PRISM: Platform for Remote Sensing using Smartphones

Tathagata Das
Microsoft Research India
Bangalore 560080, India
tathadas@microsoft.com

Prashanth Mohan
University of California, Berkeley
Berkeley, CA 94720, USA
prmohan@cs.berkeley.edu

Venkata N. Padmanabhan
Microsoft Research India
Bangalore 560080, India
padmnab@microsoft.com

Ramachandran Ramjee
Microsoft Research India
Bangalore 560080, India
ramjee@microsoft.com
Asankhaya Sharma

Microsoft India Development Center
Hyderabad 500046, India
asankhs@microsoft.com

3/8/2011 – Steve Kopman

What is PRISM?

- PRISM – Platform for Remote Sensing using Smartphones
- Generic framework that balances generality, security and scalability
- Applications run within PRISM from executable binaries
- Applications are pushed to an appropriate set of users
- Applications run in a sandbox and utilize resource metering and forced amnesia

Community Sensing

- Current research focuses on "community sensing"
  - Include computing and communication capabilities as well as sensors (GPS, microphone, etc)
- Two types
  - Participatory
    - Require user actions (e.g. taking photograph)
  - Opportunistic
    - No user action required (e.g. GPS tracking)

Challenges

- Goal is to reduce application developers need to "reinvent the wheel"
- Three main goals
  - Generality
    - Support a wide range of applications with flexibility to reuse existing code
  - Security
    - Ensure that phones remain secure and that applications do not misuse sensitive sensor data
  - Scalability
    - Allow the system to scale to large (>100,000) number of devices

Sandbox Features

- In addition to standard SW sandboxing, three PRISM specific features are utilized to provide security
- Resource Metering
  - Limits the amount of battery energy an application can consume
  - Limits the "leakage" of sensitive sensor data
- Forced Amnesia
  - Does not allow sensing applications to maintain long-term state info
- Sensor Taint Tracking and Access Control
  - Allows the user to set policies on what applications can do

Implementation Overview

- Currently runs on Windows Mobile
  - Interesting choice since this is an obsolete platform
- Infrastructure components run on Windows 7
- Three implemented applications “showcase” the generality of the PRISM Platform
  - Citizen Journalist
    - Participatory, alerts users based on GPS location when to take pictures
  - Party Thermometer
    - Allows users to query other users to determine how “hot” the party is
    - Senses music to target users that are in a party
  - Road Bump Monitor
    - Opportunistic sensing to locate and detect road bumps

Related Work

System	Generality	Security	Scalability	Privacy
Bubble-Sensing	No	**Yes**	**Yes**	**Yes**
AnonySense	OK	**Yes**	No	**Yes**
Micro-Blog	No	**Yes**	**Yes**	No
PRISM	**Yes**	**Yes**	**Yes**	Ok

AnonySense vs. PRISM

	AnonySense	PRISM
Application Language	Constrained, AnonyTL	Runs Generic Binaries
Privacy	Uses “pull” approach ; does not reveal nodes position in infrastructure	Uses “push” approach; Allows limited tracking of nodes
Sandbox applications	No	Yes

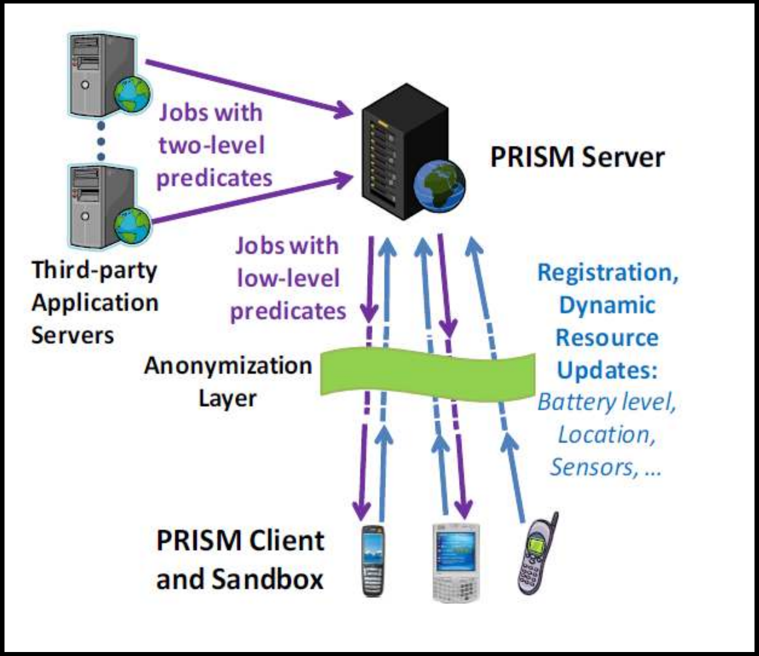
PRISM DESIGN

Assumptions

- Users trust the PRISM application and install it on their mobile devices
- Entities that submit applications have identities certified by a trusted authority
- Participating nodes, the OS on the phone and standard OS mechanisms a re trusted
- Nodes have Wireless Wide Area Network (WWAN) access

Architecture

- Application server (supplied by third parties)
  - Submits jobs to PRISM servers, for deployment onto a desired set of mobile phones
- PRISM server
  - accepts jobs from the application servers and deploys them onto an appropriate set of mobile phones
- PRISM client and sandbox on mobile
  - registers with PRISM servers and supports the execution of the jobs in a specially designed sandbox


Jobs with two-level predicates
PRISM Server
Third-party Application Servers
Jobs with low-level predicates
Registration, Dynamic Resource Updates: Battery level, Location, Sensors, ...
Anonymization Layer
PRISM Client and Sandbox

Push-based Model

- Push-based systems do not require the user or phone to retrieve data from the server
- The server sends data to the phone and provides the following benefits
  - Fast response by tracking phone resources and sending applications immediately when the phone is available
  - Efficiency by eliminating the need for each application to track a phone's resources
  - Scalability – Amount of tracking can be modulate to the load of application arrivals and the density of available phones

Registration

- Used to track a phone's resources
  - Resource loading is maintained as soft-state and expires after the registration period
    - Authors used one hour to balance privacy and overhead
  - Tracks both static, such as sensors and radios, and dynamic, such as battery and location, resource information
- PRISM, unlike AnonySense, uses a push method that requires tracking of users
  - Tracks users during the registration period only
  - Re-registration occurs after phones wait for a random time
  - Employs independent anonymization service to protect against tracking between registrations

API

- Designed to allow the server to quickly and accurately identify phones to run the application
- Identification uses a two-level predicate mechanism
  - Top-level is coarse grained and identifies phones where jobs are deployed but not activated
  - Low-level is fine-grained and determines when to activate applications
  - Implemented to reduce the phone’s fine-grain updates to the server, to ensure that the sensing opportunity is not missed and to reduce the risk of spam.

Top-Level Predicate

- Specifies the number of phones needed, the capabilities of the phones and their coarse grained location
- Server can either supply an application for each hardware/OS platform or use the hardware/OS platform as part of the search criteria

Low-level Predicate

- Can consist of locations or be based on derived attributes
  - Example is speed
- Includes a time-out parameter which determines how long the client monitors for a match to the fine-grain predicate

Deployment Modes

- Two Types
  - Deploy-or-cancel
    - Deploys the application as soon as a top-level predicate is matched
    - Good when a “large” area is specified for the top-level predicate
  - Trigger
    - Application Server sets a trigger with the PRISM server for the desired predicate
    - Good for low-density regions

Phone to Server Updates

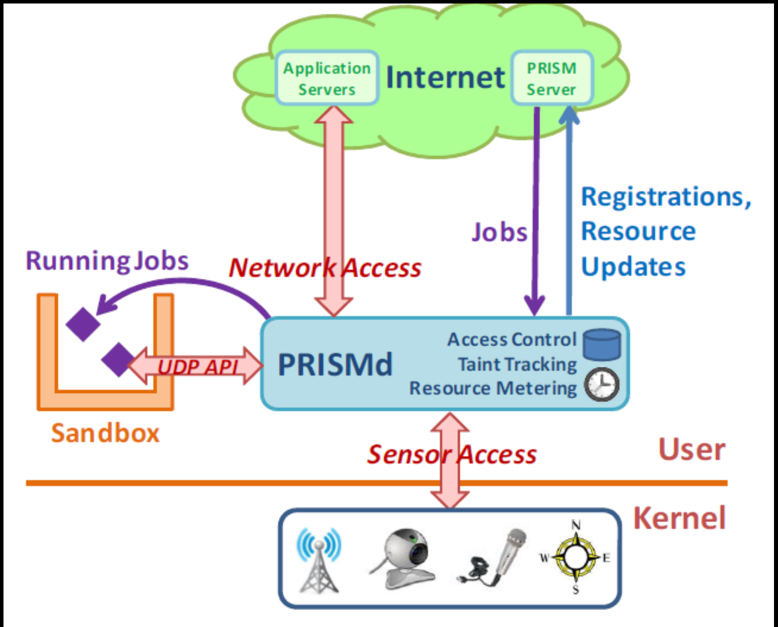
- Mobile client update messages to the server are overhead and need to be reduced
- Two techniques proposed
  - Adaptive Updates
    - $p = \min(1, \rho * n/N)$
    - Each client is notified with a parameter p at registration
    - ρ = job arrival rate, n = avg. # of phones requested by a job, N = total # of registered phones
    - The client sends updates with a probability of p that can be adjusted if there is a large # of phones (N) or if there is little application demand ($\rho * n$)

Phone to Server Updates

- Prediction-based Suppression
  - Mobile and server run identical predictors for each resource
  - Mobile only sends updates when the dynamic sensor status has changed significantly from the predictor
  - Two Types
    - Constant Predictor
      - The predictor predicts the new value is the same or “close”
    - Affine Predictor
      - Predicts the new value as an affine function of a quantity (e.g. time) that is shared by the server and client
      - Good for battery energy because exact tracking is not needed

Software Sandbox

- Provides the application binary a standard API to communicate with the PRISMd daemon.
  - Daemon controls access to sensors
- Additional security features including Sensor Access Control, Resource Metering and Forced Amnesia are also used to mitigate privacy risks



Sensor Access Control

- Three policies
  - No sensors
    - Application does not have access to the sensors however the PRISM runtime does have access to the location information
    - Useful with “human” sensor applications
  - Location Only
  - All Sensors
- Sensor Taint Tracking
  - Alternative to coarse-grained policies
  - Diminishes the ability of an application to process or transmit sensitive data
  - Example – An app that uses the microphone is tainted with microphone data that is sensitive

Resource Metering

- Applications should not drain the battery
  - User is a participant in community sensing project and does not want detrimental affects on their device
- PRISMd mediates access to sensors, tracks resource usage and limits access by not sending up sensor data
- CPU and memory utilization are monitored
  - Applications that exceed their allocation are terminated
- Energy Metering
  - Accomplished by using a simple linear function of the amount of time a resource is busy and the number of data reads/writes
  - Measured actively to ensure resources are not overused
- Bandwidth Metering
  - Limited for privacy and cost (tariffs for data)

Forced Amnesia

- Bandwidth Metering limits the amount of traffic an application can use
  - Increases privacy by not allowing large amounts of sensitive data to be exported
- What if application buffers the data and sends it out over a period of time?
  - Forced Amnesia clears the state of an application after a fixed period of time (i.e. 1 minute)
  - Most applications are not performing long computations so there is no ill effect

IMPLEMENTATION

Computing Resources

- 15 Smartphones running Windows Mobile 5.0 or 6.1
  - NOTE: Windows Mobile is a significantly obsolete OS, but research was done by Microsoft
  - All of the phones had GPS, camera, microphone, 802.11b, Bluetooth, and GPRS/EDGE/3G Radios
  - Three phones had external accelerometer sensors attached
- Infrastructure components run on Win7

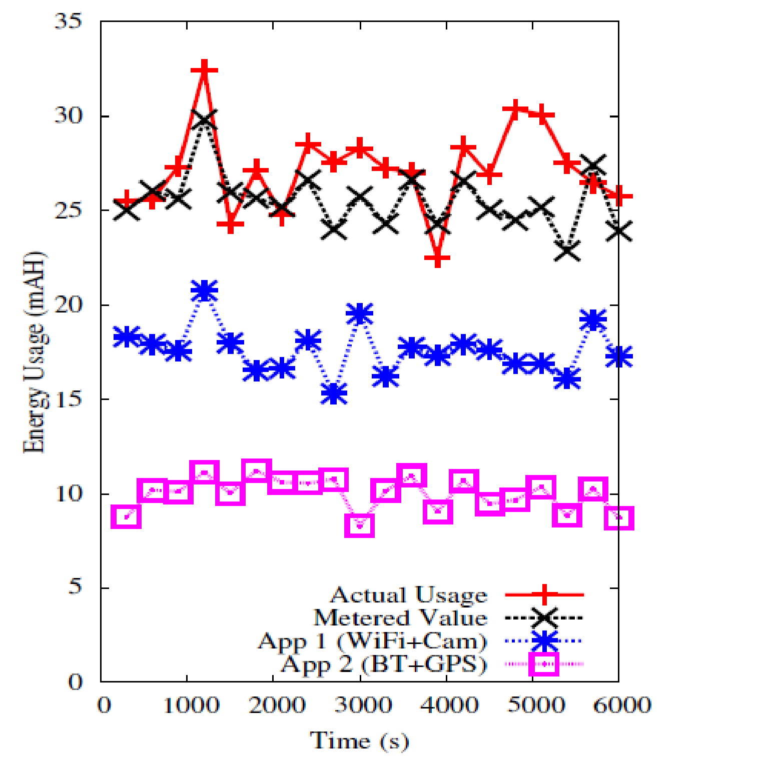
PRISM Infrastructural and Mobile Phone Components

- Infrastructure
  - Prototyped two-level predicate-based API and deploy-or-cancel and trigger modes
- Mobile Phone
  - Comprises of software sandbox
    - Includes the PRISMd daemon and the system call interposition layer (shim)
  - System call interposition is applied to block network communications except to PRISMd, device access (ioctl), registry access, spawning of child processes and file system calls that return a handle

EVALUATIONS

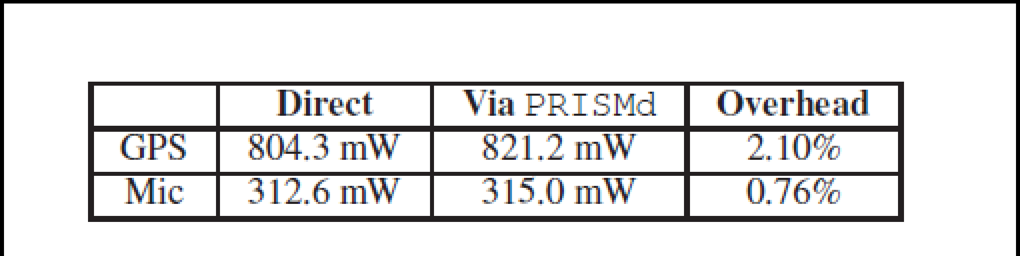
Energy Metering

- Emulated two applications
  - One cycles through using the camera sensor, performing Wi-Fi scans
  - The second uses the GPS and performs Bluetooth scans
- Linear model tracks actual usage but undershoots on the applications due to system related power that isn't metered by PRISMd



PRISMd Overhead

- PRISMd mediates access to the mobile’s sensors and needs to have a minimal impact on the system resources
- Used GPS and Microphone sensors to estimate overhead



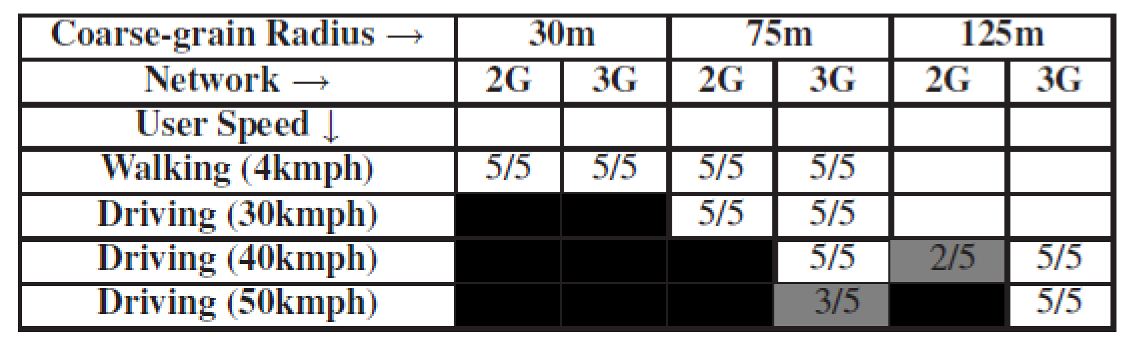
	Direct	Via PRISMd	Overhead
GPS	804.3 mW	821.2 mW	2.10%
Mic	312.6 mW	315.0 mW	0.76%

Applications (Recap)

- Three applications were implemented
  - Citizen Journalist
    - Participatory, alerts users based on GPS location when to take pictures
  - Party Thermometer
    - Allows users to query other users to determine how “hot” the party is
    - Senses music to target users that are in a party
  - Road Bump Monitor
    - Opportunistic sensing to locate and detect road bumps

Citizen Journalist

- Application sends alert to human users to take a picture or answer a query when they enter a specified location
  - Both high and low latency queries are implemented
- Location is specified by latitude/longitude with a coarse-grained radius for deployment and fine-grained radius for execution
- Benchmarking (35kB executable)
  - Fine-Grained Radius of 30m
  - Black = no success, Grey = partial success



Coarse-grain Radius →	30m		75m		125m	
Network →	2G	3G	2G	3G	2G	3G
User Speed ↓						
Walking (4kmph)	5/5	5/5	5/5	5/5		
Driving (30kmph)	■	■	5/5	5/5		
Driving (40kmph)	■	■	■	5/5	2/5	5/5
Driving (50kmph)	■	■	■	3/5	■	5/5

Citizen Journalist

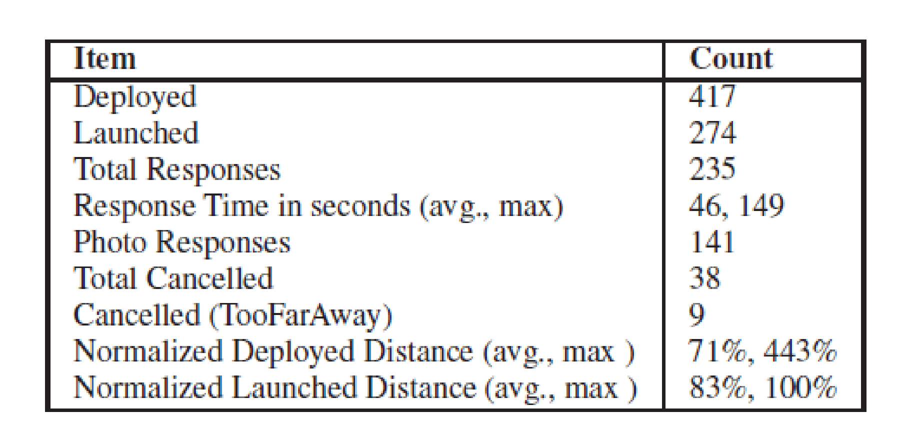
- Notes on benchmarking
  - 2G networks with a 30m coarse-grained radius often launched the app past the center point of interest
    - Larger coarse-grained radius of 75m is needed for pedestrians on 2G networks
  - 3G networks yield higher success rates due to lower latency and higher bandwidth
  - Coarse-grained radius needs to increase with user speed

Citizen Journalist

- Small-scale Pilot Deployment
  - Ten users, including three of the authors
  - Used 2G phones with GPRS
  - Total of 30 target locations within the vicinity of the Microsoft Research India lab in Bangalore
  - Fine-grained radius of 30m, Coarse-grained of 75m (speed limit was < 30kmph)
  - Application could be cancelled by user either by ignoring the phone ringing or manually cancelling it

Citizen Journalist

- Results
  - Response time (including deployment) averaged 46s
  - Normalized deployment distance average (relative to coarse-grained radii) was 71%
    - Server does not have precise GPS info
  - Normalized launch distance average (relative to fine-grained radii) was 83%
    - Mobile knows precise GPS location



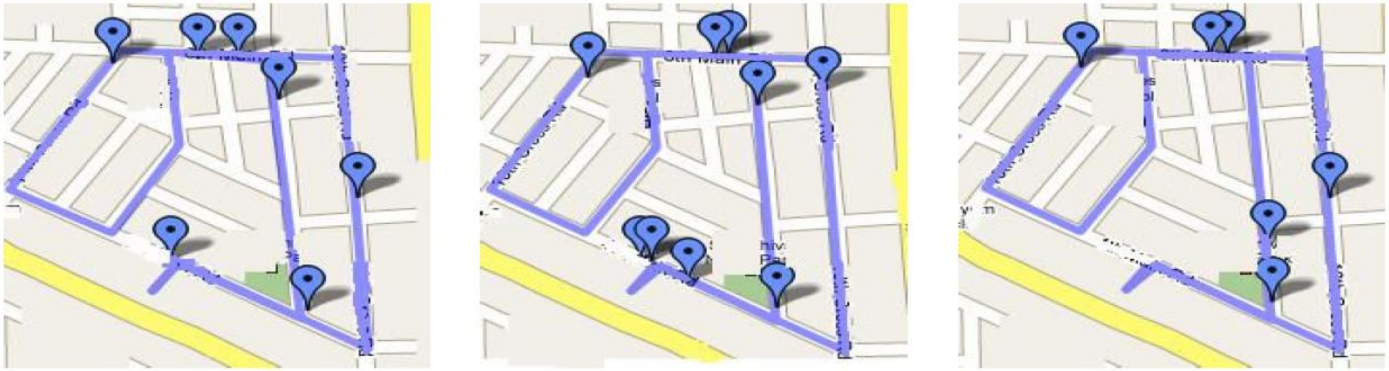
Item	Count
Deployed	417
Launched	274
Total Responses	235
Response Time in seconds (avg., max)	46, 149
Photo Responses	141
Total Cancelled	38
Cancelled (TooFarAway)	9
Normalized Deployed Distance (avg., max)	71%, 443%
Normalized Launched Distance (avg., max)	83%, 100%

Party Thermometer

- Human-query application to determine if the party is “hot”
- Detects music using microphone application
  - User must be stationary
  - Top-level predicate is a building to limit battery usage
  - Uses a FFT of the audio samples to examine spikes in the frequency domain for harmonics
    - This is the second-level predicate
- Limited testing
  - Verified that is was only deployed to users in the target location and that the music detection worked

Road Bump Monitoring

- Opportunistic sensing application
- Uses GPS and accelerometer to detects "bumps" in road
- Used a 2.5km long drive through a neighborhood
- 9 bumps, 6 correct within 12m of ground truth

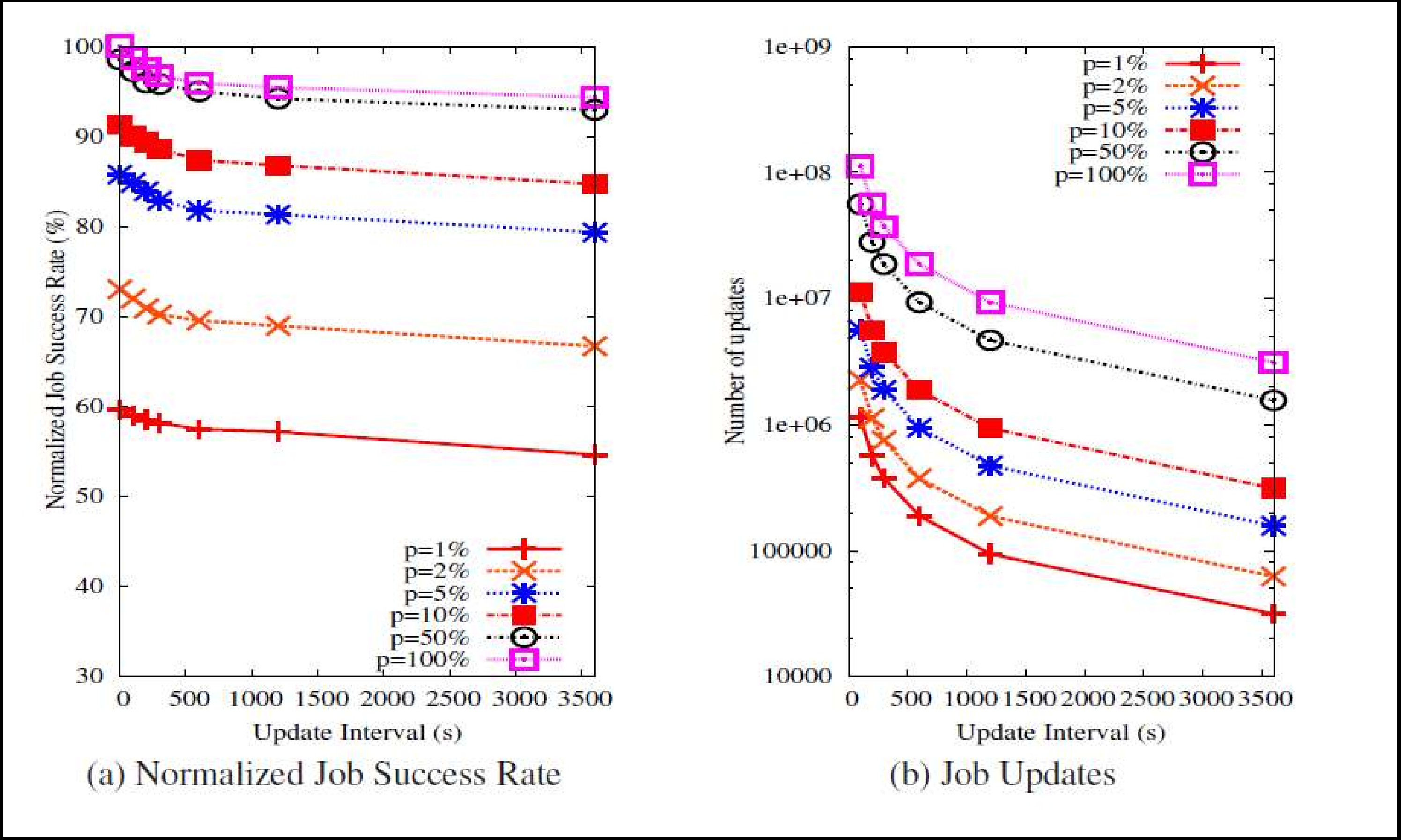


(a) Ground truth (b) Bumps detected without forced amnesia (c) Bumps detected with forced amnesia

Scalability

- Simulated a larger scale deployment
- Two key metrics to balance efficiency in resource updates and balancing the needs of applications
  - Total number of resource updates
  - Normalized job success rate
- As expected, smaller update interval the greater the success rate
  - Update interval of 100s yields a success rate within 2% of optimal

Scalability



(a) Normalized Job Success Rate

(b) Job Updates

Conclusion

- Presented a platform for participatory and opportunistic sensing
  - Uses “push” model
- Focused on scalability, security and resource utilization
- Utilizes a sandbox to protect user privacy