Improving Localization and Energy Efficiency of Smartphone Applications

Swadhin Pradhan

Supervisor: Niloy Ganguly

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

MS Defence Seminar, January 2, 2015

Outline

Impact of Smartphones

- **One in every five** people in the world carries a smart phone. (BusinessInsider study)
- Smartphone app industry is growing (expected to reach **$**40 **billion** by 2015).
- By 2019 there will be **5.6 billion** smartphones carried by people. (Ericsson study)
- But, smartphone has many issues ...

Issues in Location Sensing in Smartphone

GPS is problematic

- 10m error, Multi-path effect.
- High Energy Cost.
- Not available in Indoor.

Indoor Localization

- Infrastructure based solution not good enough. (RFID based, FM based, GSM tower based, WiFi AP based etc.)
- SLAM from robotics (kalman filter, coupled sensing etc.). But, needs costly sensor, precise movement etc.

Outdoor Localization

Assisted GPS, Collaborative GPS etc. But, still has GPS issue .

Issues in Smartphone Energy Consumption

Poor apps draining energy

- Display, Network, and CPU are main components of Energy Drain.(**MobiCom '13**)
- Poorly written apps can sap 30% to 40% of a phone's battery. (**IMC '09**)

Cellular Energy Consumption is a major part

- Network intensive applications are increasing (69% of the apps are cloud based)
- Different background services running intermittently and waking up the network card for a small duration. (**WWW '12**)

Objectives of the Thesis

Efficient Localization in Smartphones

This shows that how localization can be improved using the different sensors in smartphones by detecting environmental signatures.
(*S. Pradhan et. al., COMSNETS 2014*)

Energy Efficient Cellular Radio Usage in Smartphones

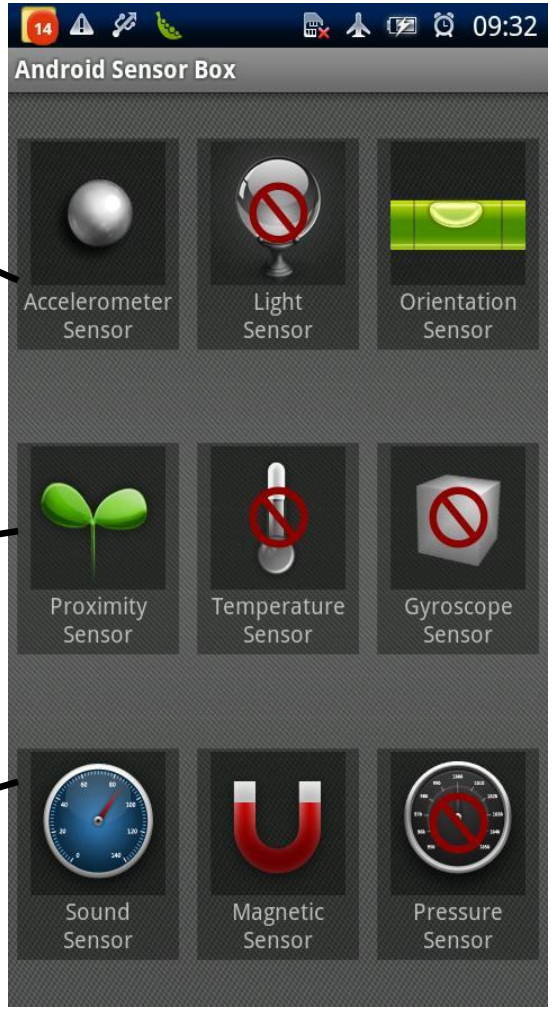
This attempts to solve the problem of energy drain in cellular radio by intelligently scheduling of concurrent requests.
(*S. Pradhan et. al., COMSNETS 2015*)

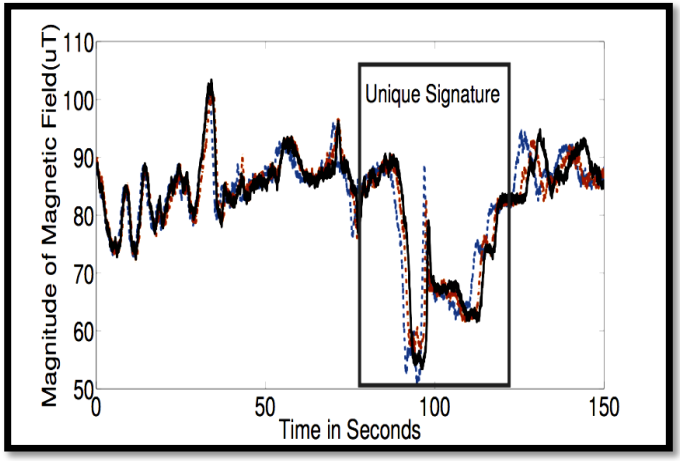
Developing Prototypes as proof of concepts

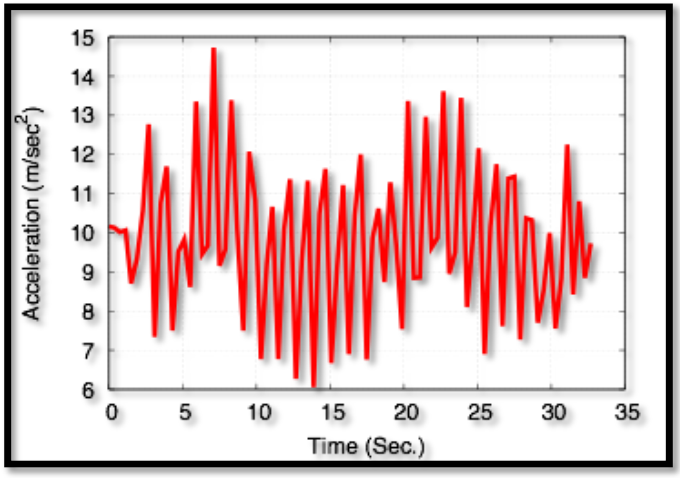
We have developed a retail app called **RetailGuide** and a virtual sign creator app called **SignFinder**.

Improving Location Sensing

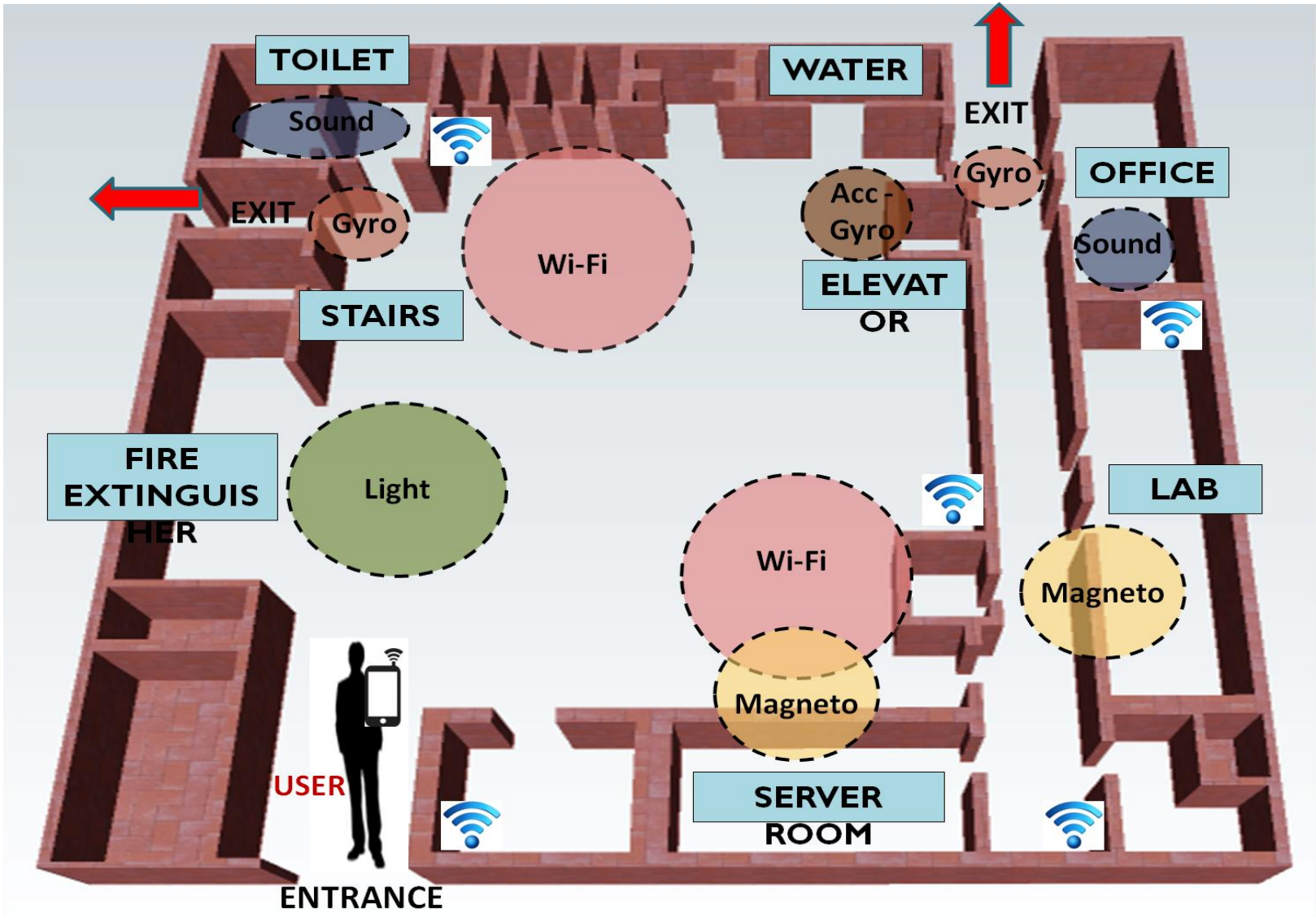
Landmarks = Sensors + Location





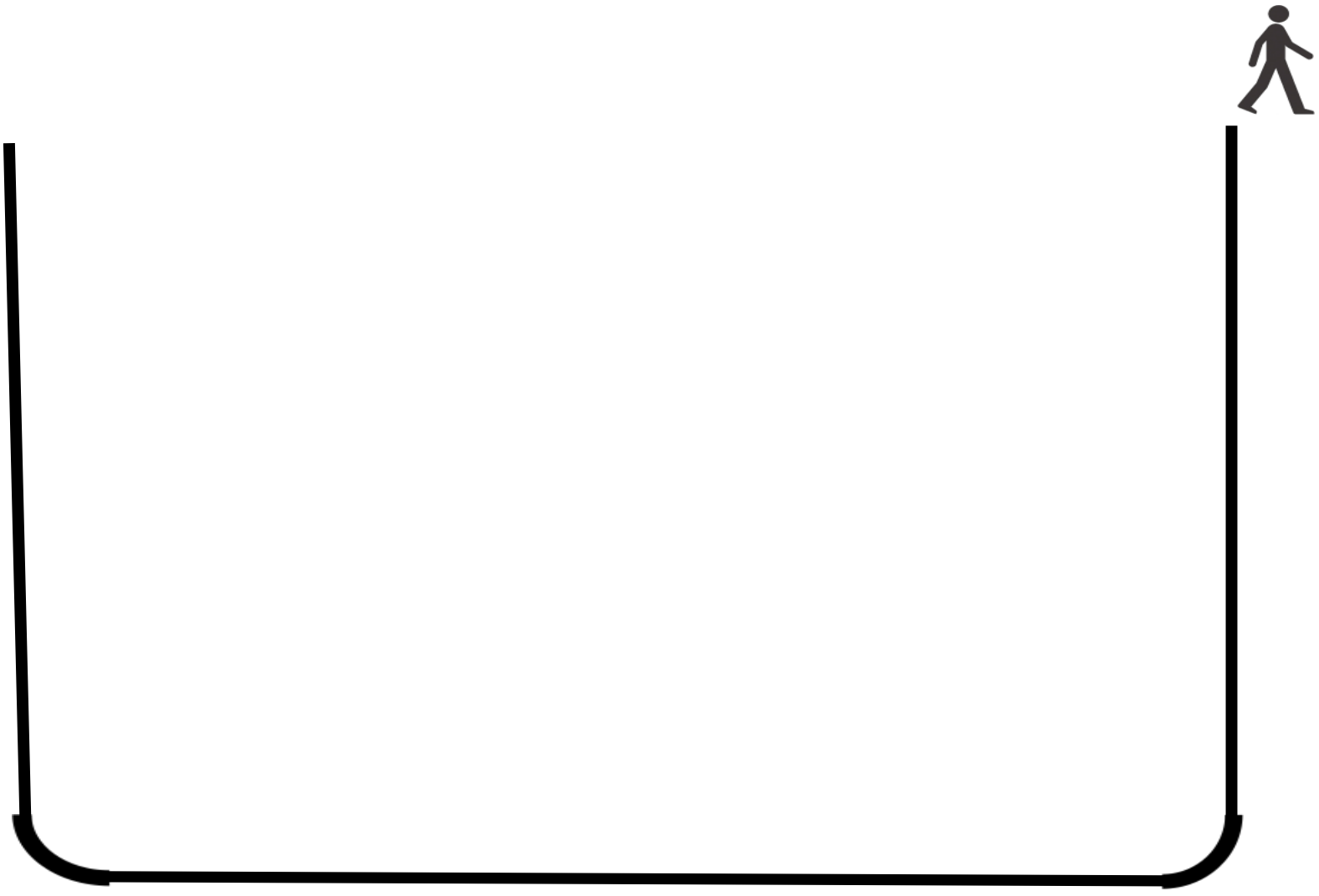


Location Annotation via Landmarks

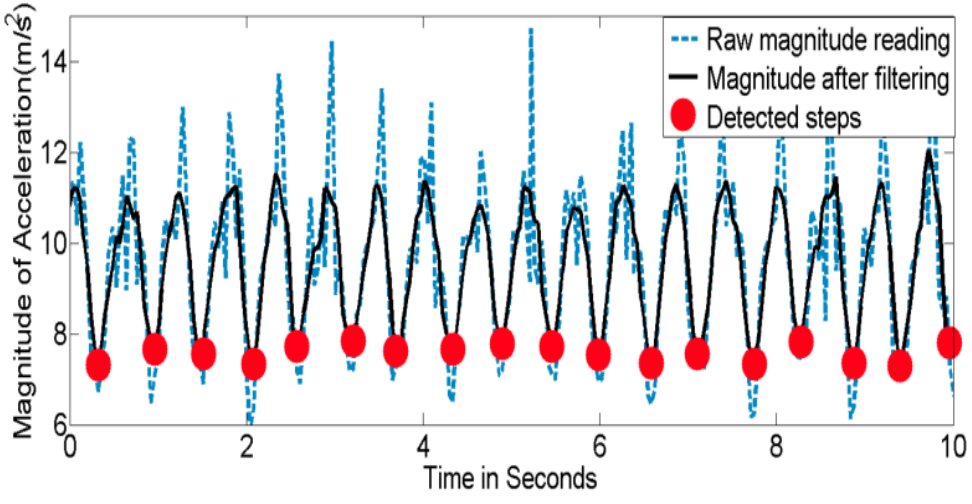


Localization using Landmarks

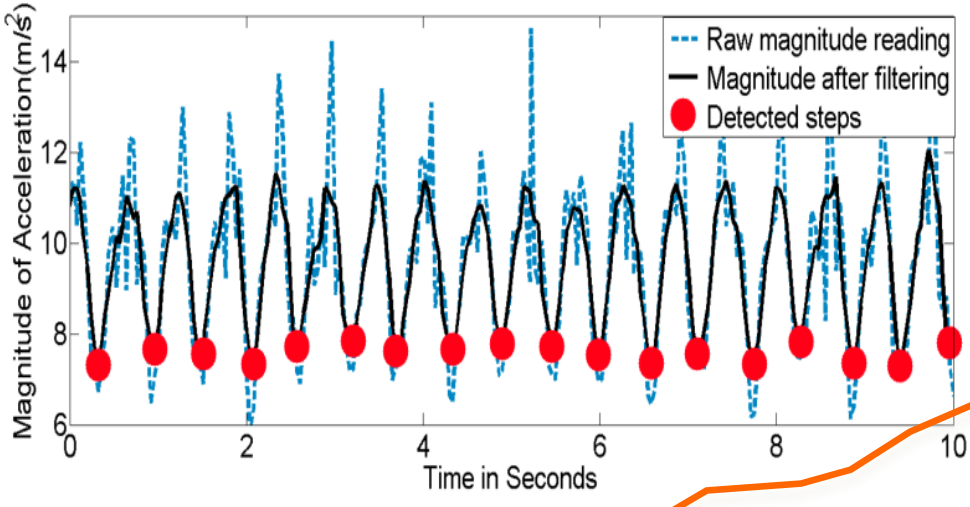
Localization using Landmarks



Localization using Landmarks

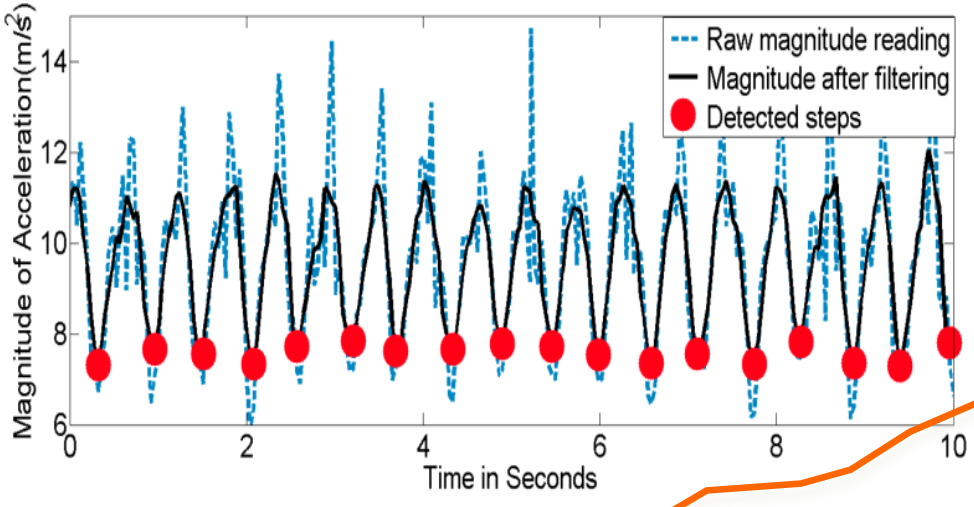


Localization using Landmarks

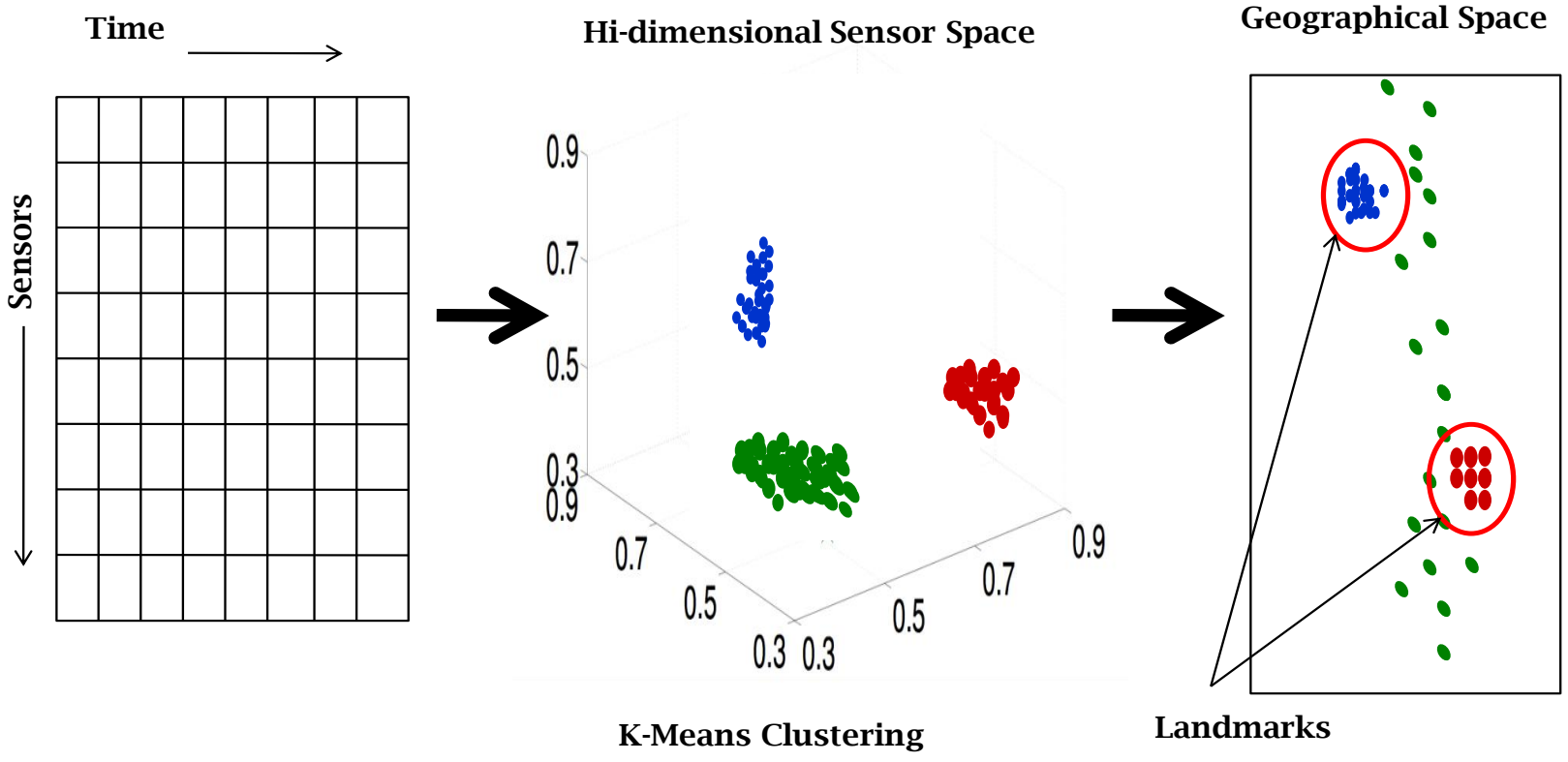


Estimated path

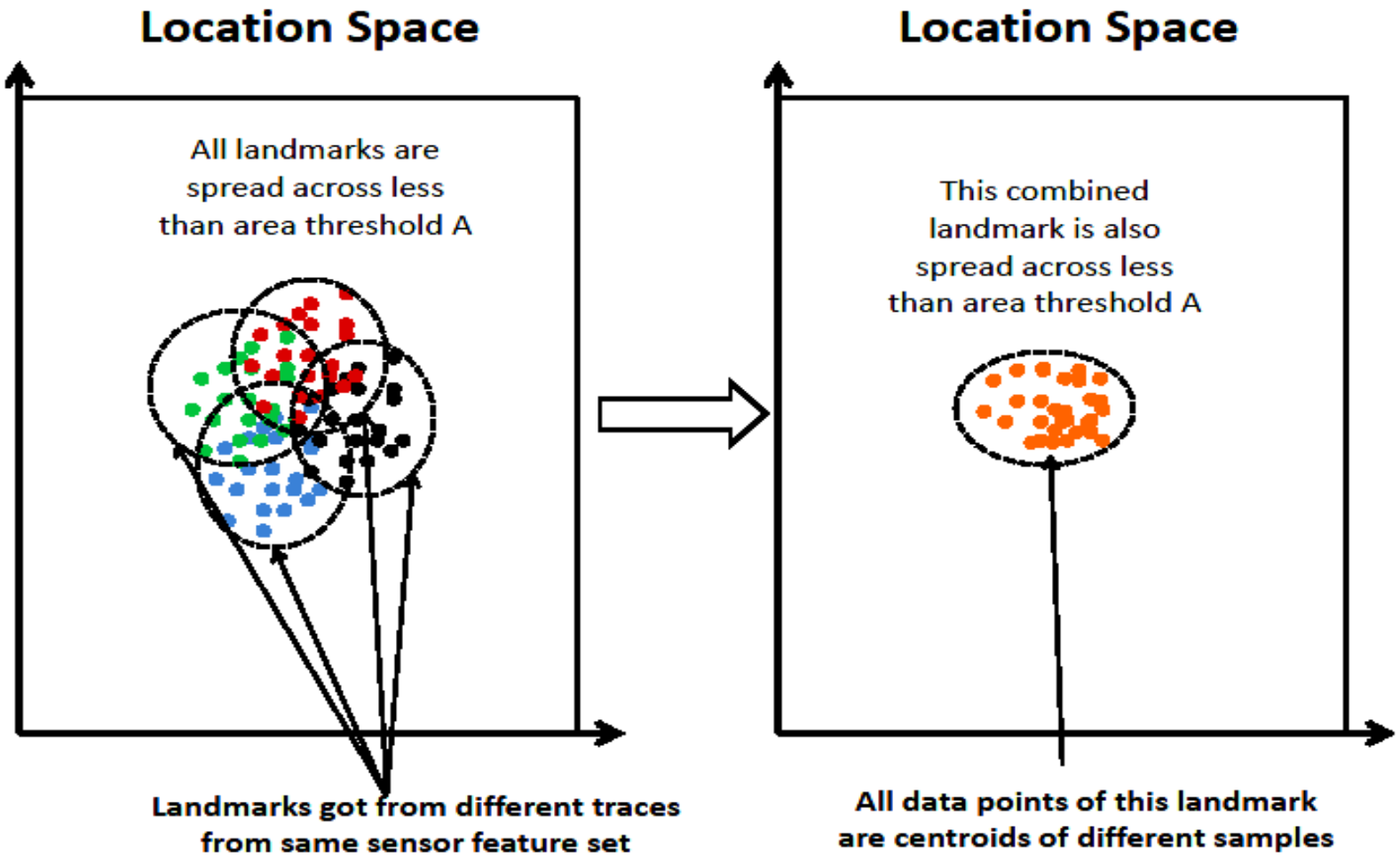
Localization using Landmarks



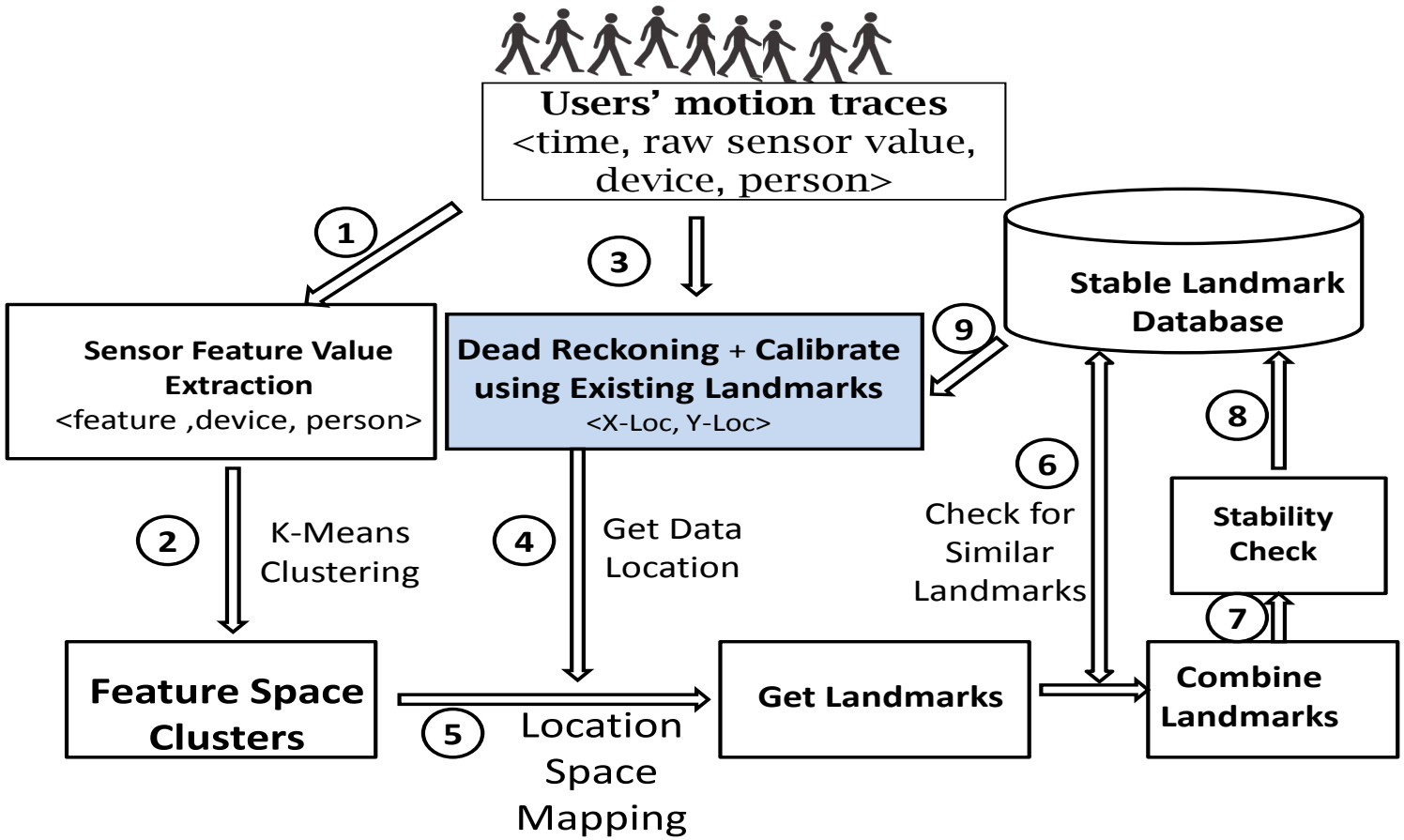
Pruning Virtual Landmarks



Finding Stable Virtual Landmarks



Architecture of the proposed system



Experimental Setup

Devices, Users, Time

- Samsung Galaxy S2, Samsung Galaxy S3, and Google Nexus
- 6 Users, Day and Night

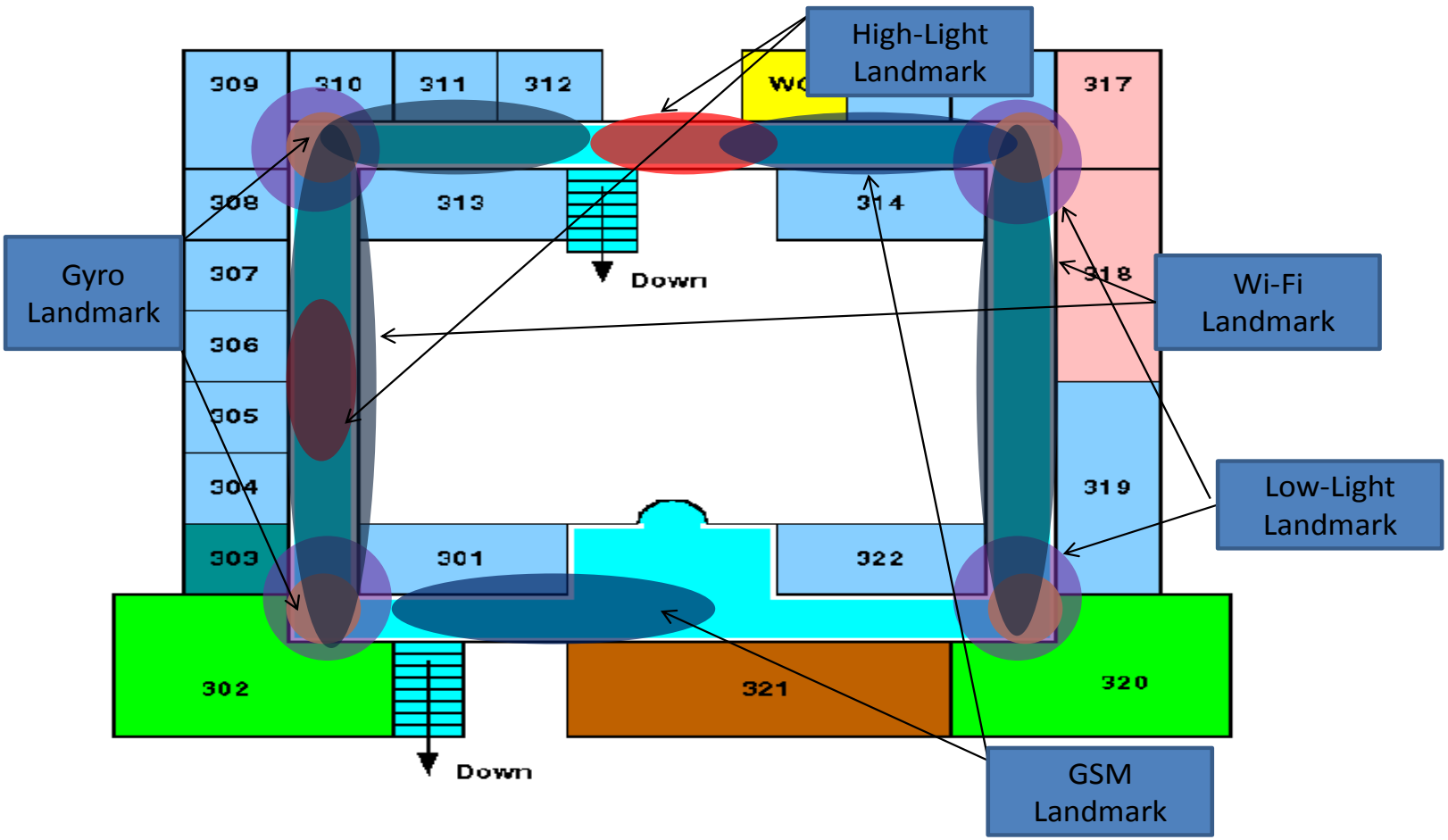
Platform

Android 4.2.x

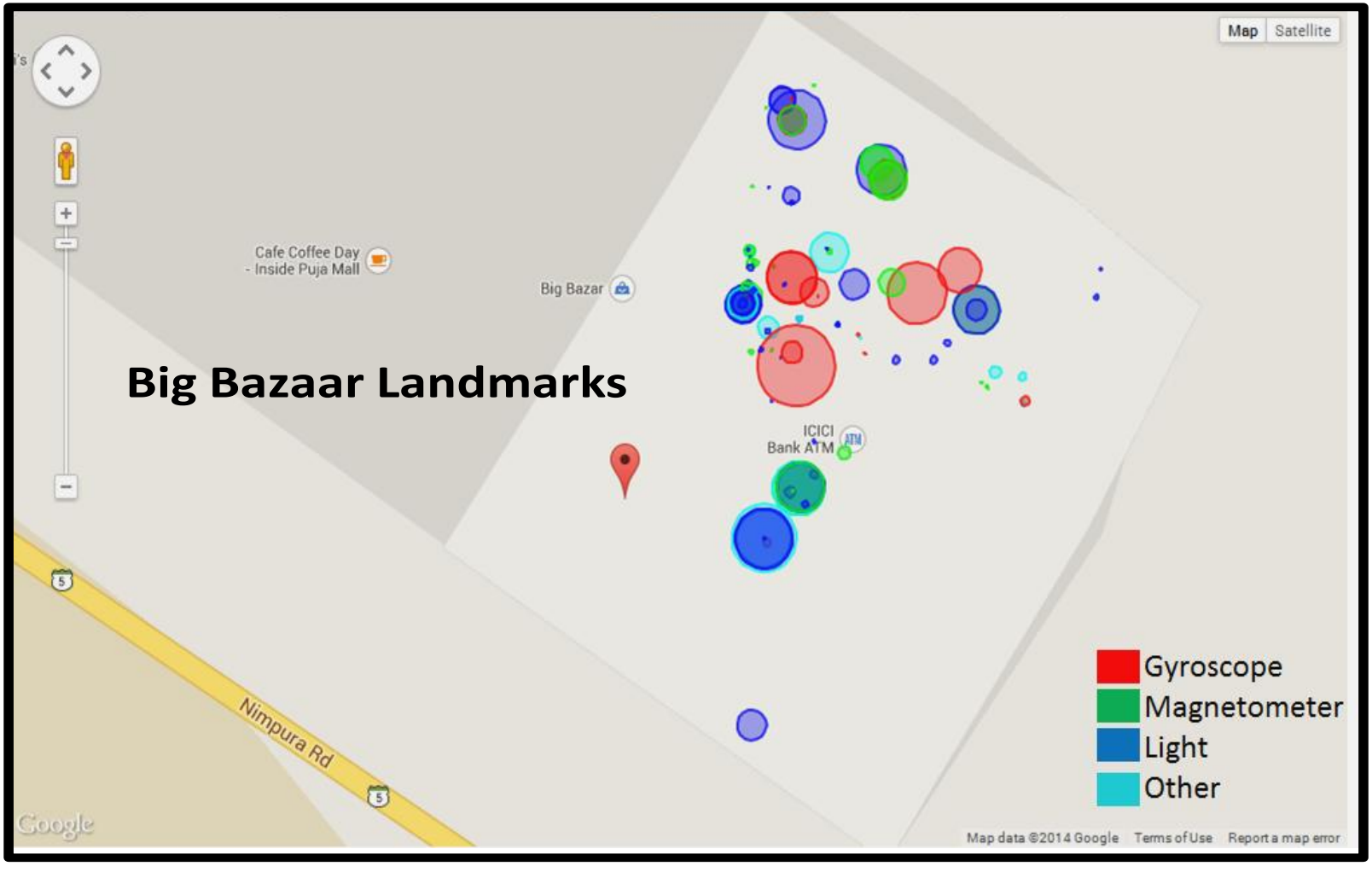
Places

- Department and Tech Market (to evaluate Indoor and Outdoor Scenario)
- Department (Pseudo Mall Scenario) and Big Bazaar (Actual Mall Scenario) [*Market Experiment*]

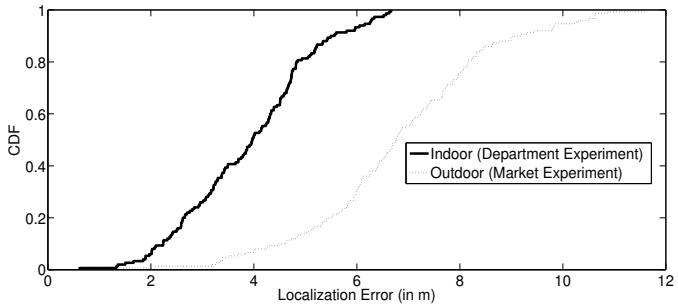
Landmarks at Department



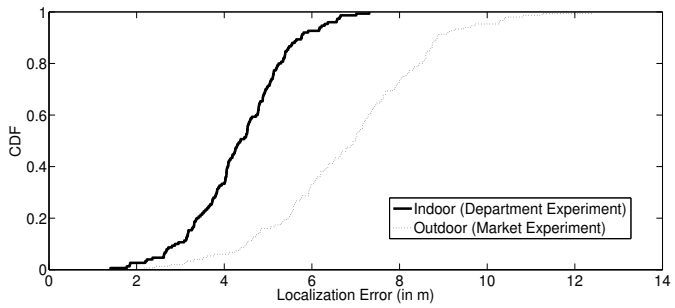
Landmarks at Big Bazaar



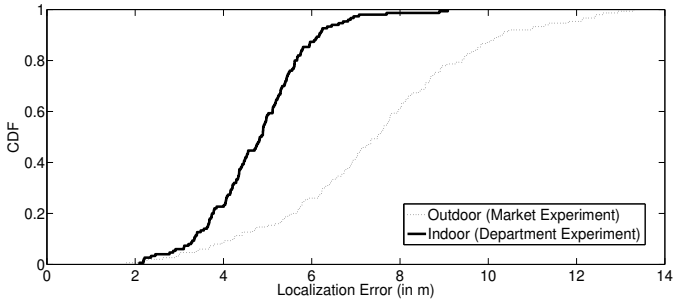
Localization Error



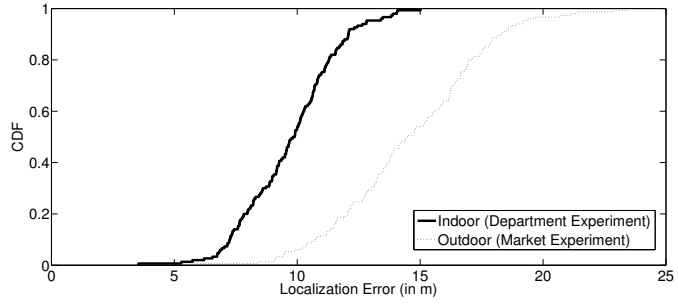
(a) Error without changing any



(b) Error with changing person



(c) Error with changing time

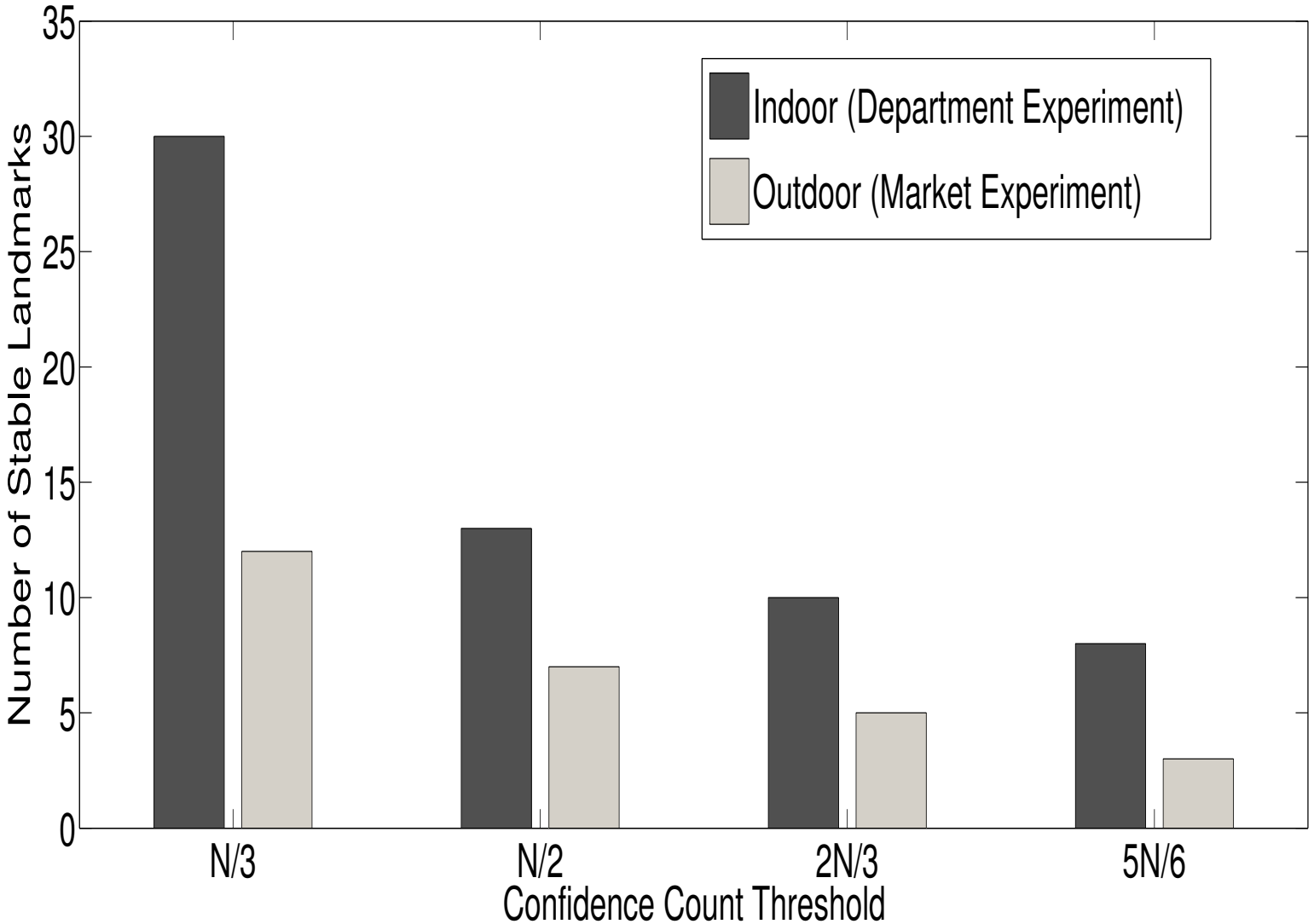


(d) Error with changing device

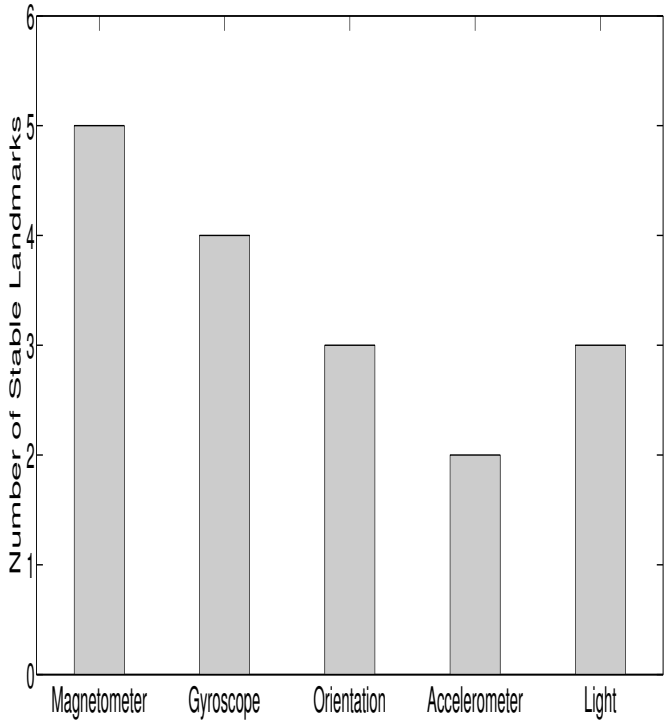
Impact on Localization Error

Device >> Time > Person

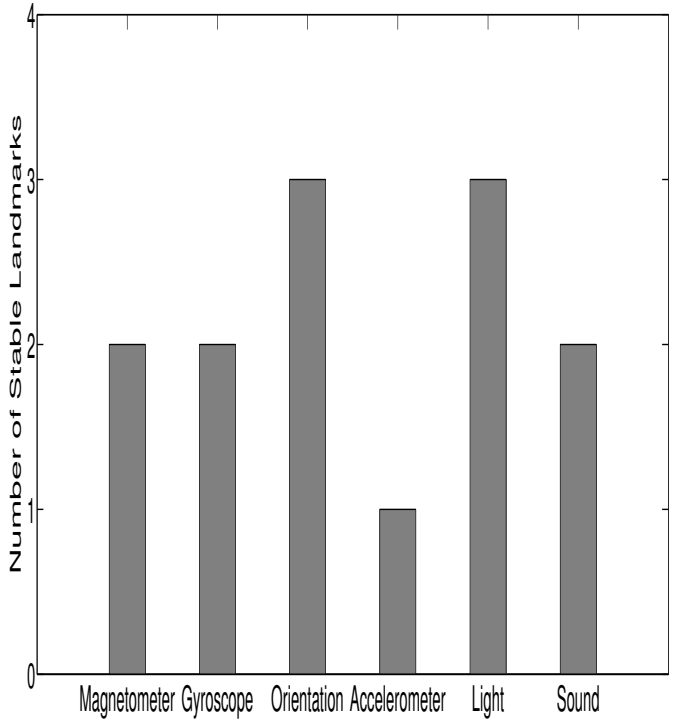
Landmarks in Indoor and Outdoor



Sensorwise Stable Virtual Landmarks in Different Experiments

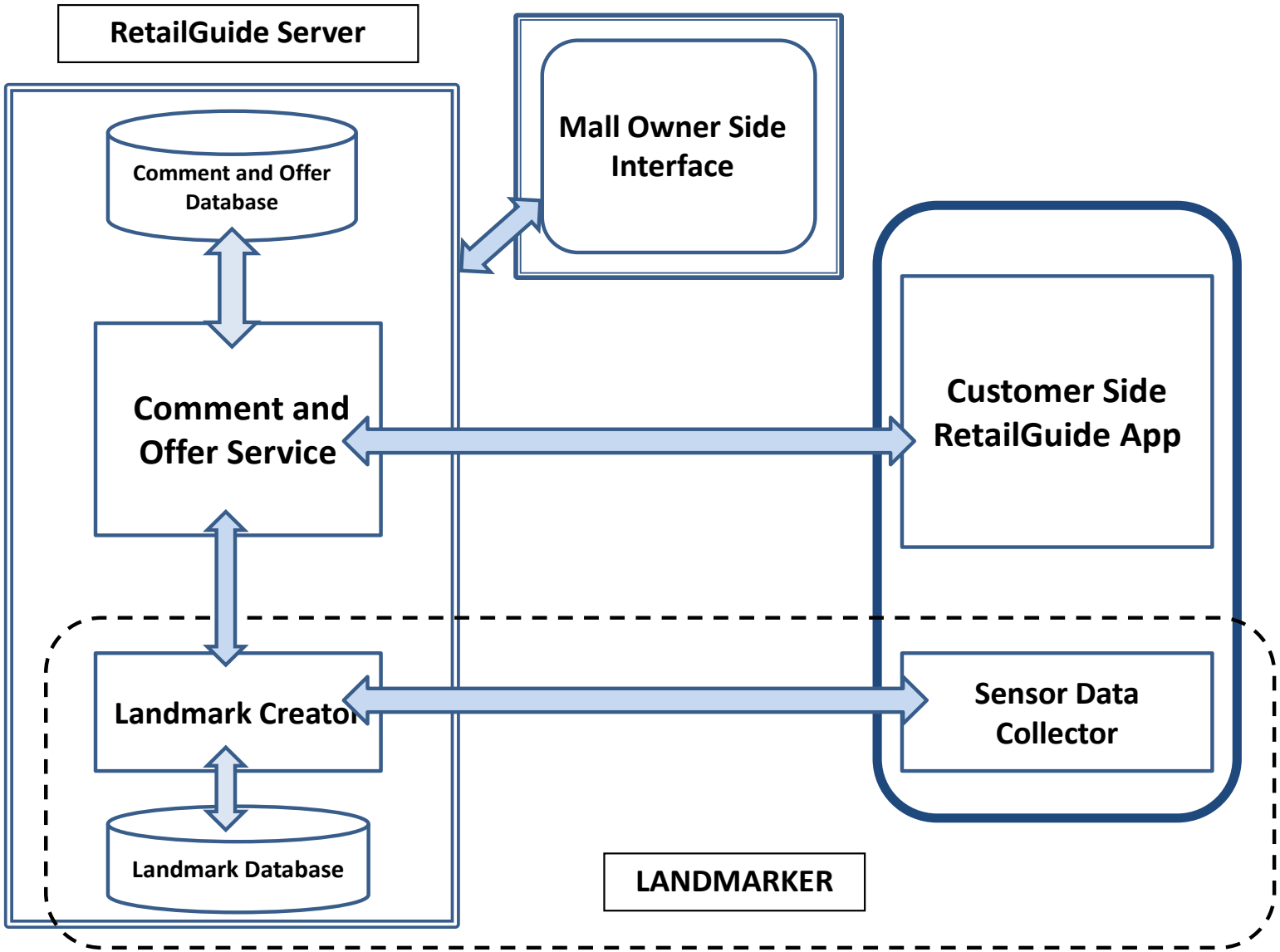


(e) Department Experiment



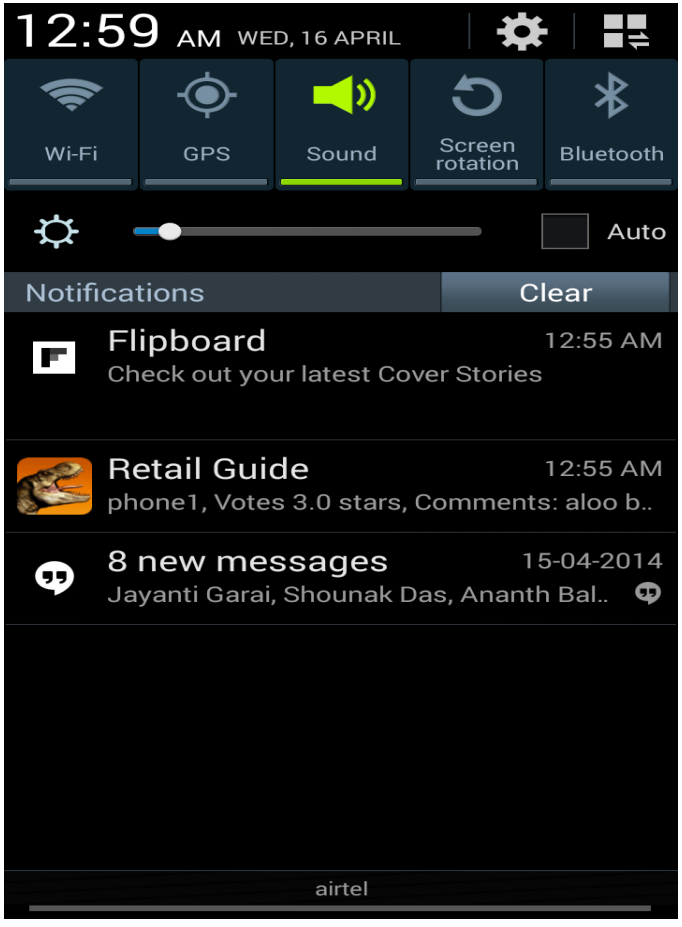
(f) Market Experiment

RetailGuide Implementation

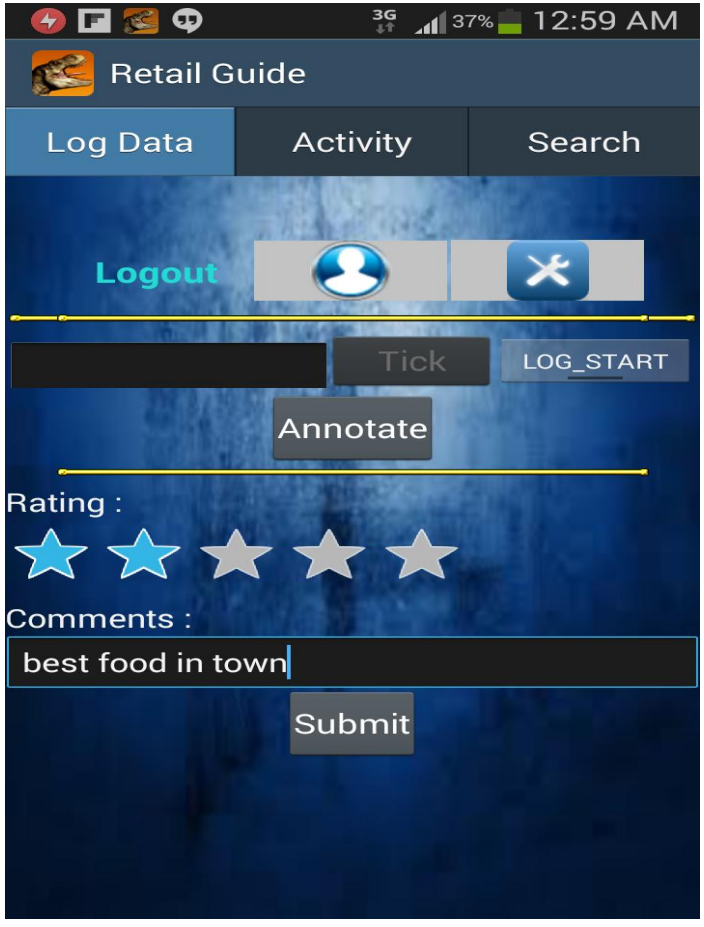


Different Applications based on Virtual Landmarks

RetailGuide : Android App



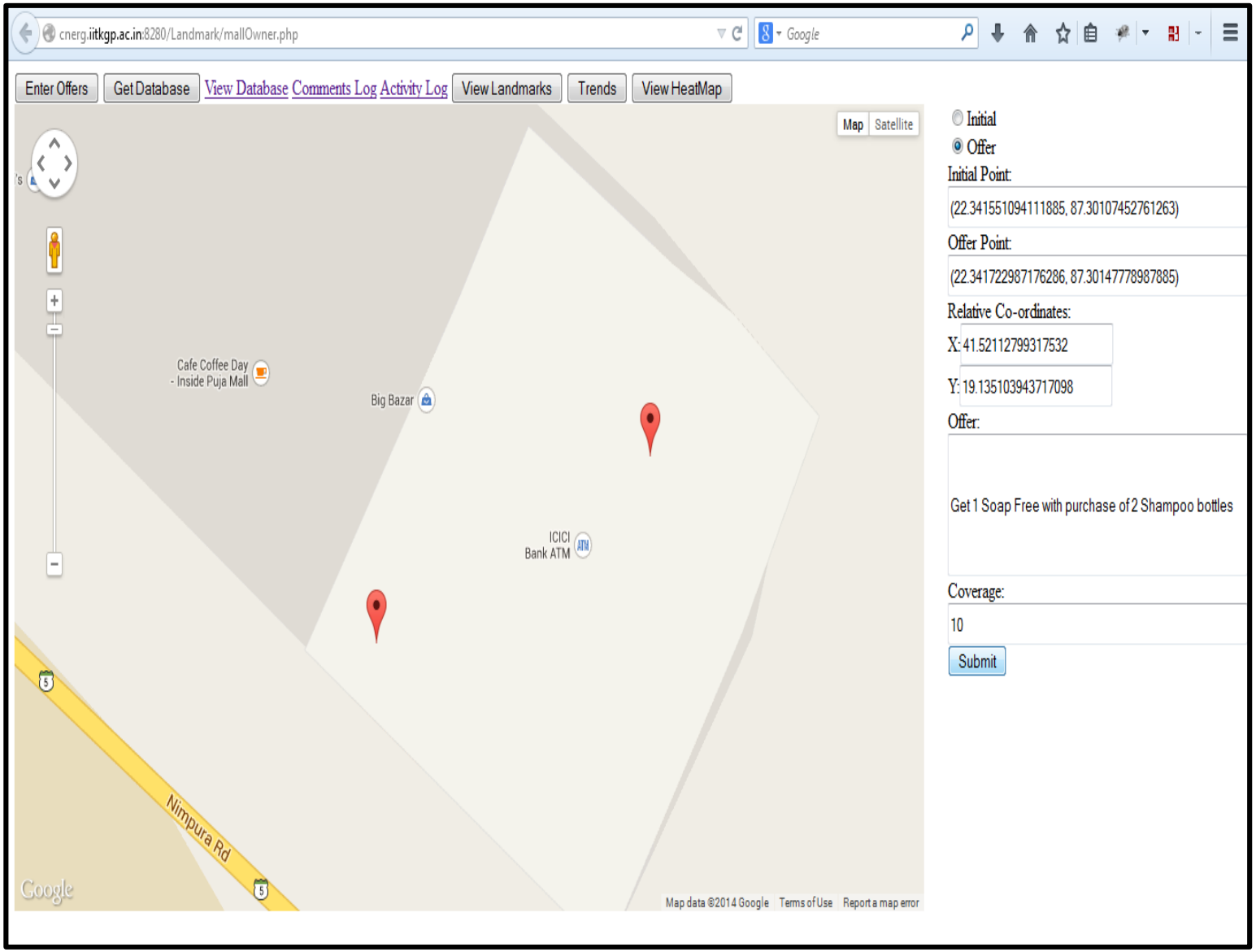
Pushed Notification of Comments



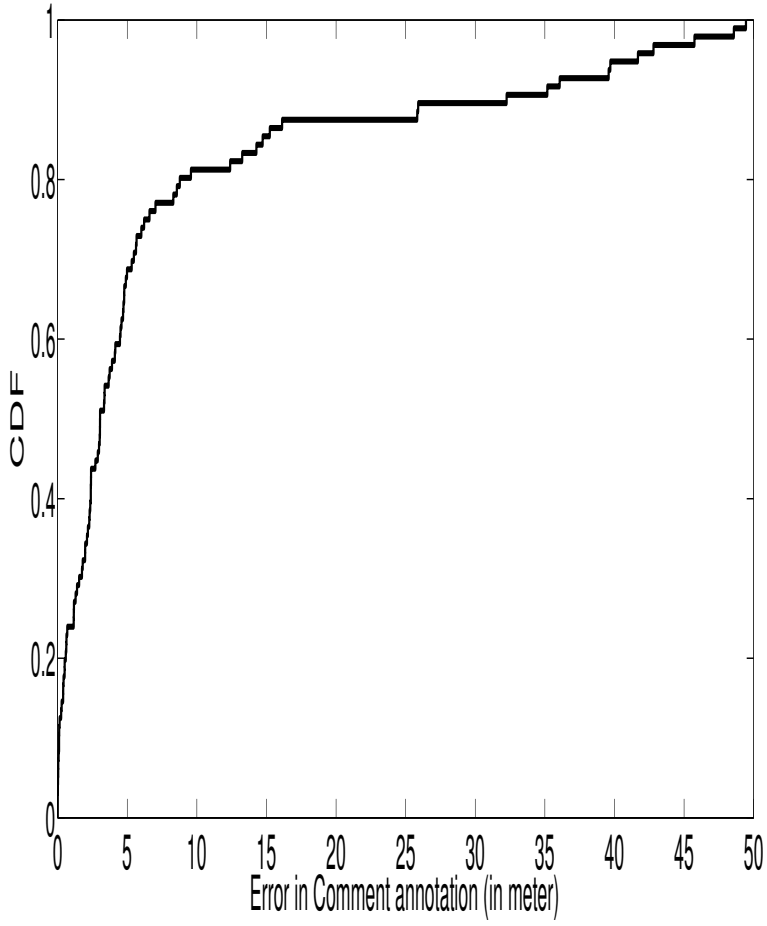
Users' Rating and Commenting Interface

Different Applications based on Virtual Landmarks

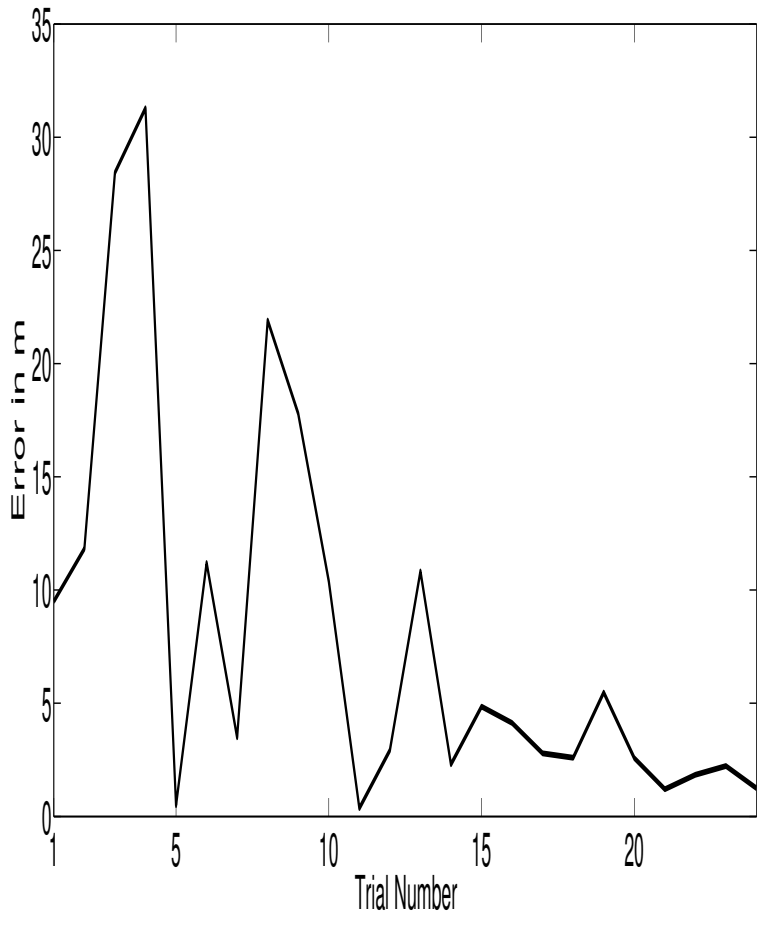
RetailGuide : Admin-side Web Interface



Comment Tagging Accuracy in *RetailGuide*



(g) CDF of Accuracy of Tagging of Comments to particular locations



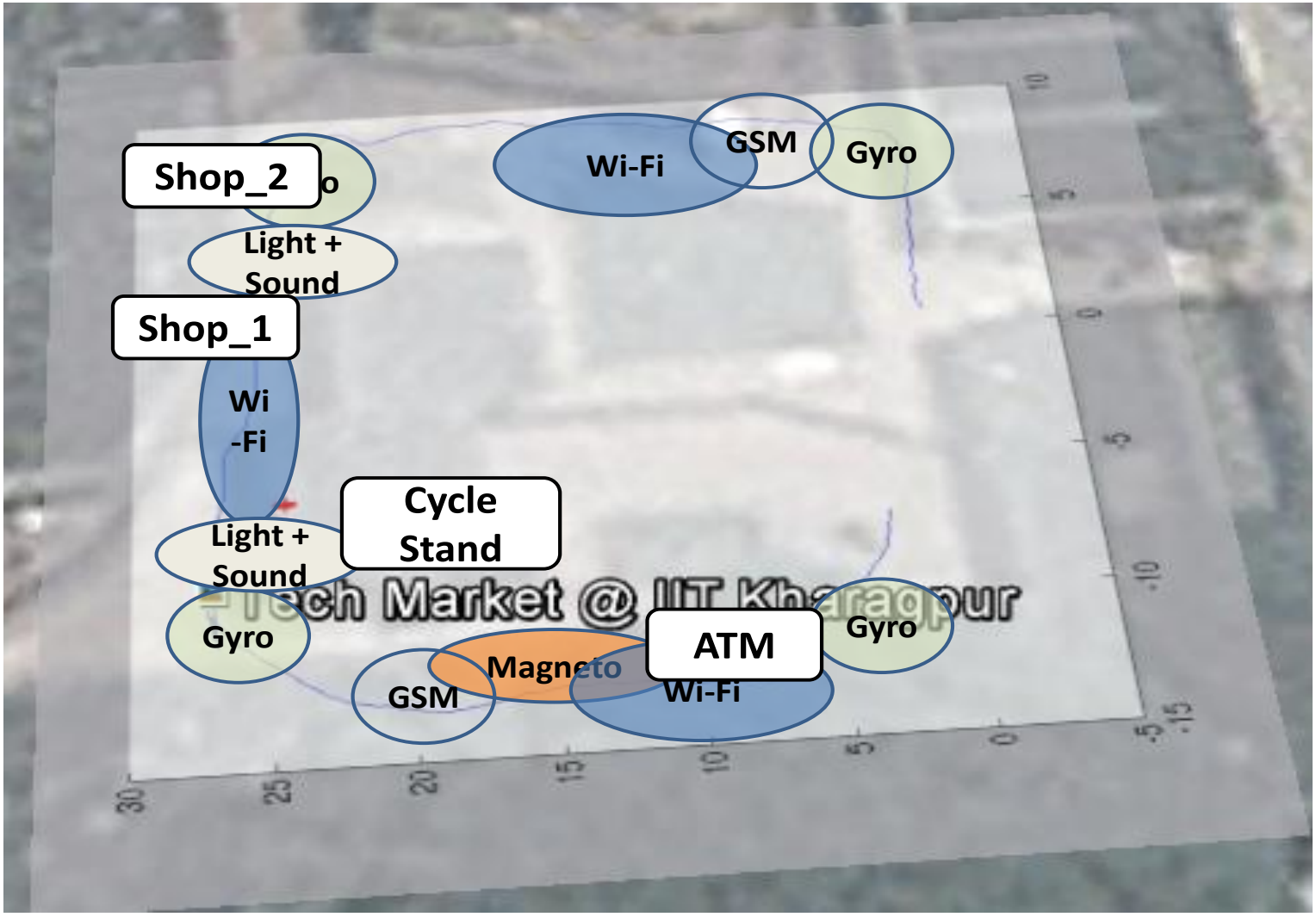
(h) Change of Accuracy of Tagging of Comments with Trials

Signfinder : Virtual Sign Creator App

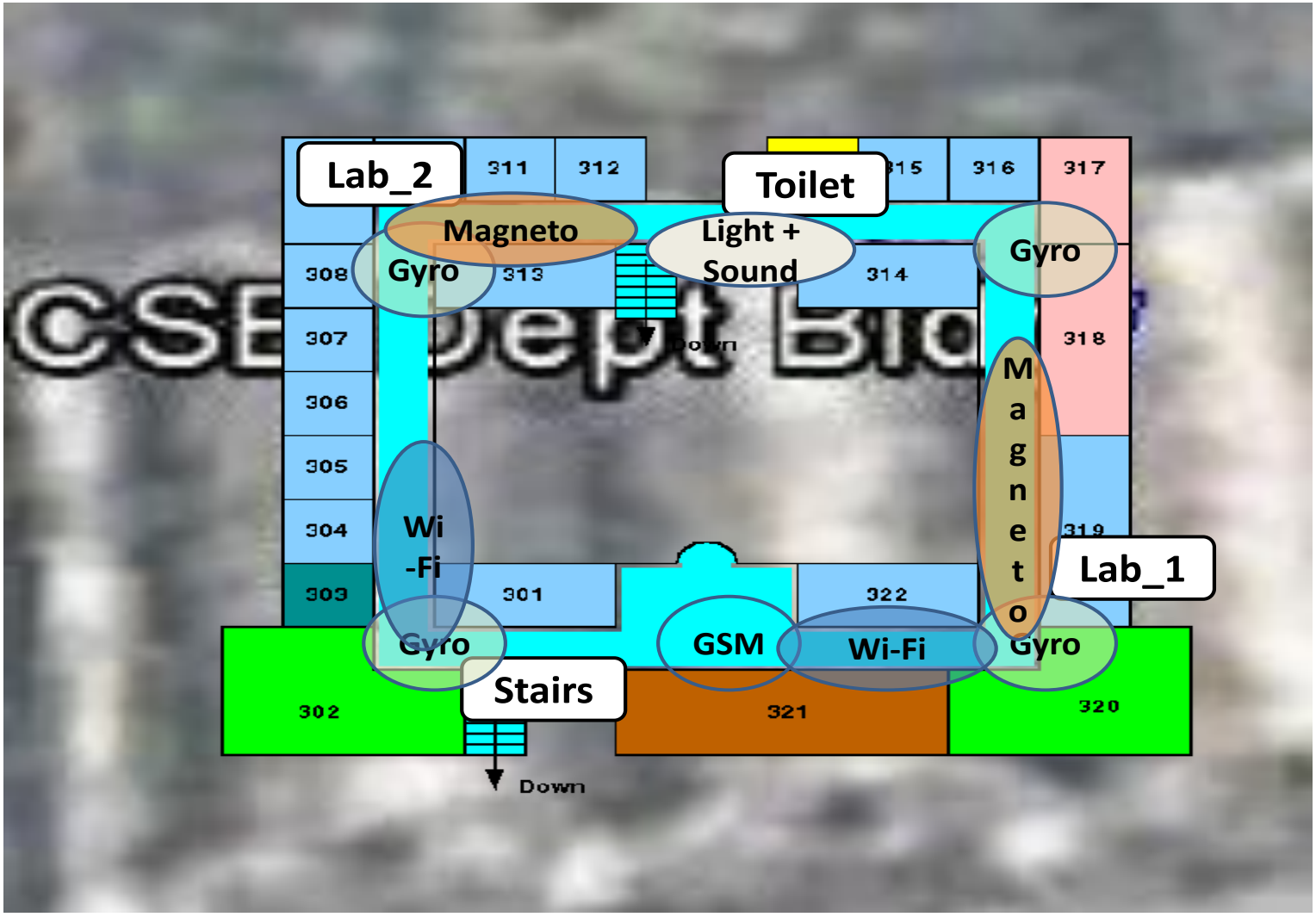
Signs with corresponding Sensor Signature Vectors

Signs	Sensor Signature
Toilet	High Sound, High Light, Low Wi-Fi, High GSM, High Relative Humidity
ATM	Low Sound, High Light, Low Wi-Fi, High Magnetic Signature, Low Relative Humidity
Lab	High Sound, High Light, Low Wi-Fi, Low GSM, High Magnetic Signature, Low Relative Humidity
Stair	High Sound, Low Light, Low Wi-Fi, Low GSM, High Gyroscope Signature Change
Shop	High Sound, Low Light, High Wi-Fi, High GSM, Low Magnetic Signature
Cycle Stand	High Sound, High Light, High Wi-Fi, High GSM, High Magnetic Signature

Virtual Signboards in Outdoor Area

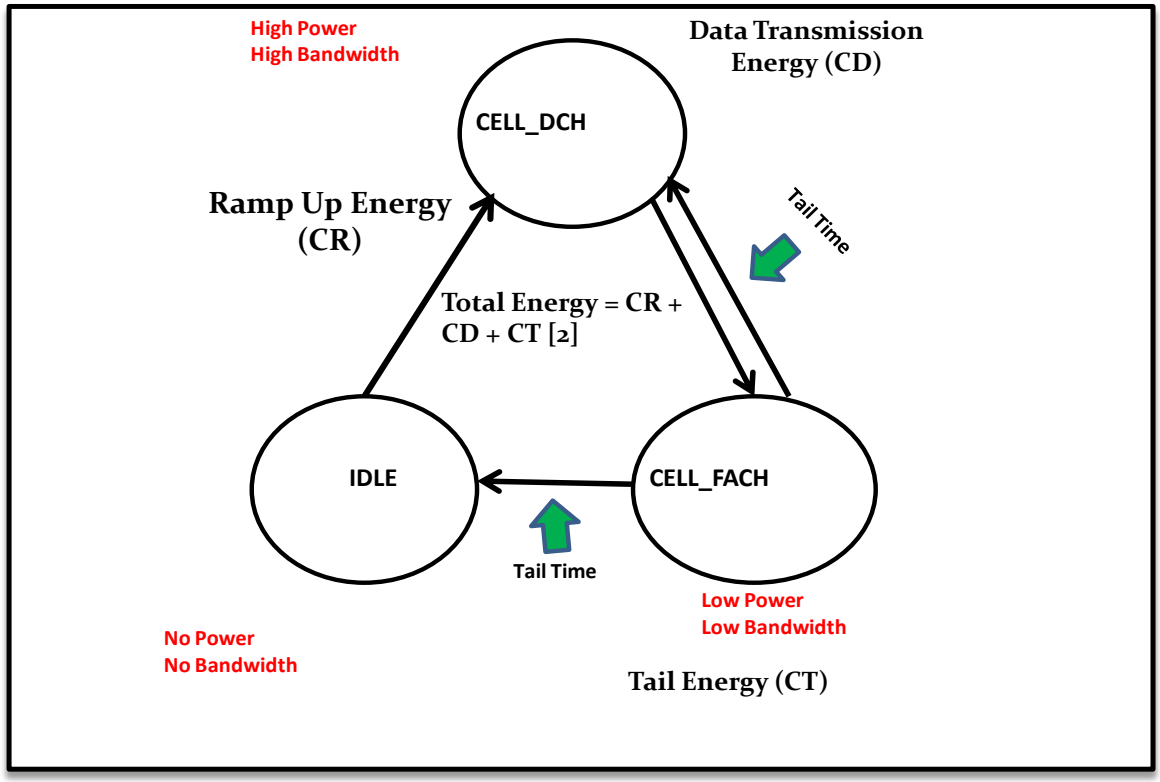


Virtual Signboards in Indoor Area



Improving Cellular Radio Energy Consumption

3G Cellular Radio Energy Model



Total Energy Consumption = CR + CD + CT ,

where CR is the ramp up energy (**IDLE to CELL DCH**),
CD is the data transmission energy,
and CT is the tail energy (in CELL FACH).

Related Works

Tail Time Prediction based Solutions

TOP (ICNP ’10), RadioJockey (MobiCom ’12) etc.

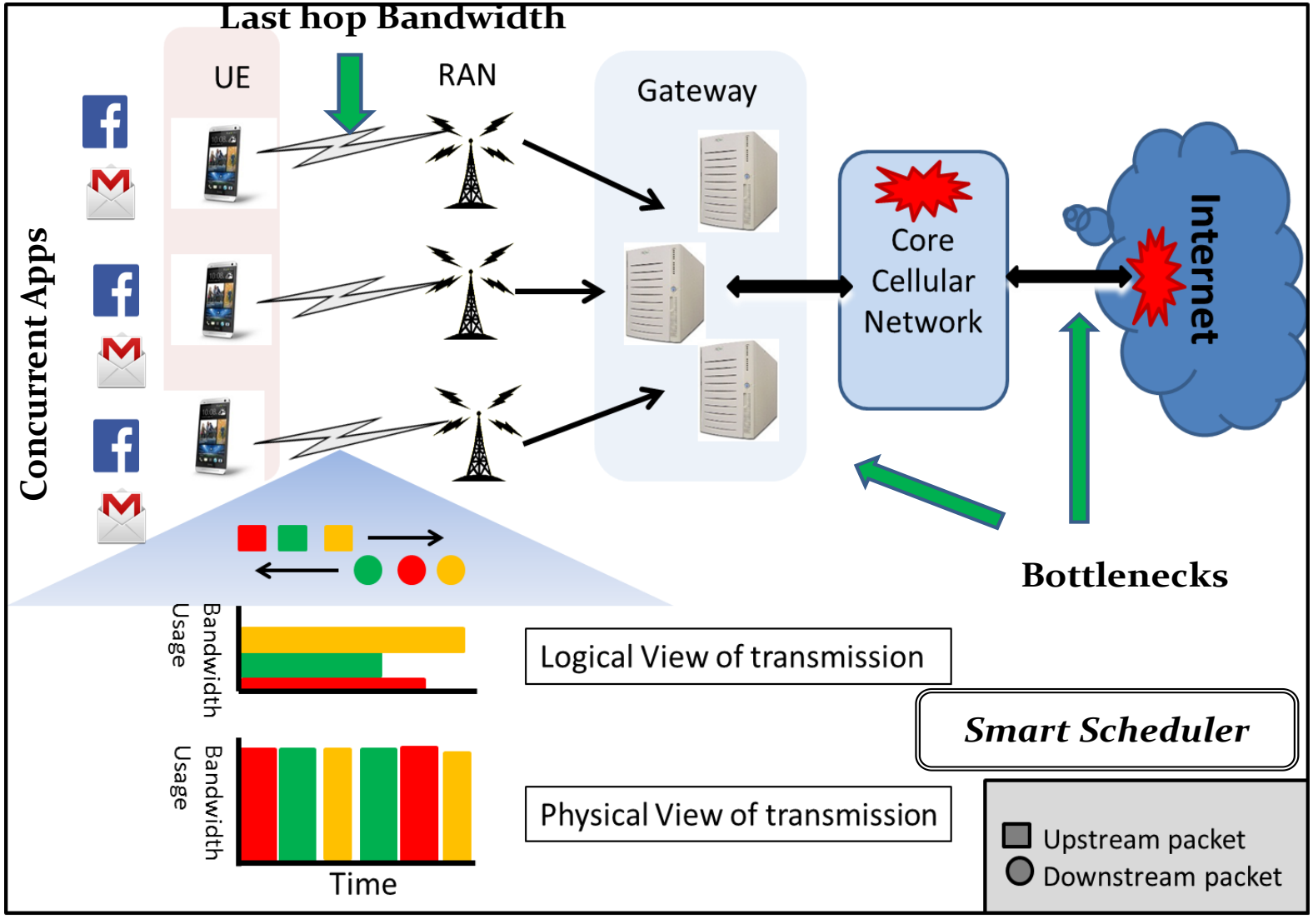
Batching and Delayed Transfer based Solutions

TailEnder (MobiCom ’10)SmartEmailSync (MobiSys ’13)TailTheft (MobiArch ’11) etc.

Proxy Based Solutions to tackle bottleneck

CatNap (MobiSys ’10) etc.

Total View of App Connectivity



Problem Description

- A set of applications, running in parallel, request for network resource intermittently.
- Depending on application, **a flexibility or slack time** is allowed to schedule each packet.
- Requests from the same application cannot be triggered simultaneously.
- Total bandwidth consumption by all the scheduled requests should be less than the available channel bandwidth.

Problem Objective

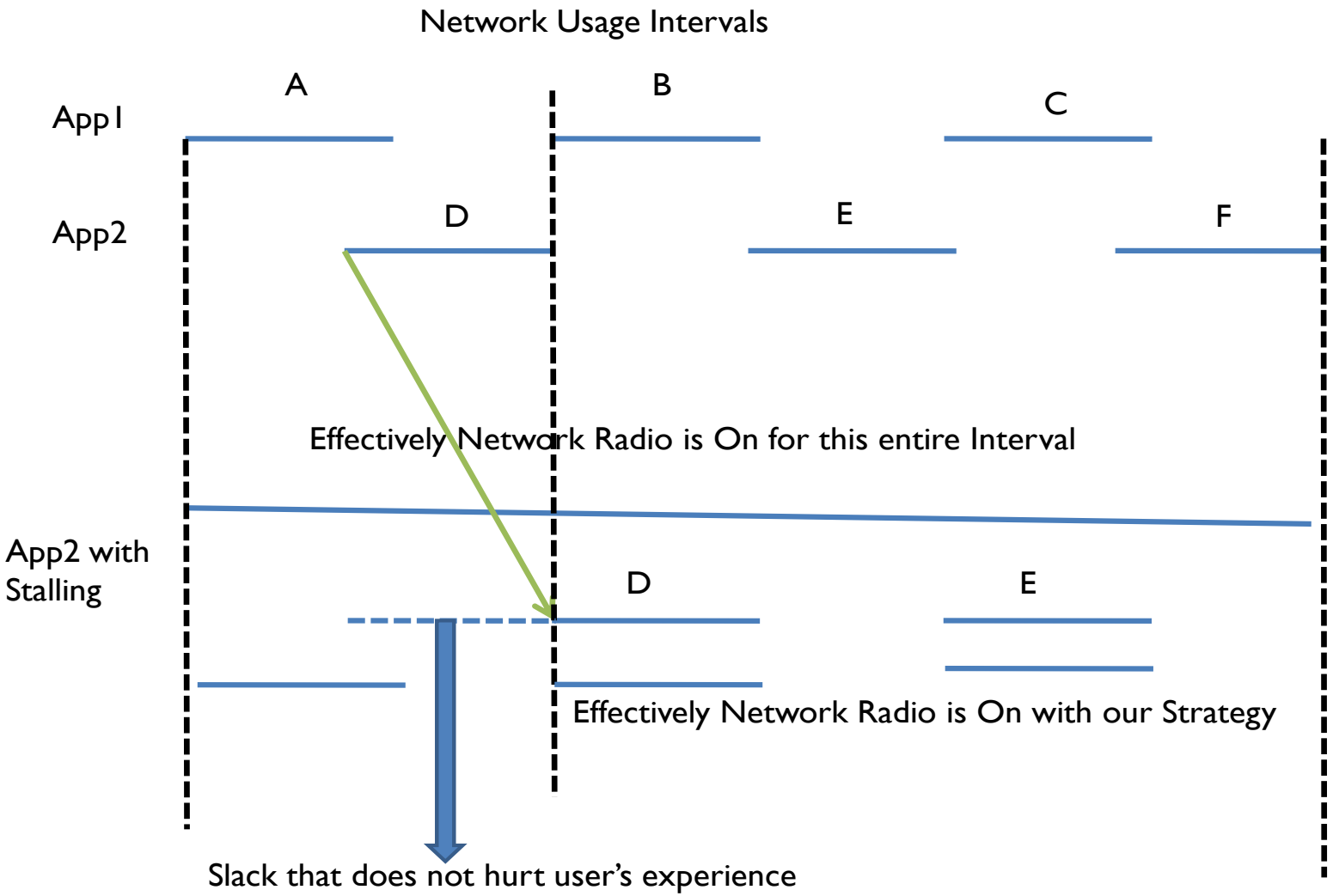
Objective

Scheduling all network requests using minimum energy without hurting user's experience.

Multi objective optimization problem

- **Minimum Energy** => **Best utilization of bandwidth** (Side-effect : **Lower Switching Frequency**).
- **User's Experience** => Request should be served **within deadline**.

Idea: Cross Application Aggregation



Approach Intuition

- If a request is delayed then there is potentially more opportunity of batching.
- If a request is delayed much, it may miss deadline.
- So, we need to develop a function to decide if a request should be scheduled or should be delayed.

Offline Algorithm

Input: Number of requests:n;
Requests[1...n];
Output: Scheduling of every request
Intialization: size=1;
while *size* $<$ *n* **do**
    start =1;
    end = n - size;
    while *start* $<$ *end* **do**
        Compute BS(*Request*[*start*],...,*Request*[*start* + *size* − 1]);
        if *violates bandwidth constraint* **then**
            Remove violation;
        end
        start++;
    end
    size++;
end

Dynamic programming approach to find the maximum possible overlap without violating the bandwidth constraint.

Online Algorithms

Lazy Scheduling

- **Running Queue(RQ)** has all the running requests served by Cellular Radio.
- **Wait Queue(WQ)** has all the pending requests.
- *Wait-as-long-as-you-can* Policy
- Push all possible requests to the **Running Queue**, when wait is no more possible.

Online Algorithms

Early Scheduling

- If **Running Queue** is empty, follow *Wait-as-long-as-you-can* policy.
- If **Running Queue** is not empty, push all incoming requests to the **Running Queue** (if Bandwidth is available).

Online Algorithms

Balanced Scheduling

if *Compatible Request found in Wait Queue* **then**
  Evaluate F function;
  if $F>0$ **then**
    Remove it from Wait Queue;
    Put it in Run Queue;
  end
end

$$F = \beta \cdot Bandwidth\_wastage + (1-\beta) \cdot Experience\_user$$

Algorithm 3: Balanced Scheduling Algorithm

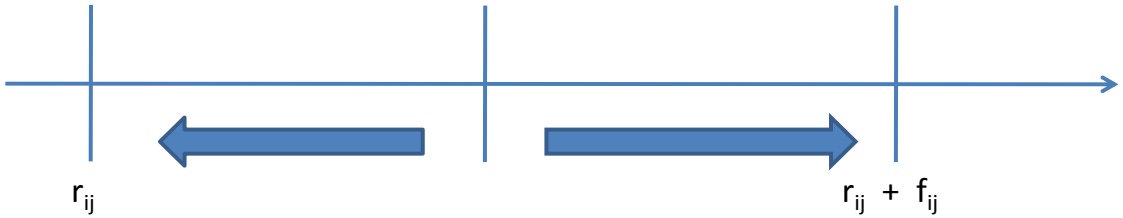
Balanced Scheduling

Terminology

- A_i= ith application
- A_{ij} = jth request of ith application
- r_{ij} = Arrival time of A_{ij}
- x_{ij} = Scheduling time of A_{ij}
- f_{ij} = slack time time of A_{ij}
- d_{ij} = service duration of A_{ij}
- ft = finish time of all requests in run queue

Deciding Function F

User Experience Component



Mid point

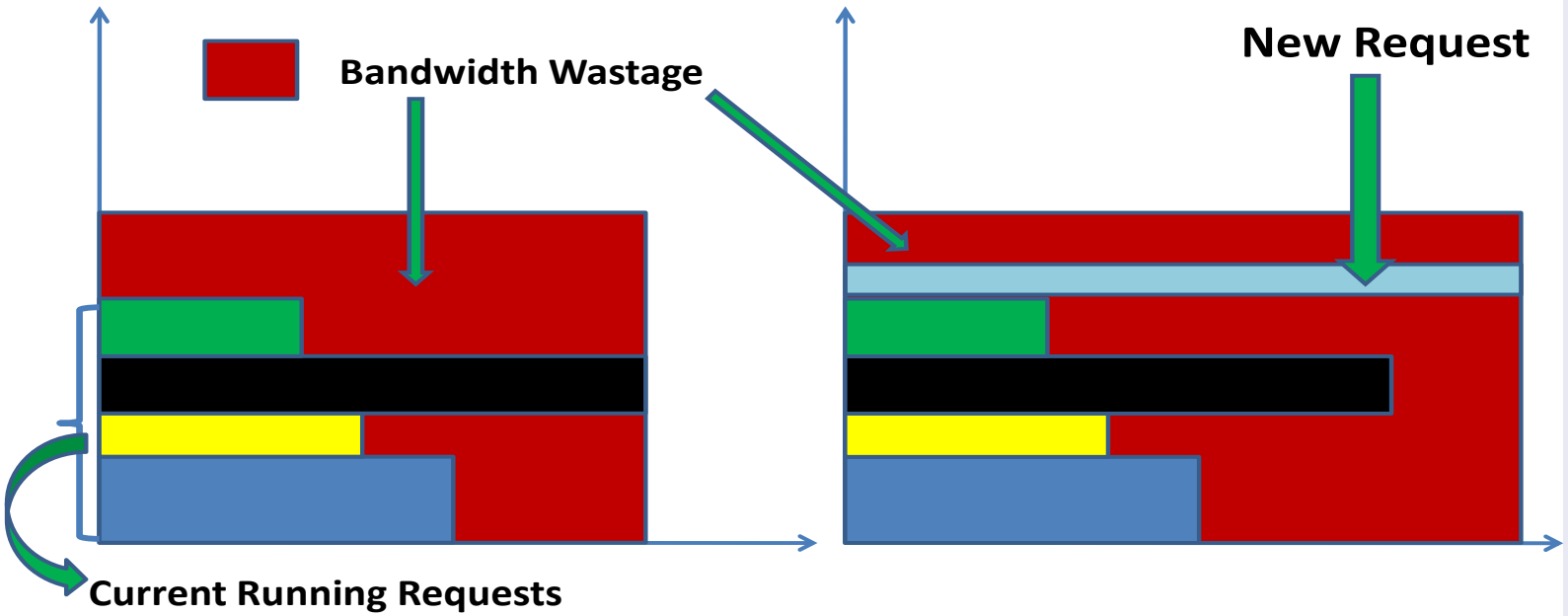
Do not schedule if **ft** lies here
Wait for more requests

Schedule if **ft** lies here

$$Experience\_user = \frac{ft - \frac{r_{ij}+(r_{ij}+f_{ij})}{2}}{Max(ft, \frac{r_{ij}+(r_{ij}+f_{ij})}{2})}$$

Deciding Function F

User Experience Component



$$Bandwidth\_wastage = \frac{BW_1 - BW_2}{Max(BW_1, BW_2)}$$

BW1 : Bandwidth Wastage Before Scheduling
BW2 : Bandwidth Wastage After Scheduling

Experimental Setup

Application Types (MobiSys '12)

Gaming (Short Bursts), Browsing (Medium Bursts), and Streaming (Large Bursts)

Synthetic Trace Generation tuning parameters

- Application Sync Timing (Fixed Intervals)
- User Interaction Timing (Power Law)
- Data Transmission Size (Power Law with set of sizes)
- Bandwidth Demand (Fixed Demand per App)
- Slack Duration (Fixed per network request type per App)

Switching Strategies

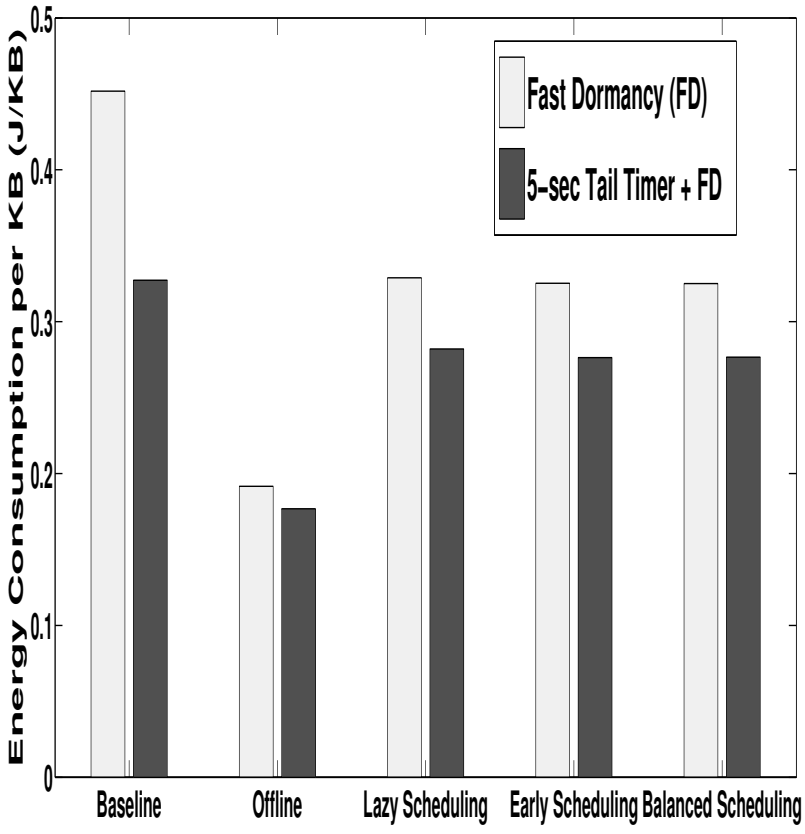
- Fast Dormancy
- Fast Dormancy with Tail Timer

Evaluation Setup

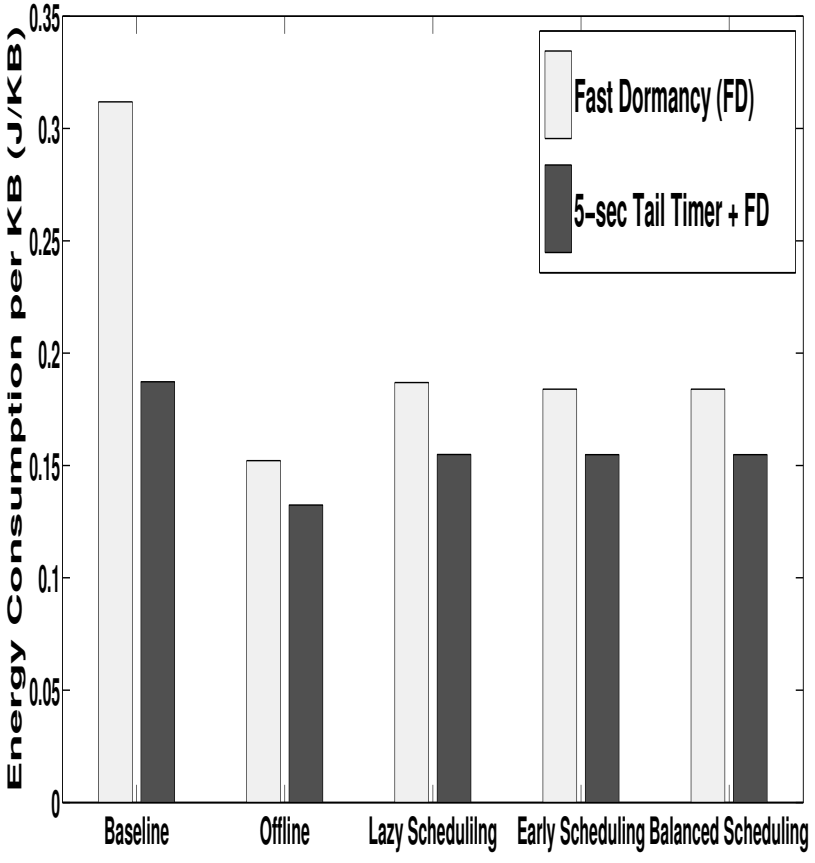
Evaluation Metrics

- **Energy Consumption per KB**: Total energy spent to transmit one KiloByte of data over the network interface.
- **Deadline Miss**: Proportion of requests which have missed their transmission deadline.
- **Switching Frequency**: Number of times per unit time the radio changes state (from IDLE to DCH, and DCH/FACH to IDLE).
- **Radio On Time**: Radio on time as a fraction of total data transmission duration.

Energy Gain



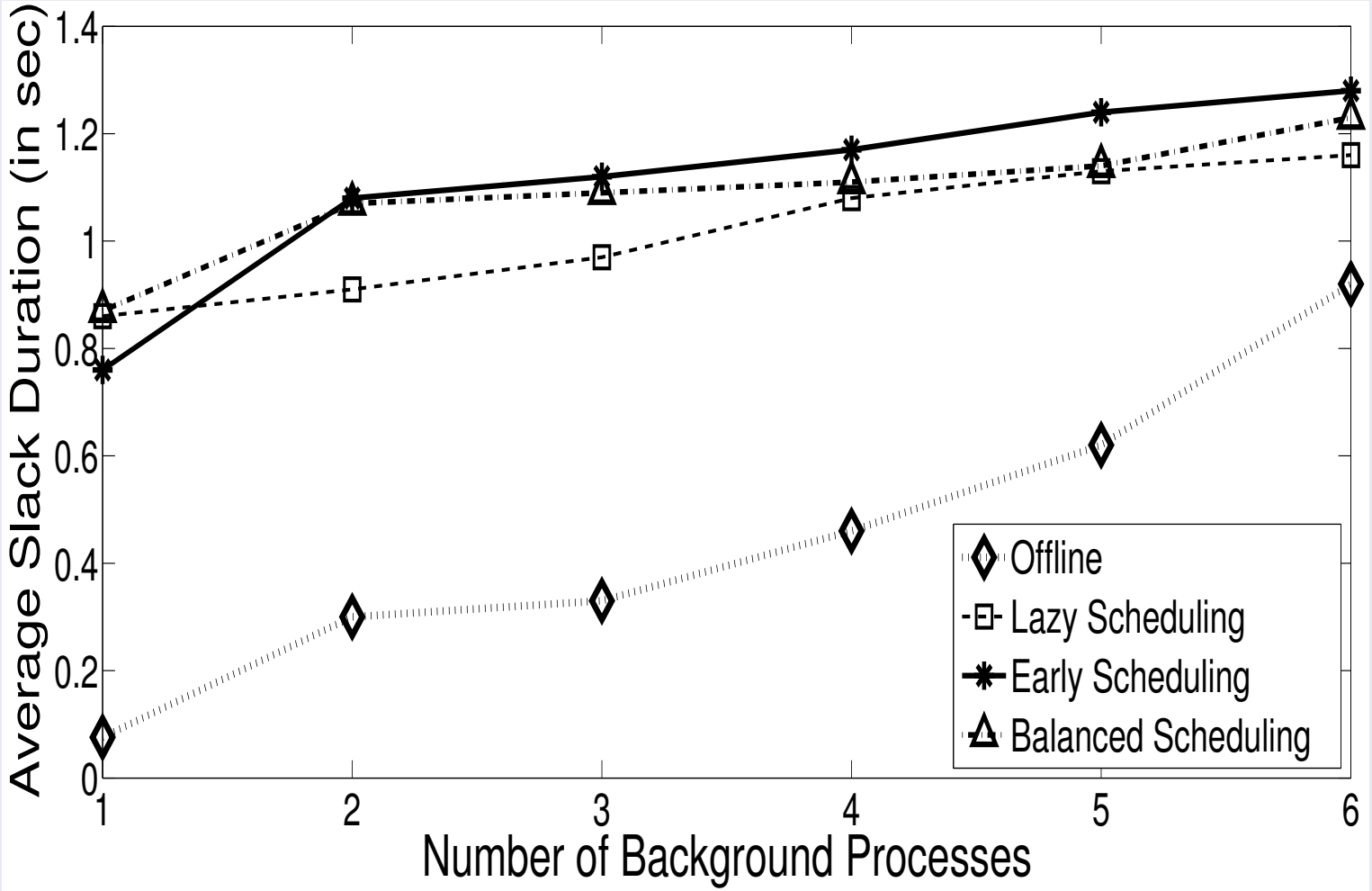
(i) Browsing Scenario



(j) Streaming Scenario

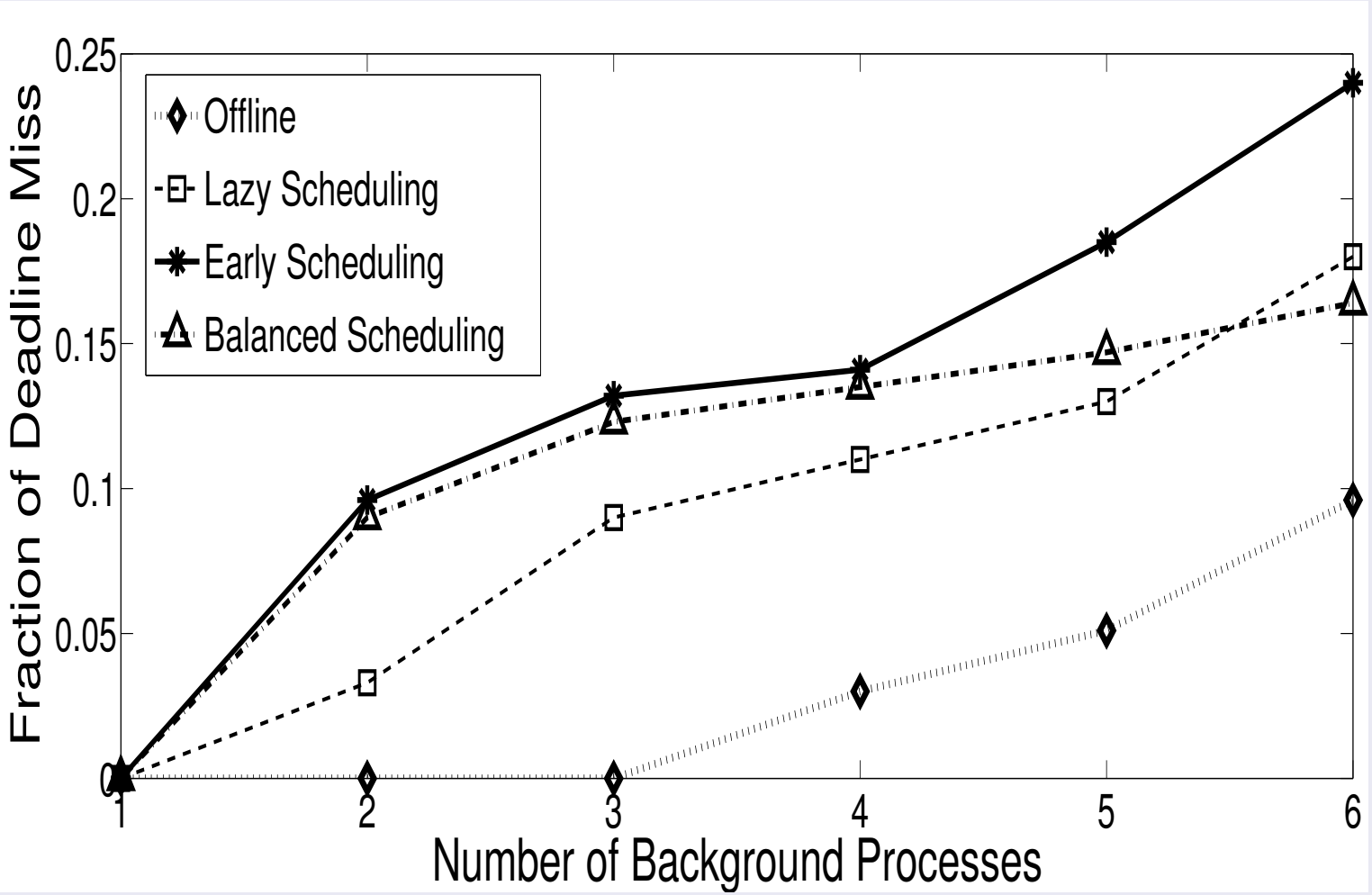
User Experience

Average slack incurred by foreground app

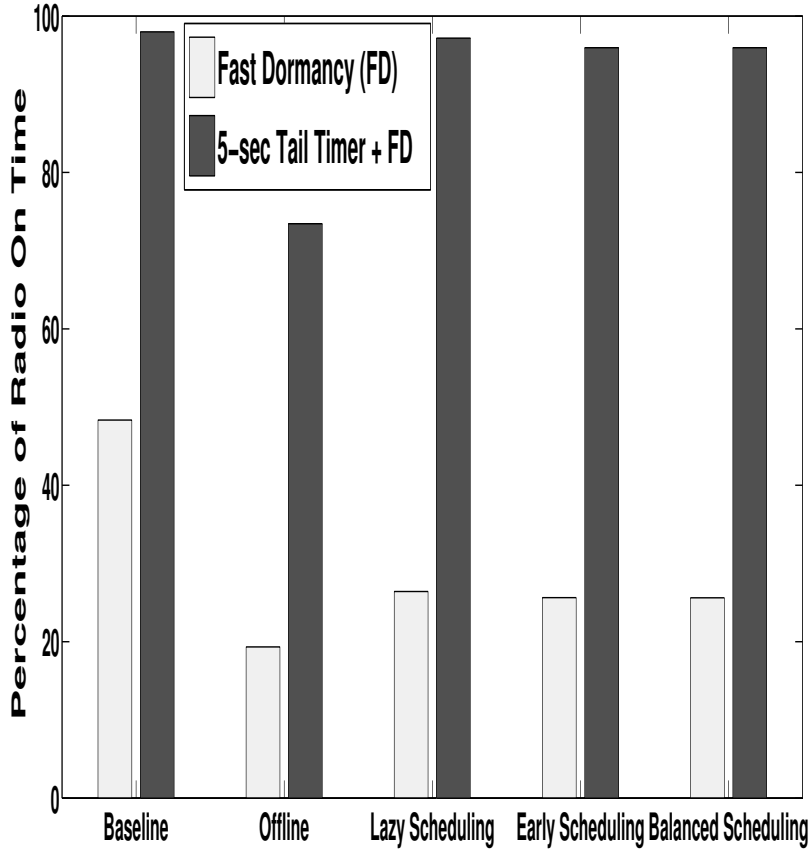


User Experience

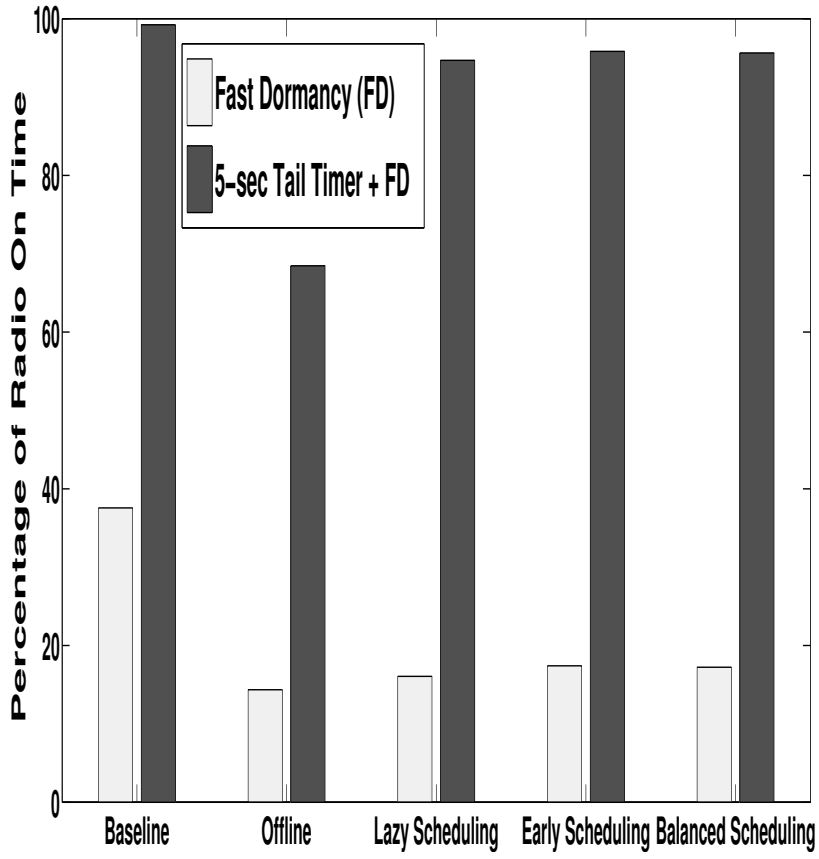
Fraction of overall packets from foreground app missing deadli



Percentage of duration when the radio was ON

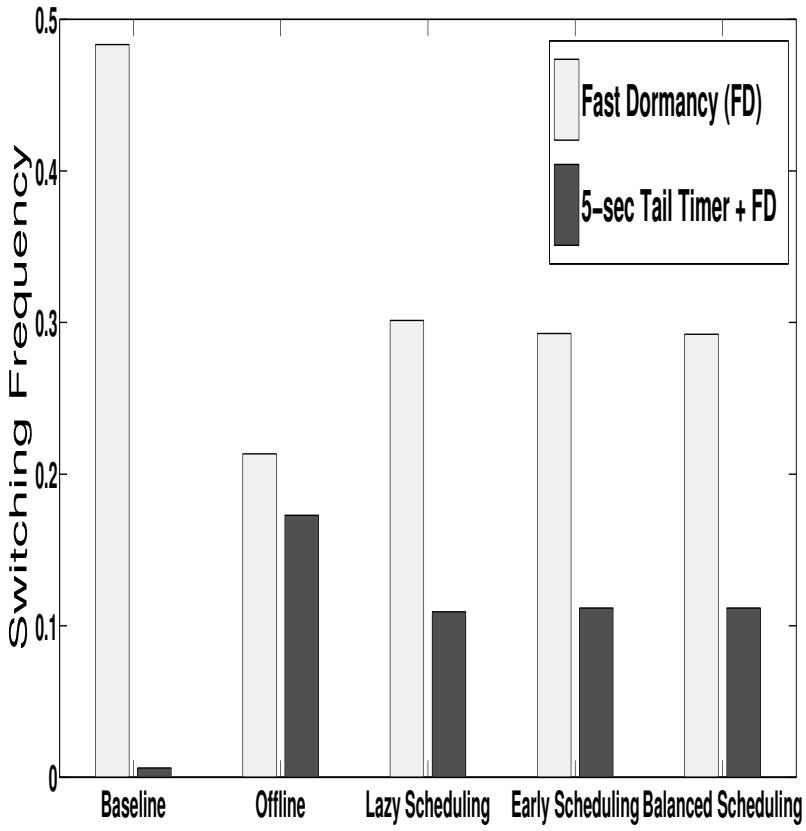


(k) Browsing Scenario

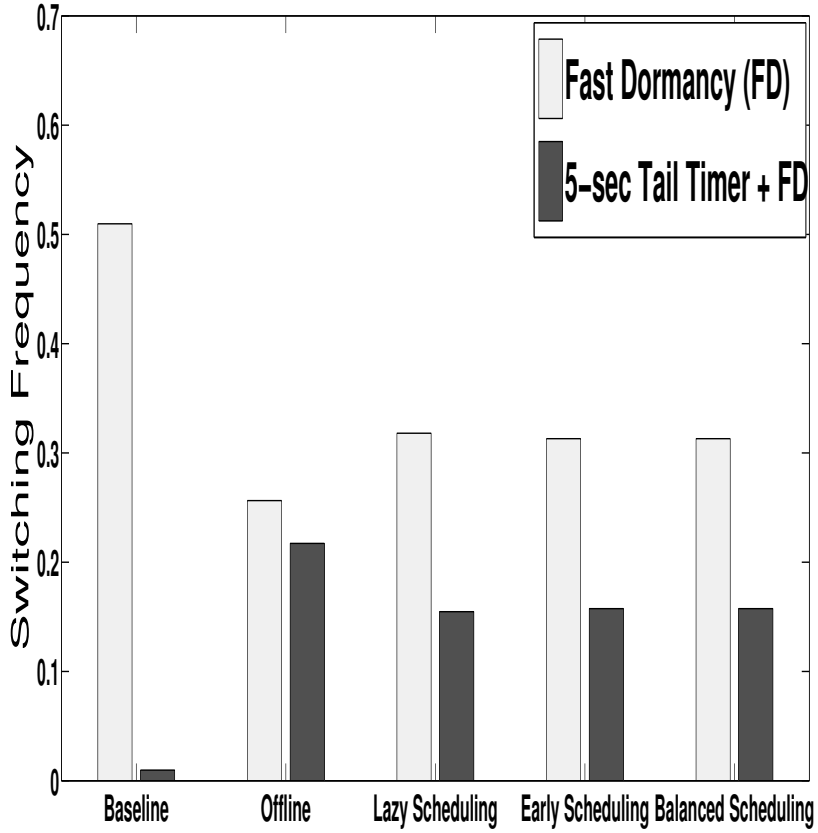


(l) Streaming scenario

Switching frequency comparison



(m) Browsing scenario



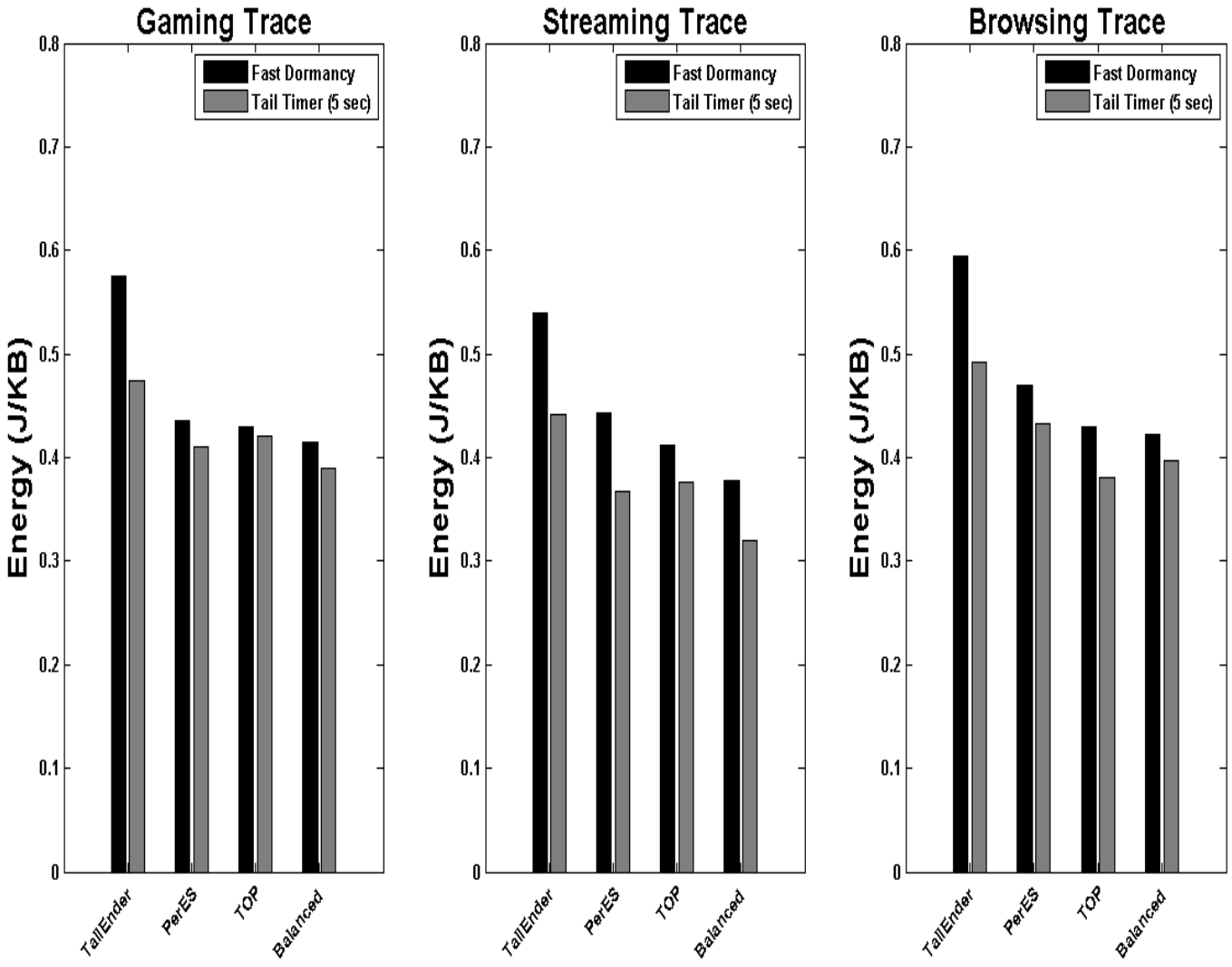
(n) Streaming scenario

Comparing with Different State-of-the-art Techniques

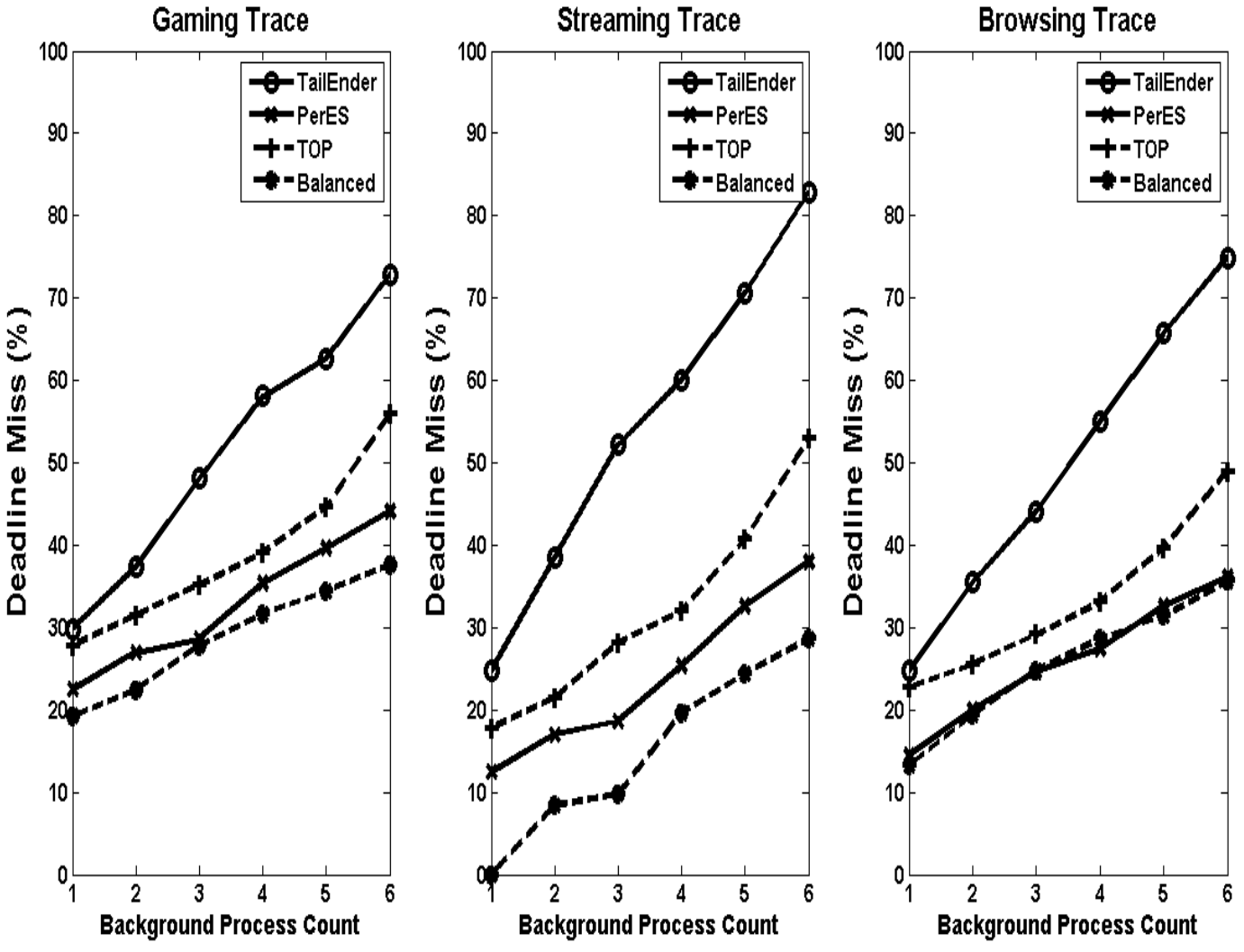
Competing Schemes

- **TailEnder**: TailEnder uses threshold based tail time prediction by considering deadlines of packets of an application.
- **PerES**: Models cross application energy-delay trade-off as an optimization problem and applies Lyapunov optimization framework.
- **TOP**: Tail Optimization Protocol (TOP) reduces tail energy wastage by predicting the application behavior.

Energy Gain Comparison



Deadline Miss Comparison



Result Summary from Synthetic Trace

- For Gaming Scenario and Streaming Scenario, Onine Algorithms give around 30% Energy Gain.
- Switching Frequency and Percentage of Radio On time is reduced by around 20%.
- Percentage of Deadline Misses for Foreground App remains satisfactory.
- Around 10% better than PerEs and TOP in Energy Gain wise, but far better from TailEnder.

Real Trace Collection and Challenges

Through VpnService

- PROS : No Rooting
- CONS: Multiple Apps not in parallel

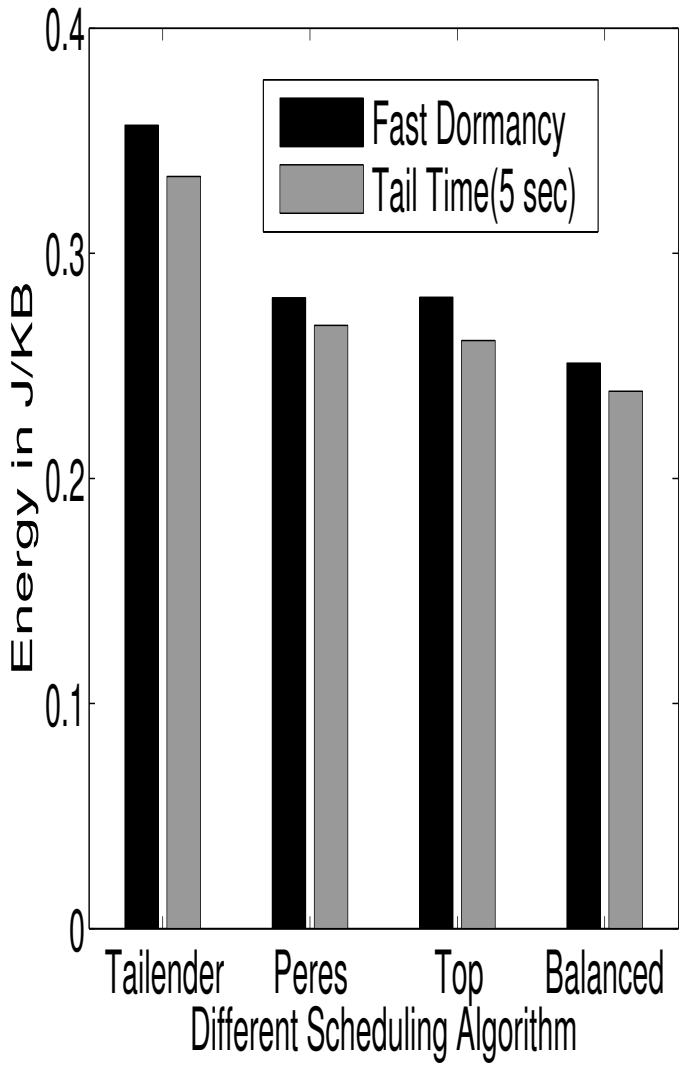
Through tcpdump, netstat, and ps

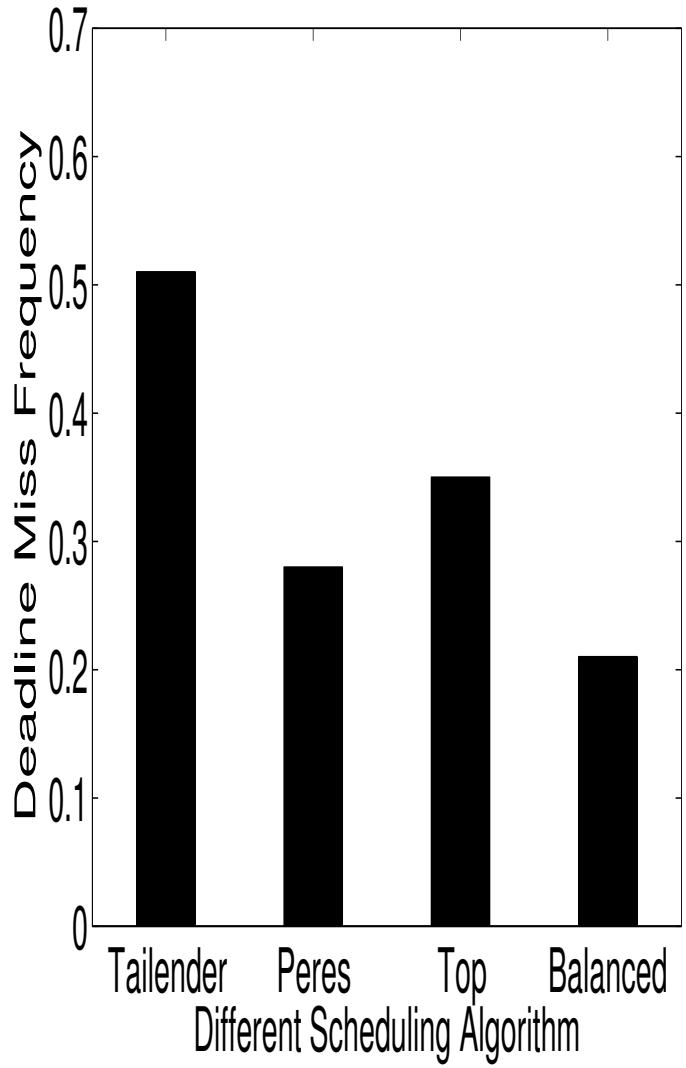
- PROS: Multiple Apps, More detailed
- CONS: Rooting

Through ARO tool and netstat and ps (We Used)

- PROS: Multiple Apps, More detailed
- CONS: Rooting

Real Trace based Results





Takeaways

- Reasonable localization accuracy using virtual landmarks which can help to build many applications like *RetailGuide* or *SignFinder*. Device heterogeneity is the main reason of instability of virtual landmarks.
- Balanced Scheduling across Applications yields more benefit and gives around 10 to 40% energy gain (due to better utilization of *CELL_DCH*). This aggregation strategy does not hurt much the user experience.

Queries from Prof. C.S. Kumar

Q1: Can the coupled use of multiple sensors with Markov model be adopted to increase the accuracy of localization?

We have used single or multiple sensors to create landmarks and then used landmarks for localization. However, we have not adopted coupled use of multiple sensors with markov decision model which is norm in SLAM kind of system in robotics. The main reasons are the following:

- For coupled use of multiple sensor for error correction, we need very accurate sensors which are available in robotics but real world smartphone sensors are very noisy .
- Robotic movements are very accurate. So, we can use one sensor to simultaneously correct other sensors using kalman filter like approach which is sadly not possible in smartphone scenario.

Queries from Prof. C.S. Kumar

Q2: Can these schedulers be used in online setting (e.g. in real multiple application running scenario) to reduce energy?

Yes, our scheduler can be used in online setting to handle multiple requests coming from different apps running in smartphone. Specially, balanced scheduling actually trades off between bandwidth wastage and user experience to handle different requests from different apps in online setting.

Queries from Prof. C.S. Kumar

Q3: Can a default energy model be used to compare the energy consumption?

We have used initially baseline scheme (FCFS) for comparing energy. Later, as per the suggestion, we have introduced different state-of-the-art online competing schemes like PerEs, TOP etc. to compare with our balanced scheduling algorithm. The new results are incorporated in the section 4.4 and section 4.5 of the thesis.

Queries from Prof. Vinayak Naik

Q1: 50% of how many data points to consider the stability of landmarks?

We have used 6 users with 4 trials each for 4 (time,device) pairs. So, we have got 6*4*4 = 96 data points for our experiments. If one landmark appeared in more than 48 data points, we consider it as stable. However, we initially tried with more trials with two users, but the experiments did not provide more stable landmarks.

Queries from Prof. Vinayak Naik

Q2: How often one should check wait queue in Early Scheduling?

Packets which are ready for transmission are taken out of wait queue and put into run queue. The readiness of the packet will be computed in wait queue by using early strategy. Scheduler at every time instant checks wait queue but only wakes up run queue whenever insertion/deletion happens based on the strategy. Therefore, the frequency of wait queue checking depends upon the trace.

Future Works

- Large scale deployment and different architectural tweaks (distributed landmark database) can be done, for *RetailGuide*.
- If we add this landmark based localization system to other infrastructure based indoor localization, how it will work?
- Building middleware which will run our aggregation strategy across applications.
- Extension and implementation of in other elements like sensors, GPS etc.
- Building a Application network activity recorder tool which can be installed without rooting.

Thanks to all Collaborators

- Prof. Niloy Ganguly (IIT KGP)
- Prof. Bivas Mitra (IIT KGP)
- Prof. Pradipta De (SUNY, Korea)
- Prof. Romit Roy Choudhury (UIUC, USA)
- Sourav Dandapat (PhD Student, IIT KGP)
- Ananth Balashankar (Dual Student, IIT KGP)

Publications from the Thesis

- **S. Pradhan**, A. Balashankar, S. Dandapat, B. Mitra, N. Ganguly, “(Stable) Virtual Landmarks: Enhancing Localization centric Smartphone Applications”, communicated to *IEEE Transactions on Mobile Computing*.
- **S. Pradhan**, S. Dandapat, B. Mitra, N. Ganguly, P. De, “Aggregating Inter-App Traffic to Optimize Cellular Radio Energy Consumption on Smartphones”, **COMSNETS**, 2015, (A poster version has been showcased in *XRCI* Open, 2014).
- **S. Pradhan**, A. Balashankar, B. Mitra, N. Ganguly, “(Stable) Virtual Landmarks : Spatial Dropbox to enhance Retail Experience”, **COMSNETS**, 2014.

Thank You

Any Questions