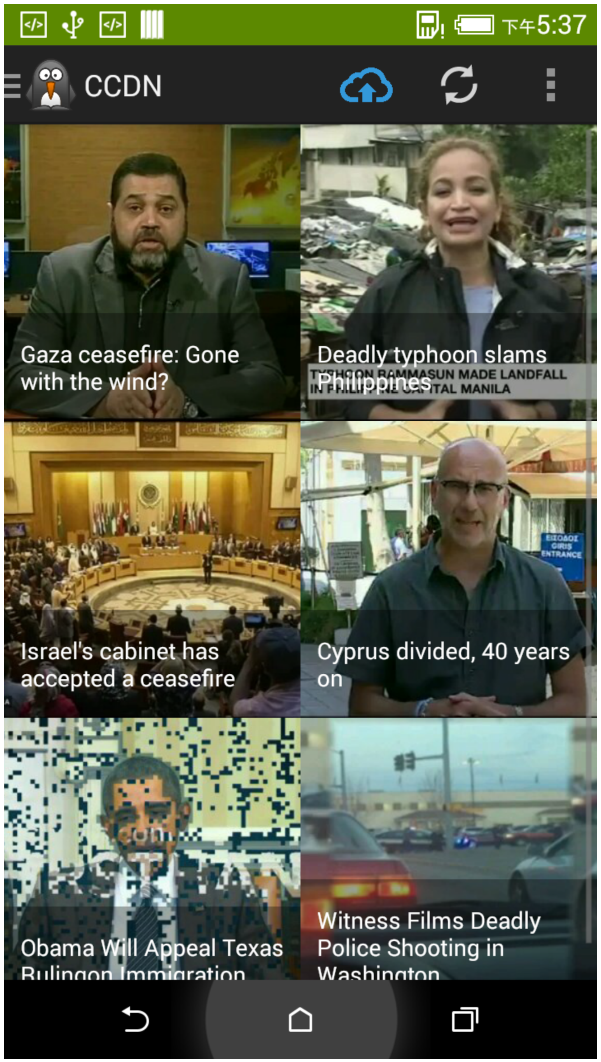
Simulation Results - Realistic Scenario

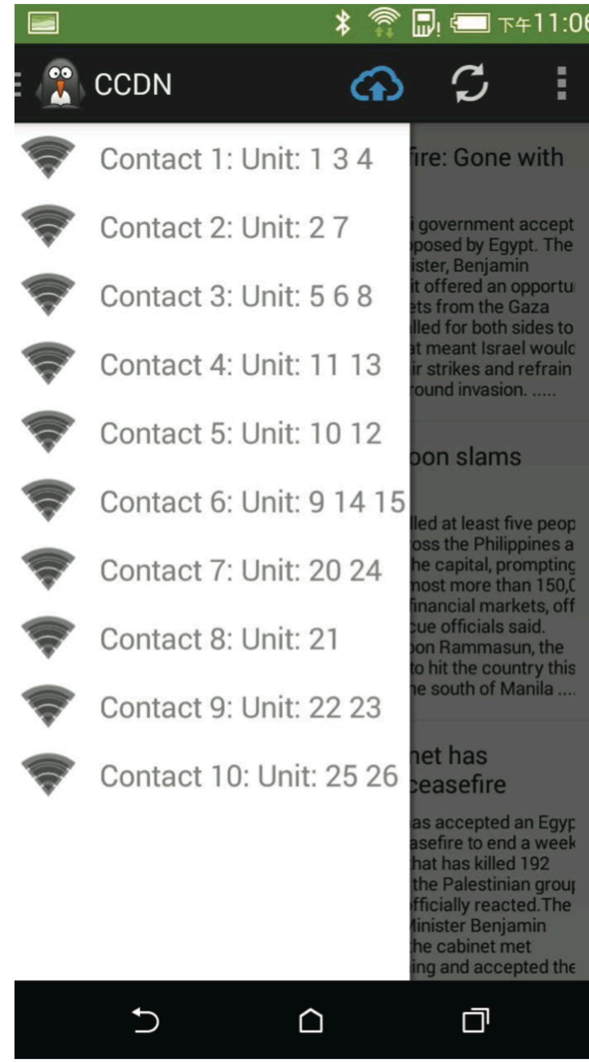
- Disk Budget = 250 MB
- Energy Budget = 2000J
- In realistic scenario, CDRR algorithm outperforms others by up to 4 times, while the energy consumption is only higher than others up to 15%





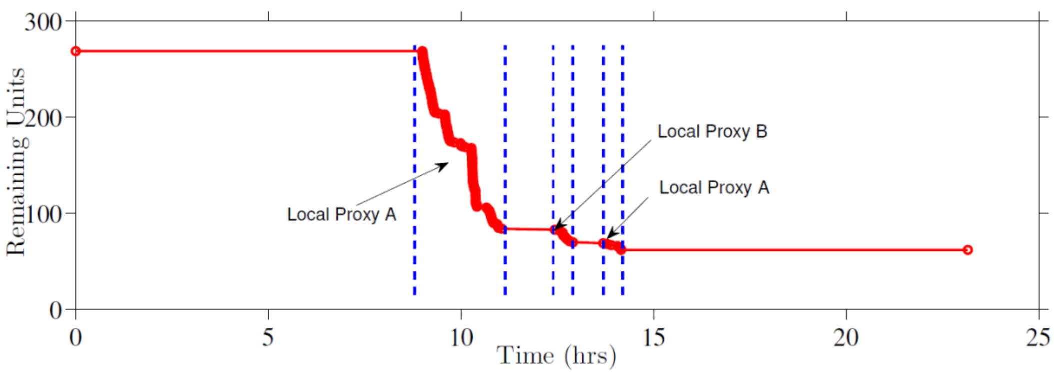
Prototype - Client

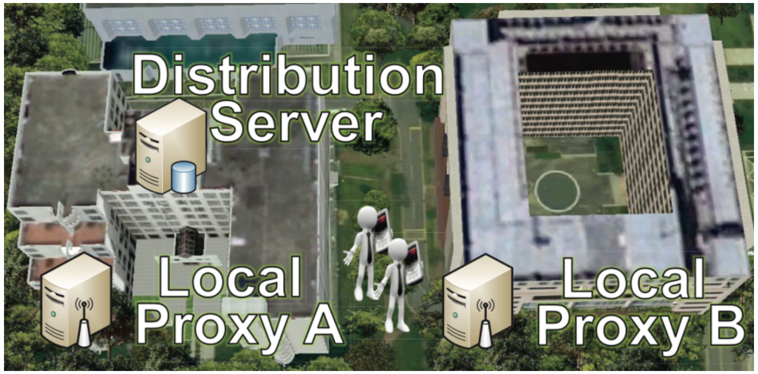




Preliminary Experiments

- Collect 3 days social profiles for creating the distribution plan of a user
- Running 24 hours experiments with
  - 2 local proxies, 1 distribution server and 54 news with 5 layers
- We clarify our prototype can follow the plan to download 77% of the multimedia content





Questions?

Contact me at chsu@cs.nthu.edu.tw anytime