Enhancing Reliability of Community Internet of Things Deployments with Mobility

Qiuxi Zhu, Nalini Venkatasubramanian (Advisor), Department of Computer Science, University of California, Irvine

"Extending smart city and community systems to leverage the advantages of mobile computing"

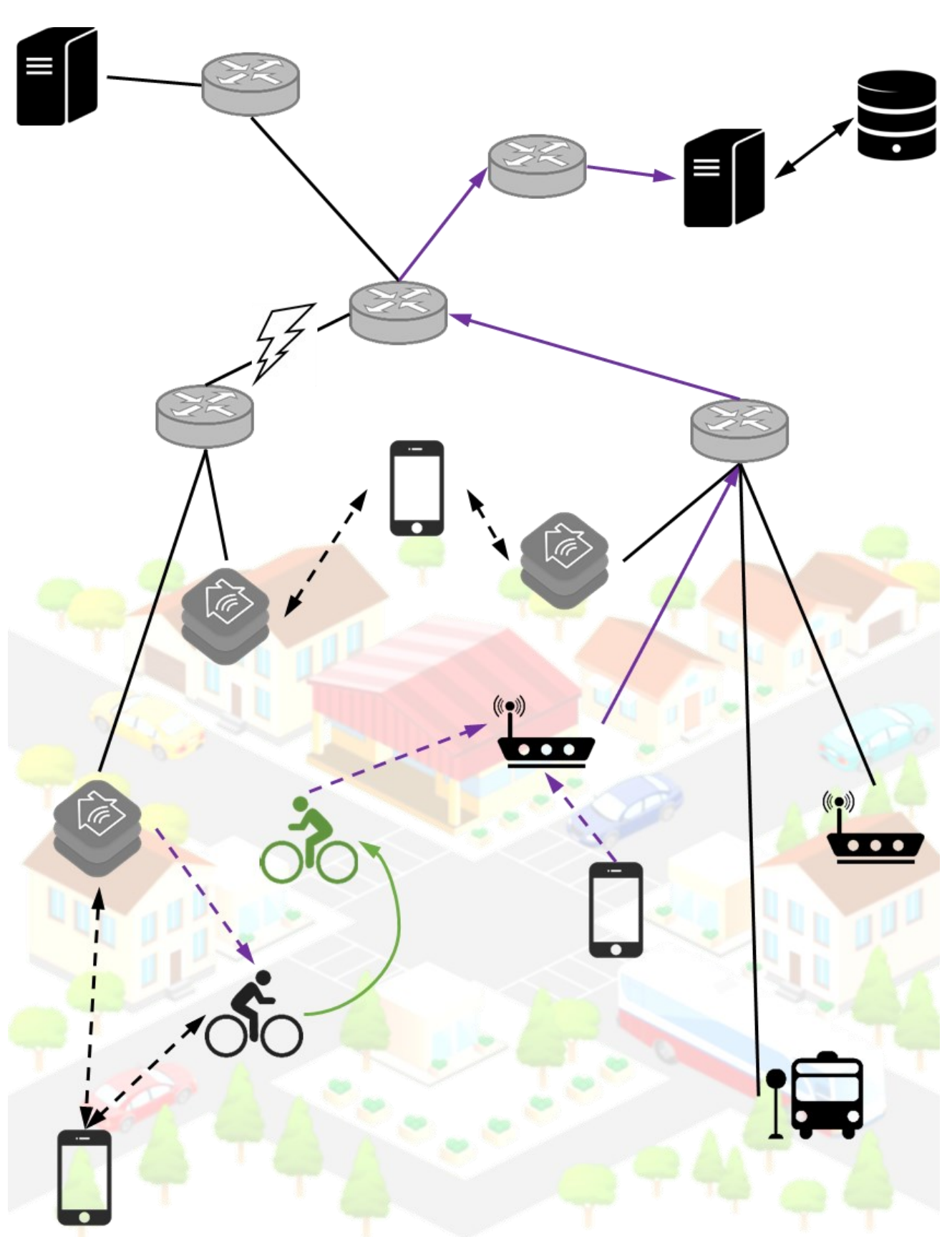
Goals and Overview

Communities and cities are interested in Internet of Things (IoT) systems. However, the current large-scale IoT systems have the following limitations.

- IoT systems depend heavily on public infrastructures
- Maintenance of devices costs a lot of money and labor
- Terrain limitations make it hard to deploy in some areas

A flexible approach to address these issues is demanded!

Mobile devices are popular, with various sensors and network capabilities. It is promising to leverage these capabilities of the mobile devices in our communities to extend the smart community systems.



Challenges

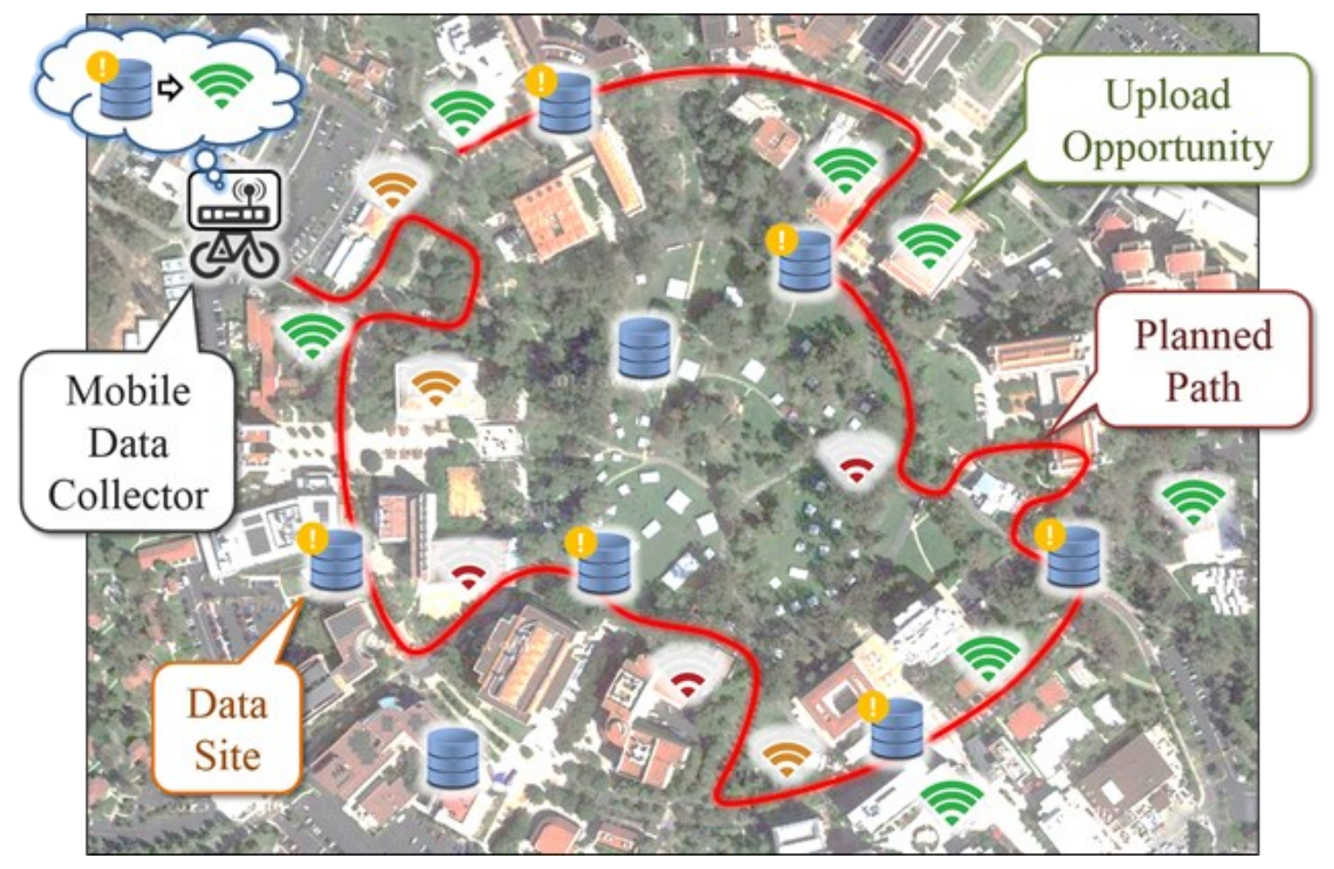
1.Scalability of system architecture

2.Dynamics in network availability and environment

3.Interoperability of heterogeneous devices

Upload Planning for Mobile Data Collection

A mobile data collector (MDC) is given a path, where there are several sites to fetch data, and several access points to upload data.

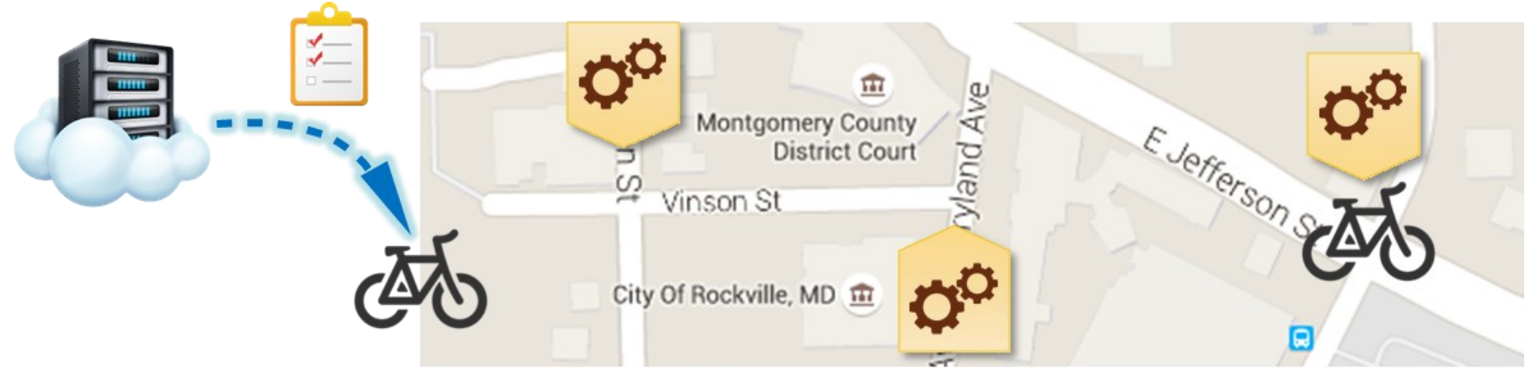


Plan for each *data chunk* fetched from data sites, which *upload opportunity* to use to upload it, in order to improve the overall timeliness.

Challenges: 1. Non-uniform network connectivity,

2. Data heterogeneity (e.g. size, importance, timing),

3. Environmental dynamics



We proposed a two-phase approach to address these issue, which consists of a static planning phase (on server) and a dynamic adaptation phase (on MDC).

Given an ordered list of data chunks $\{\boldsymbol{a}_i\}, i = 1, \dots, N$, with increasing $x(\boldsymbol{a}_i)$, and an ordered list of opportunities $\{\boldsymbol{u}_j\}, j = 1, \dots, M$, with increasing $x(\boldsymbol{u}_j)$.

Find global plan λ and its corresponding plan matrix $\boldsymbol{\Lambda}$, to maximize the WOU subject to the cause-and-effect constraint, i.e.

$$\text{maximize}\ U(\lambda, l) = \sum_{i=1}^{N} p(\boldsymbol{a}_i) \cdot f\big(\Delta(\boldsymbol{a}_i, \lambda, l)\big) \Big/ \sum_{i=1}^{N} p(\boldsymbol{a}_i),$$

$$\text{s.t.}\ \boldsymbol{\Lambda}_{i,j} \le \mathbf{C}_{i,j}, \forall i = 1, \dots, N, \forall j = 1, \dots, M.$$

We formulated a simplified version of the upload planning problem as a constrained optimization problem (shown on the left).

The upload planning problem is proven NP-hard: A 0-1 knapsack problem can be reduced to an upload planning problem.

Static planning using Balanced DOP

Input: Lists of data chunks $\{\boldsymbol{a}\}$ and opportunities $\{\boldsymbol{u}\}$

Output: Plan $\mathbf{j}$ that specifies which $\boldsymbol{u}$ to use for each $\boldsymbol{a}$

```
Initialize empty $\mathbf{j}$ and waiting set $W=\{\boldsymbol{a}\}$
Take data chunk $\boldsymbol{a}$ with the earliest deadline from W
If $\boldsymbol{a}$ can be planned on time without affecting any previously scheduled data chunks:
  Put it at the fastest possible opportunity
Else:
  While $\boldsymbol{a}$ is not planned yet:
    Select a previously scheduled low-priority chunk
    Mark it as sacrificed
    If total utility lost of sacrificed data is lower than utility gain of $\boldsymbol{a}$:
      If a can be planned on time:
        Put back all sacrificed data chunks
        Put $\boldsymbol{a}$ at the fastest possible opportunity
    Else:
      Clear the sacrificed marks and plan $\boldsymbol{a}$ for a late delivery
```

Dynamic adaptation using Lyapunov control

State of our system $\phi(t)$ at time slot t is a vector of two queues

$$\phi(t) = [Q(t), T(t)]$$

Quadratic Lyapunov function:

$$L\big(\phi(t)\big) = {}^{1}\!/_{2}\, Q^2(t) + {}^{1}\!/_{2}\, (T(t) - \beta)^2$$

$Q(t)$: Queue backlog at the MDC

$T(t)$: Measures the amount of time elapsed since the MDC started its operation

β: Time that is supposed to have elapsed according to the static plan

Lyapunov drift: $\Delta(t) = \mathbf{E}\big[L\big(\phi(t+1)\big) - L\big(\phi(t)\big) \big| \phi(t)\big]$

Minimize $\Delta(t) - V \cdot R(t)$, where $R(t)$ is the reward of all selected data chunks, which is equivalent to maximizing

$$\sum_{\boldsymbol{a}} \sigma(\boldsymbol{a}) \cdot s(\boldsymbol{a})(Q(t) - (T(t) - \beta)/r(t)) + V \cdot p(\boldsymbol{a}) \cdot f\big(T(t) + s(\boldsymbol{a})/r(t) - d(\boldsymbol{a})\big)$$

Static planning and dynamic adaptation algorithms we proposed for the two-phase approach

Scenarios and Testbeds

Victory Court Senior Apartments is a senior people's home in Montgomery County, MD. We have deployed our SCALE system to monitor the indoor environment. Now we are looking at extending this system for air quality.

- Only Internet infrastructure is the Wi-Fi AP in the building
- Get data from outdoors dev.

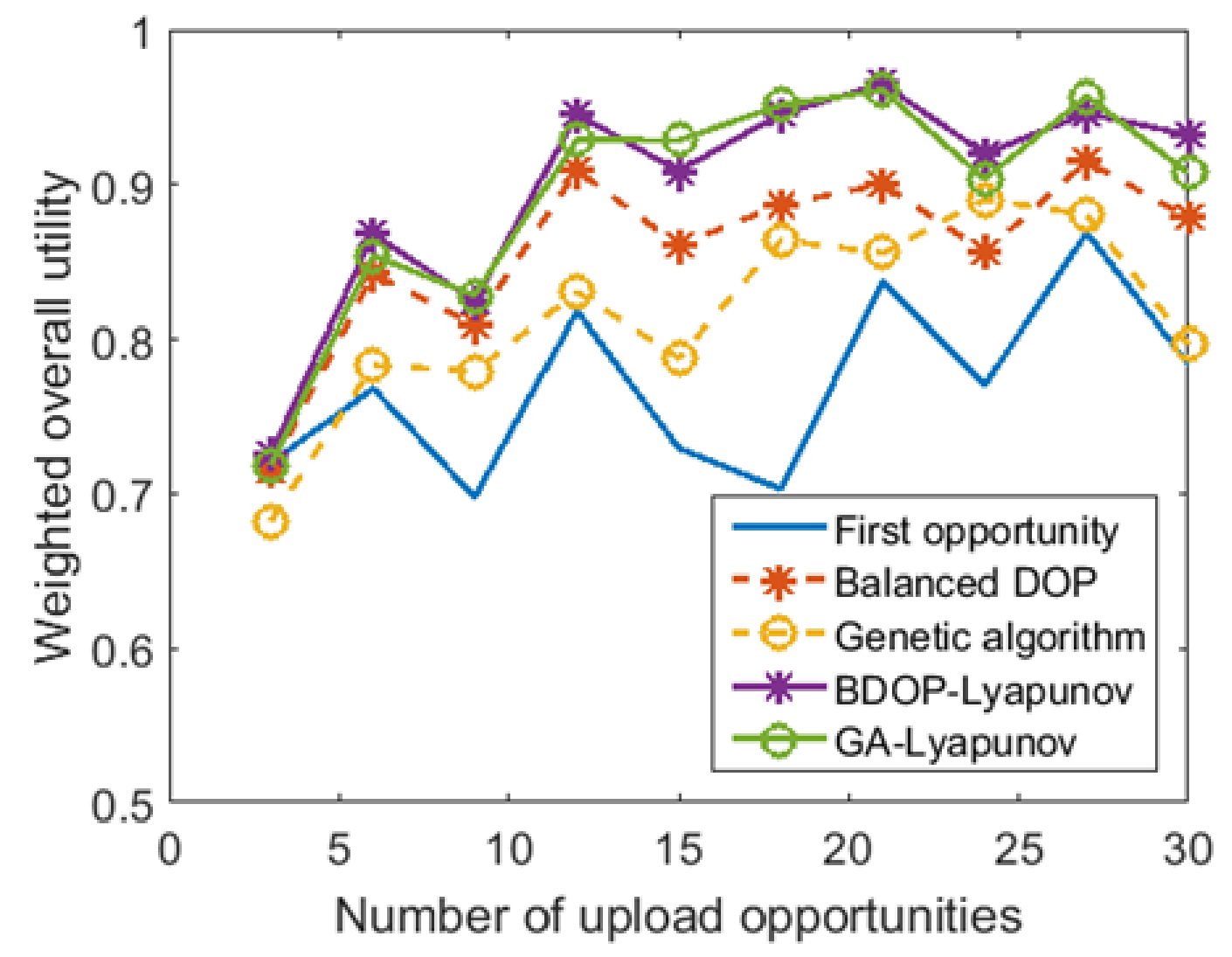
On UCI campus where we also have a SCALE deployment, we would like to create heat-maps with readings from multiple types of sensor on in-situ and mobile sensing devices.

- Campus Wi-Fi is available but coverage is non uniform
- Use of available knowledge
- Collaboration of heterogeneous sensing devices

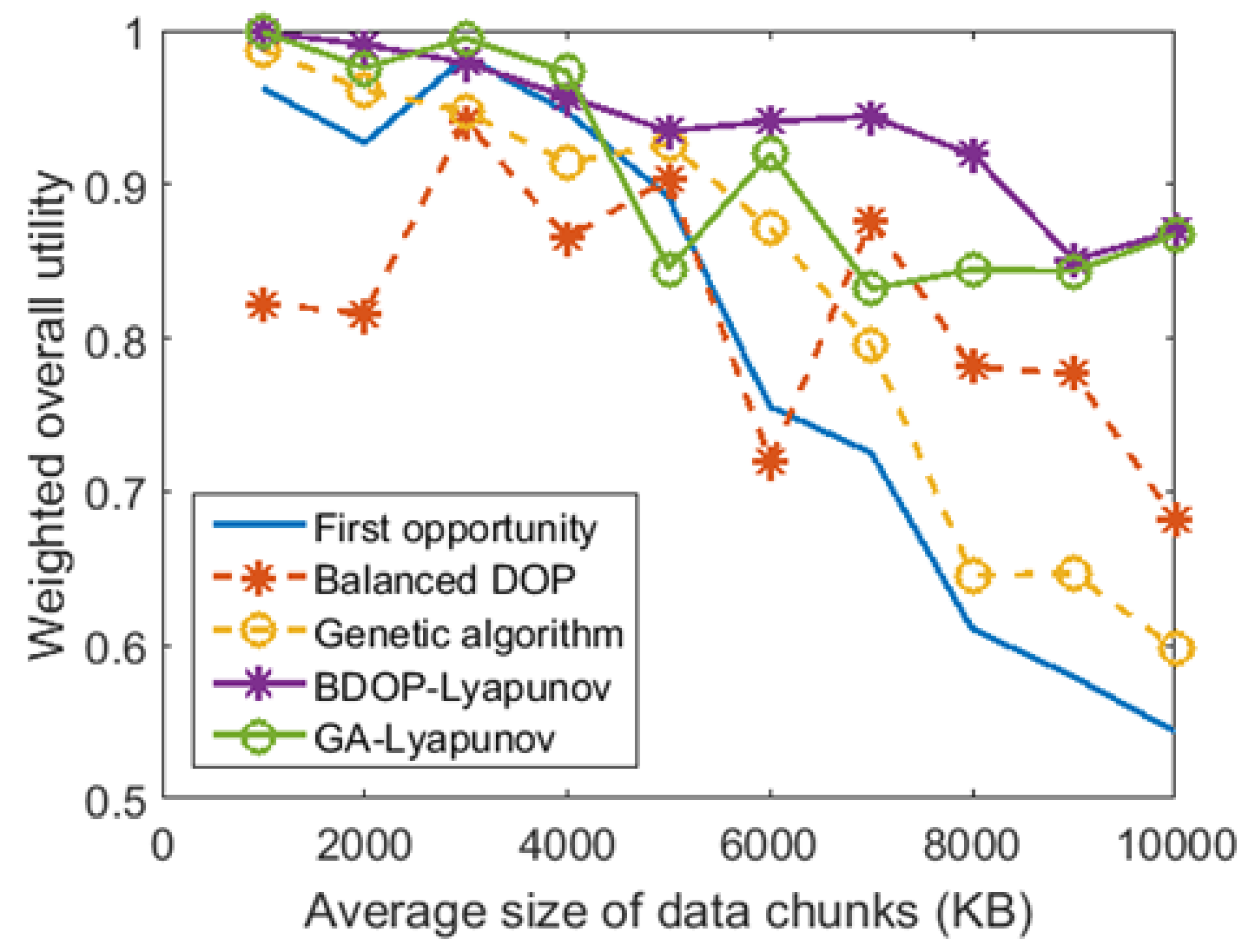
In emergencies (e.g. fire), we hope collect relevant data, but infrastructures are often damaged or congested in these scenarios.

- Some sensors (e.g. poisonous gas) only useful in such events
- Collecting info and leveraging the working infrastructure
- Data exchange for isolated communication islands

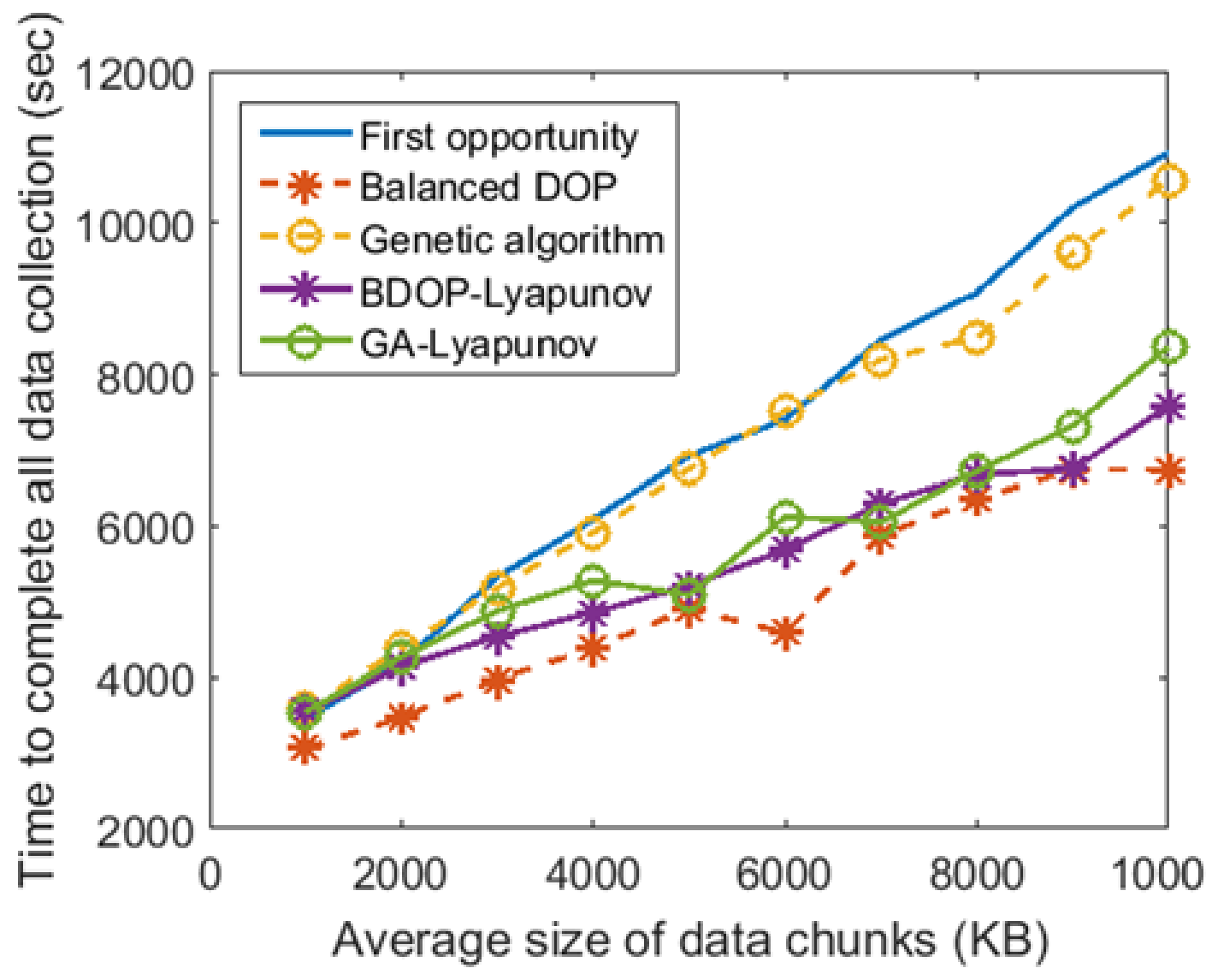
Experimental Results



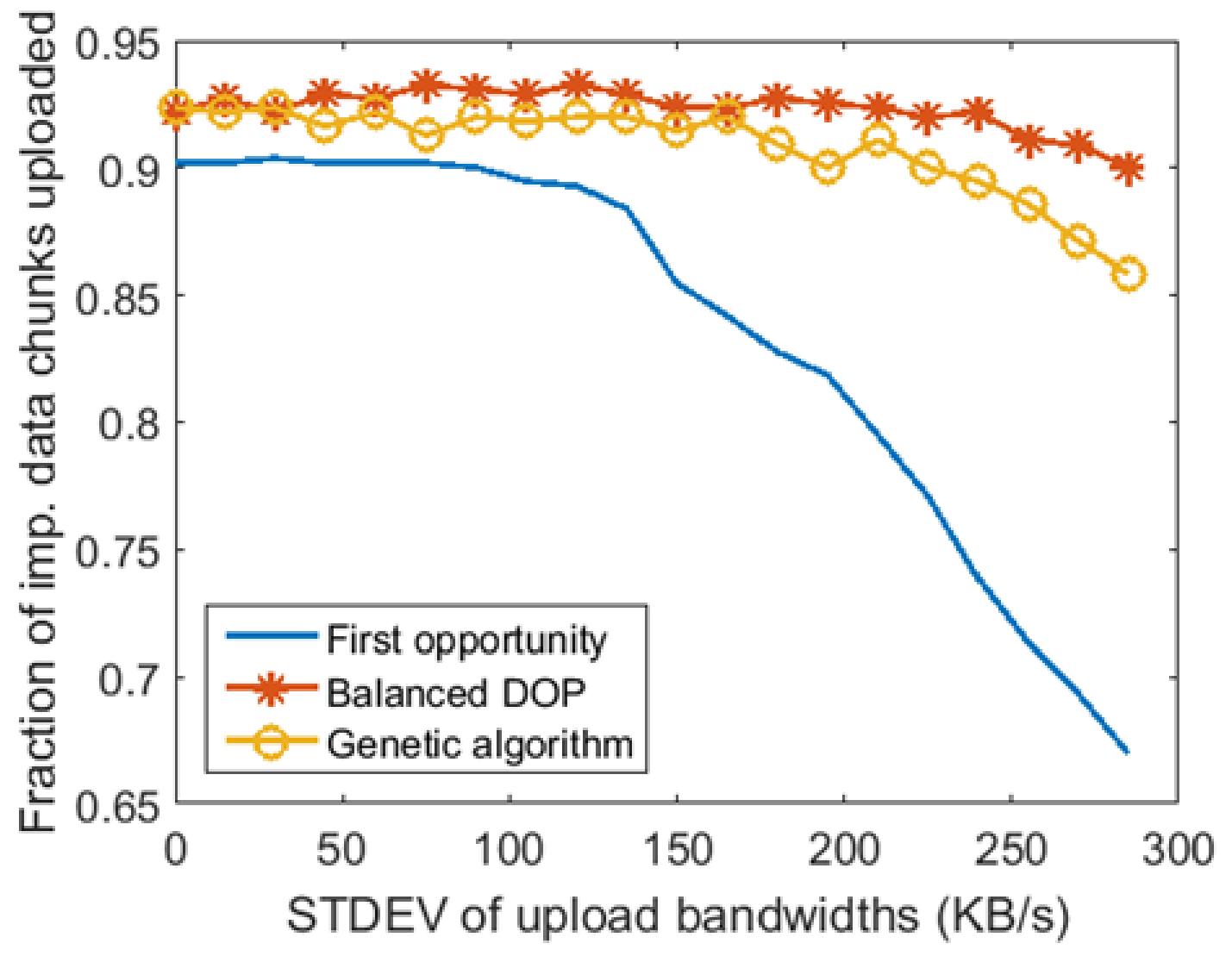
The proposed two-phase approach performed better than both naïve approach and the static planning only approach, resulting in 14-24% improvement in weighted overall utility (WOU).



The proposed two-phase approach also performed more stable for large data chunks. For ~8 MB chunks, BDOP-Lyapunov resulted in 36-60% improvement in WOU.



Our approach also saves time for data collection! Compared with the naïve approach (first-opportunity), BDOP-Lyapunov saves up to 30% of time in completing all data collection tasks.

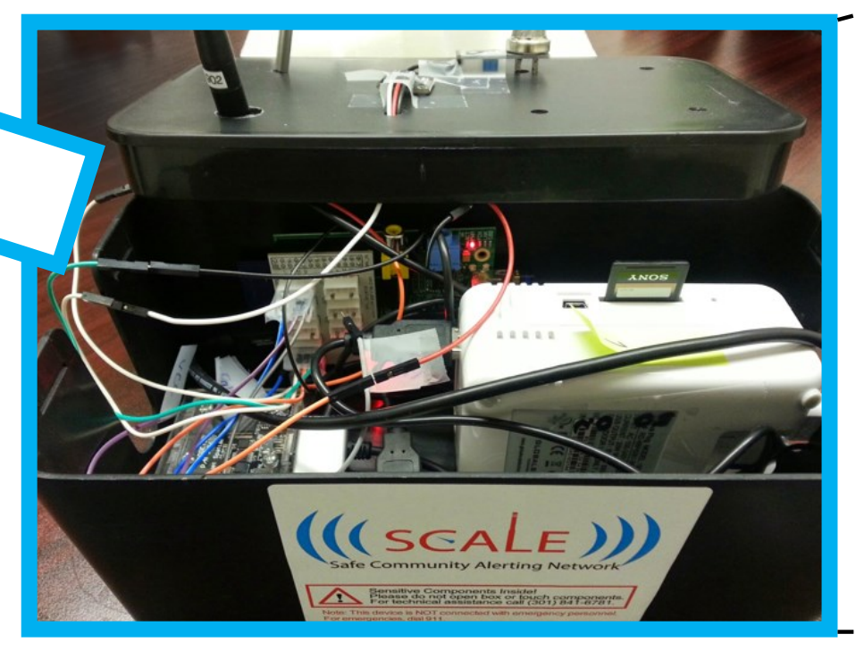


In a non-uniform wireless networking environment, our static planning algorithm (BDOP) avoids APs with poor connections, and make full use of those with good connectivity.

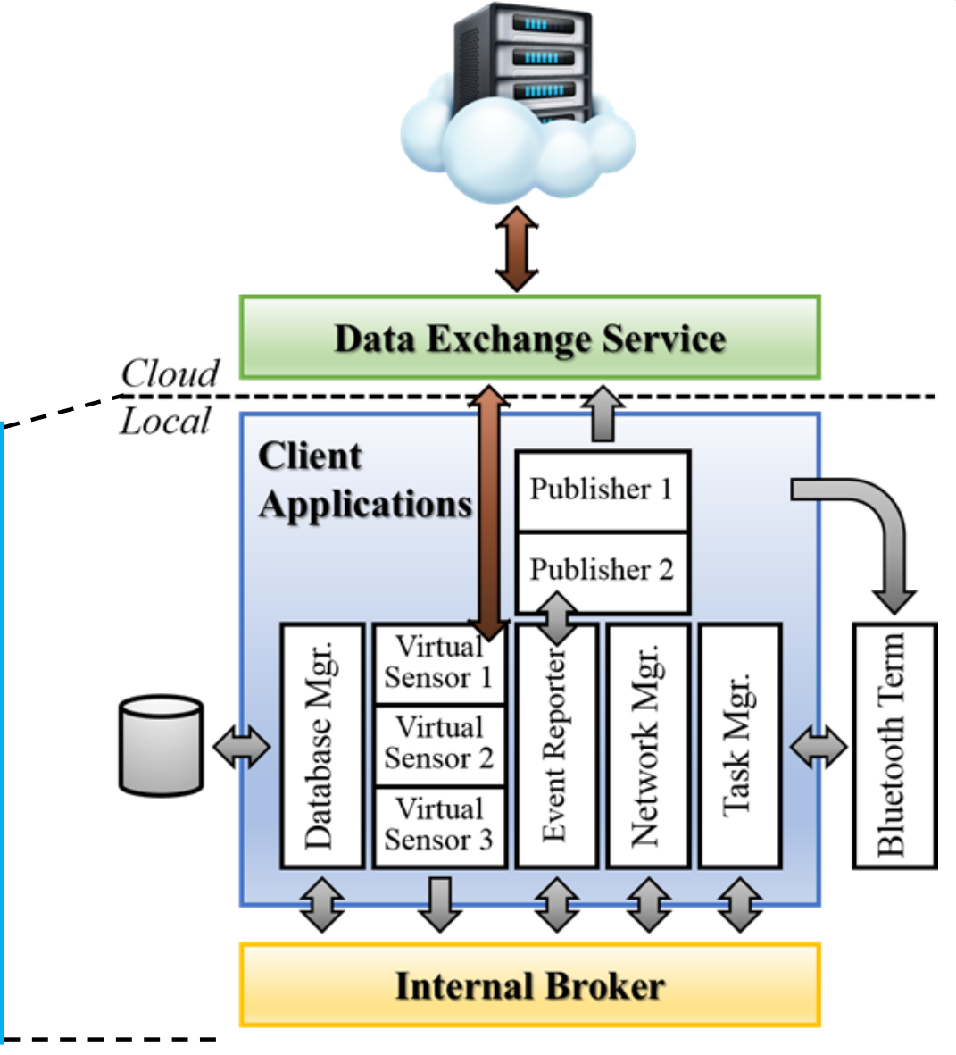
BDOP-Lyapunov performed stably when scenario scales up, which out-performed GA.

SCALECycle Platform

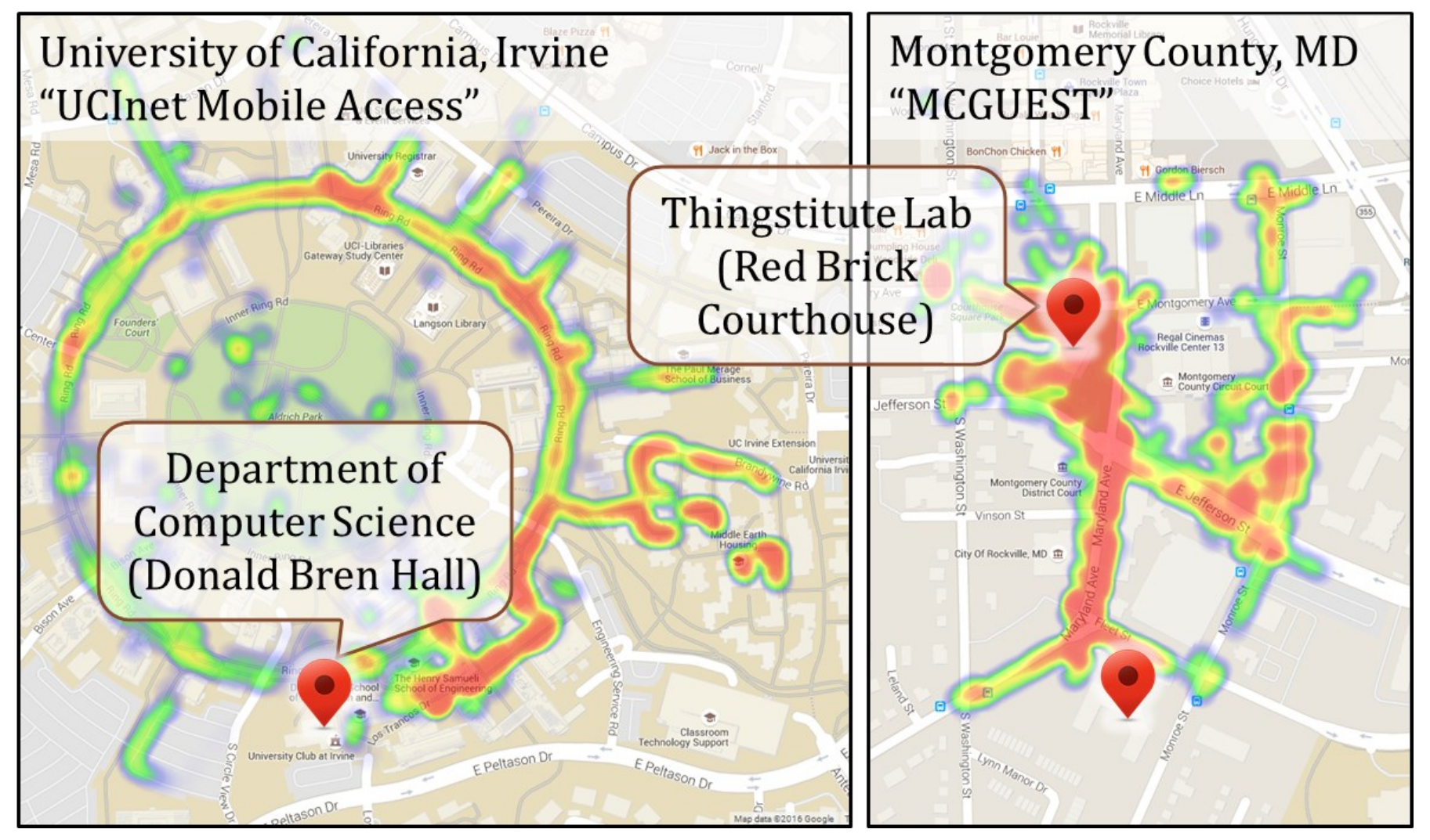
SCALECycle prototype on bike



An open SCALE sensor box



SCALECycle system architecture



Wi-Fi RSSI heat-maps created with SCALECycle measurements

Hardware prototype

- Raspberry Pi Model B
- Various sensors + Wi-Fi adapter
- Bluetooth GPS and terminal

Software

- Abstracted virtual sensors generate sensed events, that are published via MQTT or stored in local MySQL

Measurements

- Wi-Fi heat-mapping on two testbeds

Multi-timescale scheduling for crowd augmented urban sensing

Heat-mapping of real-time sensor data (e.g. air quality, noise pollution) is a commonplace application for smart city. Crowd sensing is a mechanism that leverages the sensing capacity on personal mobile devices to feed such applications. Due to the heterogeneous nature of the physical variables that we are interested in, different types of sensor data usually have different timing (update rate) and accuracy requirements. In this work, we look at building

- A flexible and scalable system architecture to support a large number of heterogeneous devices, and fulfills real-time user queries on demand
- A scheduling mechanism that decides which sensors on what devices to use for each time slot to reduce the cost, and handles the timing requirements of different data types

Probabilistic Communication for Mobile Data Collection in IoT Islands

In structures like smart buildings and parking lots, there is a large number of connected IoT nodes. In special events like emergencies, where the public infrastructure breaks down, these nodes are disconnected from the cloud and form communication islands. To facilitate data exchange among islands, and between islands and cloud, we can dispatch mobile agents to provide them with data collection or network access. Due to the heterogeneity of devices, we look for

- A local messaging mechanism that collects and fuses local data, selects the sink node, and delivers them to the cloud server or data collector in a timely manner
- A scheduling and dispatching algorithm, given the data exchange requirements, assigns roles to mobile agents and plans their trajectories

Future Works

Future works will concentrate on security and privacy issues that come up with smart communities/ cities, mobile sensing, participatory crowd-sensing, and complex network topologies. It is also important to enhance the backend services and the autonomous agents to support a complete feedback loop of a smart community.

This project is funded under National Science Foundation Award No. CNS 1450768.
Thanks to Daniel Hoffman from Montgomery County, MD for providing testbed and devices.