Mobility-Assisted Energy-Aware User Contact Detection in Mobile Social Networks

Wenjie Hu*, Guohong Cao*, Srikanth V. Krishnanamurthy†, and Prasant Mohapatra‡

*: Department of Computer Science and Engineering, The Pennsylvania State University
†: Department of Computer Science and Engineering, University of California, Riverside
‡: Department of Computer Science, University of California, Davis
Motivation

- User contacts in Delay Tolerant Networks (DTNs)

- Routing in DTN needs
  - User contact possibility, frequency
  - Community structure
Motivation

- User contacts in Mobile Social Network
- Flu prevention
- Sociability
User Contacts Detection

- Bluetooth user contact detection
  - To be discovered: discovery (discoverable) mode
  - To discover others: Bluetooth scan

### Table 1. Power consumption (mW)

<table>
<thead>
<tr>
<th>Modes</th>
<th>Mean</th>
<th>Std. dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>idle</td>
<td>23.43</td>
<td>2.92</td>
</tr>
<tr>
<td>Bluetooth on</td>
<td>25.23</td>
<td>1.36</td>
</tr>
<tr>
<td>BT Discovery</td>
<td>140.10</td>
<td>21.56</td>
</tr>
<tr>
<td>BT Discov. + Server</td>
<td>141.06</td>
<td>41.42</td>
</tr>
<tr>
<td>BT Scan (process)</td>
<td>307.59*</td>
<td>32.29</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>1.78*</td>
<td>0.86</td>
</tr>
</tbody>
</table>

* means the offset of power consumption.

Power Consuming!!!
Existing Work

- Adaptively adjust Bluetooth scan intervals
  - Decrease BT scan interval when detecting contacts
  - Increase BT scan interval otherwise

Why?

- Power consuming
- High miss rate
- High error to estimate contact duration
Our Insight

- User contact is related with user movement
- Detecting user movement consumes little power

Table 1. Power consumption (mW)

<table>
<thead>
<tr>
<th>Modes</th>
<th>Mean</th>
<th>Std. dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>idle</td>
<td>23.43</td>
<td>2.92</td>
</tr>
<tr>
<td>Bluetooth on</td>
<td>25.23</td>
<td>1.36</td>
</tr>
<tr>
<td>BT Discovery</td>
<td>140.10</td>
<td>21.56</td>
</tr>
<tr>
<td>BT Discov. + Server</td>
<td>141.06</td>
<td>41.42</td>
</tr>
<tr>
<td>BT Scan (process)</td>
<td>307.59*</td>
<td>32.29</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>1.78*</td>
<td>0.86</td>
</tr>
</tbody>
</table>

* means the offset of power consumption.
Relationships Between Contact Changes and User Movement

- **Contact change**: the start and end of a contact

**Experiment settings**
- 20 students in PSU, 10 days (8am-8pm)

**Trace collection**
- **Contact trace** (Bluetooth scan results)
  - Every 30 seconds, record detected neighbors
- **Movement trace** (Accelerometer results)
  - 50 Hz, record mean, std. dev, peak number of three axes within 5 sec
Relation 1: Contact Changes vs. Start Points

- Start point: User begins to move
- Contact change

- Time: ~ Exp. Dist.
- Time from start point to contact changes follows exp. distribution
- Probability to detect contact changes from the start point drops quickly
Relation 2: Contact Changes vs. End Points

- Contact changes happen closely to the end points

![Graph showing contact changes vs. end points]

- End point: User stops to move
- Contact change
- Time: 60 sec

Graph: Number of contact changes vs. time before stop point (sec)
Relation 3: Estimate Communication Range Using BT RSSI

- There is high possibility to lost a neighbor when $RSSI < -60dB$

- Lost RSSI: the last BT RSSI before lost contact

![Diagram showing Bluetooth devices A and B with CDF graph]
MAUC Design

- Mobility-Assisted User Contact detection
  - Restricts BT scan in moving periods
Moving User-Moving Periods Detection

- Accelerometer data
  - Mean, std. dev., and peak number of three axes
- Decision tree to classify user mobility status
  - Moving
  - Static

<table>
<thead>
<tr>
<th></th>
<th>Static</th>
<th>Moving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>93%</td>
<td>7%</td>
</tr>
<tr>
<td>Moving</td>
<td>1.4%</td>
<td>98.6%</td>
</tr>
</tbody>
</table>
Moving User-User Contact Detection

- Adaptive adjusts BT detect interval within moving period
  - Exponential increase: $Int_{i+1} = Int_i \times e^{backoff}$
  - Multiplicative decrease: $Int_{i+1} = Int_i / k$

- End point detection (EP)

<table>
<thead>
<tr>
<th>Detection rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAUC with EP</td>
</tr>
<tr>
<td>MAUC w/o EP</td>
</tr>
</tbody>
</table>
Moving user $B$ notifies static user $A$ about his movement

- Once when $B$ first detects $A$
- Every $\Delta$ time when $B$ finds $A$’s $rssi < \text{lost-RSSI threshold}$ until
  - $B$ moves back, or moves out of the $A$’s communication range, or stops
Static User-User Contact Detection

- When receiving notifications from unknown neighbors, A’ll
  - Detect immediately once

- When receiving notifications from existing neighbor, say B, A’ll
  - Sets B’s timer by $\Delta$
  - Detect when timer expires
How to Save More Energy?

<table>
<thead>
<tr>
<th>Energy</th>
<th>Mean</th>
<th>Std. dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluetooth Discovery</td>
<td>140.10</td>
<td>21.56</td>
</tr>
<tr>
<td>Bluetooth Scan (12s)</td>
<td>307.59</td>
<td>32.29</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>1.78</td>
<td>0.86</td>
</tr>
</tbody>
</table>
E-MAUC Design

- Energy-aware MAUC
  - All users are *synchronized* and use the same Bluetooth discovery schedule

- Bluetooth discovery interval $T$
  - min BT scan interval of MAUC

- Bluetooth discovery time $T_d$
  - max BT scan duration

- Adjust the scan time of MAUC to the nearest time when discovery mode is enabled
E-MAUC Design (cont’d)

Challenge

- Original Android OS needs user’s permission to set Bluetooth in discovery mode

Solution

- Change the Android OS to remove the limitation
- Propose adaptive Bluetooth discovery mode
  - Provides an interface to set Bluetooth turn on/off discovery mode based on users’ schedule
Performance Evaluation

- Comparison algorithms
  - SociableSense: adjust BT detection rate
  - STAR: adjust BT detection number in next $m$ minutes
  - AMD: detect every 30 seconds within moving periods

- Performance metrics
  - Accuracy: detection rate
  - Energy consumption: number of Bluetooth scans, power
MAUC can improve detection rate by 21% and 32% than SociableSense and STAR
Energy Comparison

- MAUC can reduce more than 57% of Bluetooth scans
- The power reduction is not obvious, due to the high power Bluetooth discovery mode
Energy Improvement of E-MAUC

- E-MAUC saves 45.6% power than MAUC
Conclusion

- Problem
  - Detect user contact with high accuracy and save energy

- Finding
  - User contact is closely related with user movement

- MAUC
  - Mobility-Assisted accurate user contact detection

- E-MAUC
  - Save more energy than MAUC and keep the accuracy
Thank You

Questions?