Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

# Characterizing and modeling people movement from mobile phone sensing traces

Long Vu [a,*], Phuong Nguyen [b], Klara Nahrstedt [b], Björn Richerzhagen [c]

[a] *IBM T. J. Watson Research Center, NY, USA*

[b] *Department of Computer Science, University of Illinois, IL, USA*

[c] *Technische Universität Darmstadt, Darmstadt, Germany*

ARTICLE INFO

*Article history:*
Available online 10 December 2014

*Keywords:*
People movement prediction
WiFi trace
Bluetooth trace

ABSTRACT

With the ubiquity of mobile phones, a high accuracy of characterizing and modeling people movement is achievable. The knowledge about people's mobility enables many applications including highly efficient planning of cities' resources and network infrastructures, or dissemination of safety alerts. However, characterizing and modeling people movement remain very challenging due to difficulties in (a) capturing, cleaning, analyzing and storing real traces, and (b) achieving accurate predictions of different future contexts.

In this paper, we present our effort in measuring and capturing phone sensory data as real traces, cleaning up measurements, and constructing prediction models. Specifically, we discuss design methodology, learned lessons from the implementation and deployment of a large-scale scanning system on 123 Google Android phones for 6 months at University of Illinois campus. We also conduct a characterization study on collected traces and present new findings in location visit pattern, location popularity, and contact pattern. Finally, we exploit joint location/contact traces to derive: (1) predictive models of missing contacts, and (2) prediction framework that provides future contextual information of people movement including locations, stay duration, and social contacts.

## 1. Introduction

For many years, social scientists have been observing and tracking people, their movements and overall mobile communities, collecting coarse information via non-digital tracking methodologies such as surveys and questionnaires. From these observations interesting socio-economic models came out, but they were coarse, with time-scales ranging from months to years [1]. With the pervasiveness and digital connectivity of mobile devices, we can (1) collect large amounts and fine-granularity of multimodal and detailed sensory data about people movement, (2) discover new characteristics of people movement from these collected traces, (3) validate existing socio-economic models with high accuracy, and (4) derive novel predictive mobility models.

The ability to accurately predict people movement is crucial to numerous domains and applications including network resource planning, human–computer interaction, socio-economic modeling for urban planning, public transportation planning, public safety assurance and other application domains [2]. Due to the ubiquity of mobile devices, characterizing and modeling people movement are now achievable. While predicting the movement of people, many approaches seek

[*] Corresponding author.
*E-mail addresses:* lhvu@us.ibm.com (L. Vu), pvphuong@illinois.edu (P. Nguyen), klara@illinois.edu (K. Nahrstedt), bjoern.richerzhagen@kom.tu-darmstadt.de (B. Richerzhagen).

http://dx.doi.org/10.1016/j.pmcj.2014.12.001

answers to fundamental contextual questions such as: (1) Where will the person stay at a future time (i.e., location)?; (2) How long will he stay at the location (i.e., stay duration)?; (3) Who will he meet (i.e., contact)?

Providing answers to these questions remains challenging due to the: (a) complex nature of people movement, (b) difficulties in efficiently collecting and analyzing realistic people movement traces, and (c) complexities in constructing accurate predictive models of people movement. For example, several projects collected contact traces using portable devices such as iMote, cellphone, PDA [3–7]. These traces could be used to answer the question about future contacts. However, these traces did not have the location information and thus could not be used to answer location and stay duration at a particular location.

Several projects collected laptop traces and used them toward prediction models to predict location of people movement, but not stay duration or contact. For example, a large number of previous papers used association traces between the laptop/PDA and the Wifi access points (i.e., WLAN traces) to derive and evaluate their location predictors [8]. However, there was a fundamental weakness of using WLAN traces [9] in constructing location predictor since the laptop user did not always turn on the laptop and did not always carry it with her. So, the WLAN traces could be potentially used to understand the wireless usage rather than to accurately predict the location of people. Several other projects used past GPS coordinates [7] or cellular base station ID [10] to predict future locations. However, GPS coordinates may be inaccurate if the devices that collected the GPS traces are indoor while the use of cellular base station ID may not always provide accurate location prediction, since the transmission range of cellular base stations varies from meters (e.g., 500 m) to kilometers (e.g., 30 km). Other prediction methods answered the questions about location and stay duration [11–13]. McNamara et al. predicted the stay duration of commuters to select the best sources of the media content [11].

In this paper, we discuss methodologies about how to collect real traces on mobile phones, analyze the collected traces, and construct prediction models from these traces to provide answers to three questions above. Specifically, we first present the implementation of a scanning system on Google Android phones to collect WiFi access point MAC addresses and Bluetooth MAC addresses of Bluetooth-enabled devices in the proximity of experiment participants [14,15]. We then conduct an extensive characterization study on collected WiFi/Bluetooth traces, which shows that people usually visit regular places and make regular contacts in their daily lives. We observe that WiFi access point information can be used to infer location while Bluetooth MAC information can be used to infer contact [3]. So, we exploit joint location/contact traces and propose two novel predictive models. The first one infers missing contacts from the collected traces while the second predicts future people movement [15].

The paper encompasses three major sections after Section 1. In Section 2, we discuss the methodology, implementation, and large-scale real deployment of a sensing system on mobile phones. In Section 3, we characterize the collected sensing traces. In Section 4, we present a context-aware predictive model of missing contacts and a context-aware predictive model of people movement. Finally, we conclude the paper in Section 5.

## 2. Collecting people movement trace using mobile phones

In this section, we propose a methodology about the design of a trace collection system on mobile phones. Then, we apply the methodology in the implementation of a scanning system on Google Android phones. We also discuss how our system achieves the methodology and perform a sensitivity analysis on scanning period, a critical parameter that significantly impacts the quality of collected traces. Finally, we present learned lessons from a real large-scale deployment at the University of Illinois campus. In what follow, we use the terms "sensing" and "scanning", "dataset" and "trace", interchangeably.

### *2.1. Design methodology*

Mobile phones nowadays come equipped with advanced "sensors" that can capture (or sense) a wide range of contextual information. Typically, a mobile phone can capture audios/videos, GPS coordinates, cellular base station identity, WiFi scans, and Bluetooth scans. Further, a mobile phone can measure proper acceleration, detect motion, and so on. These contextual measures can be used to infer not only people movement patterns but also their personal lives. In theory, we can always turn on all phone sensors and collect all types of sensing data. However, in practice, we have several issues: (1) turning on many sensors quickly drains the phone's energy, for example a phone may run out of energy if its WiFi connection is on for 8 h, (2) sensing data must be collected for a long enough period, with few missing values, to be useful for analysis. Therefore, we propose the following methodology:

*Sensor selection*. Selecting right sensors for the sensing task is the most critical factor. A good sensing system should collect sensing traces that can be used to infer both location and contact information of people movement. More importantly, selected sensors should complement each other in terms of their collected information. For example, a scanning system may not include a WiFi scanner and a GPS coordinate scanner since they both only provide location information. Further, we need to understand the tradeoff between the quality of the sensing traces one sensor collects and the amount of energy it consumes. Some sensor captures high quality sensing data but might consume too much energy for a prolonged sensing task. As a result, we may have to use sensors that provide less quality data in order to capture longer sensing traces. Once the sensors are determined, sensing applications can be implemented.

*Scanning period*. Given the sensing applications, the next step is to decide how frequently each sensor collects its sensing data [16]. This is crucial since it directly impacts the quality of collected traces. A typical mobile phone without sensing

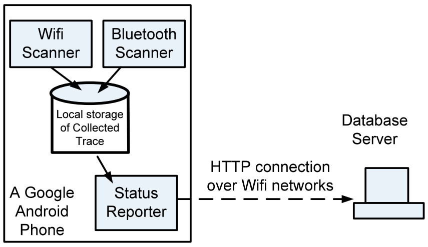
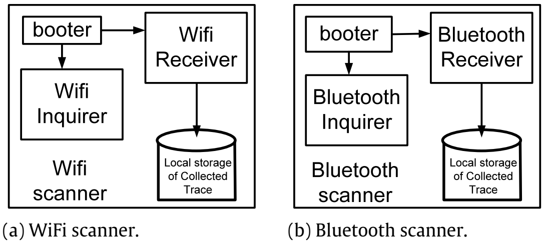


**Fig. 1.** The implemented scanning system architecture.



(a) WiFi scanner. (b) Bluetooth scanner.

**Fig. 2.** The implemented WiFi scanner and Bluetooth scanner.

applications may need to be recharged every two or three days. In order to obtain prolonged non-broken traces, phone carriers must remember to recharge their phones. As shown in our real deployment, if a phone carrier does not use the phone as her daily phone, she would likely forget to recharge it. To make the phones usable for the carriers, we need to ensure that the sensing applications do not unreasonably drain the phones. On the other hand, if we set too long scanning periods, the collected sensing data may not provide the needed granularity of people movement.

*Decoupled sensing applications*. Collecting sensing data on mobile phones is a "best-effort" task and we always have to be prepared for the worst scenario, i.e., the implemented sensing applications may fail due to unanticipated reasons. A sensing application essentially is a software, which coexists and competes with other phone applications for resources, and thus it may crash or halt at anytime. If we bundle all sensing applications into one single application, and one of the sensing components crashes, then it is likely that the entire sensing application fails. If this happens, no sensing data is collected.

*Minimizing interference*. Phone users usually install many phone apps. A good sensing application should incur little interference on other phone applications and should not interrupt the usage of the phone users. In other words, a sensing application must: (1) start by itself whenever the phone reboots (robustness), (2) run in the background and not display messages on the phone GUI (transparency), (3) keep running even if other applications halt or crash (resilience).

In the next section, we apply this methodology to the implementation of a scanning system on Google Android phones.

### 2.2. Implementation of a scanning system on Google Android phones

In this section, we present our implementation of a scanning system on Google Android phones named UIM, which stands for University of Illinois Movement. As discussed above, the first step is to choose the sensors that can capture location and contact information of people movement. We choose to implement a WiFi scanner and a Bluetooth scanner since WiFi scans can be used to infer location while Bluetooth scans can be used to infer social contact. Fig. 1 shows our system architecture, which has a WiFi scanner, a Bluetooth scanner, a database server for sensing data storage, and a Status Reporter for sensing status update. Since the Status Reporter consumes extra energy, it is only optionally turned on by several experiment phones. Next, we present the WiFi scanner and the Bluetooth scanner in detail.

#### 2.2.1. WiFi scanner

The WiFi scanner is shown in Fig. 2(a) with three decoupled components: a booter, a WiFi inquirer, and a WiFi receiver. Each component runs as a separate process and interacts with each other via the message passing mechanism within the Google Android phone operating system (OS). WiFi scanner runs as a background service, anytime the phone restarts, the phone OS triggers the booter, which starts the WiFi inquirer and the WiFi receiver. This design achieves robustness since anytime the phone reboots, the WiFi scanner can start its scanning work automatically. The inquirer and the receiver work in an asynchronous fashion in which the inquirer uses a request timer to periodically (i.e., every 30 min) issue a WiFi scanning request to the phone OS. After sending the request, the inquirer goes to sleep, and wakes up for the next request when the request timer expires. On the other hand, the receiver always sleeps and is only waken up by the phone OS whenever the WiFi scans are available for collection. Upon receiving a WiFi scan that includes a set of MACs of WiFi access points in proximity of the experiment phone, the receiver writes the WiFi scan and a timestamp to a log file (see Table 1), and then

**Table 1**
Example of WiFi trace $W$.

| Scan time | WiFi MACs |
|---|---|
| 03/08/10 09:20 | $a_1, a_3$ |
| 03/08/10 09:50 | $a_1, a_5$ |
| 03/08/10 10:20 | $a_6$ |
| 03/08/10 13:50 | $a_4, a_7, a_9$ |
| 03/14/10 08:20 | $a_1, a_3$ |

**Table 2**
Example of Bluetooth trace $B$.

| Scan time | BT MACs |
|---|---|
| 03/08/10 09:20 | $u_1, u_3$ |
| 03/08/10 09:21 | $u_1, u_3$ |
| 03/08/10 09:22 | $u_1$ |
| 03/08/10 13:50 | $u_4, u_9$ |
| 03/14/10 08:14 | $u_1, u_3, u_8$ |

goes to sleep. Our design also allows the receiver to *opportunistically* receive WiFi scans, which result from other usages of WiFi connectivity, since each time the WiFi connection is initiated, a WiFi scan is performed by the phone OS. Note that keeping WiFi connection up and issuing WiFi scanning requests is much more energy-consuming than receiving WiFi scans. In order to conserve phone battery, we configure the WiFi inquirer so that it only issues scanning requests from 7AM of a day to 1AM of the next day. As a result, we can collect most of people movement while saving phone energy.

There are two reasons the WiFi scanning period is set to 30 (min). First, our scanning system was deployed at a university campus where people usually stay in one location inside buildings for a long period (e.g., a class session is usually 50 min). Second, a higher WiFi scanning period may drain the phones quickly and make them unusable as daily phones for phone carriers.

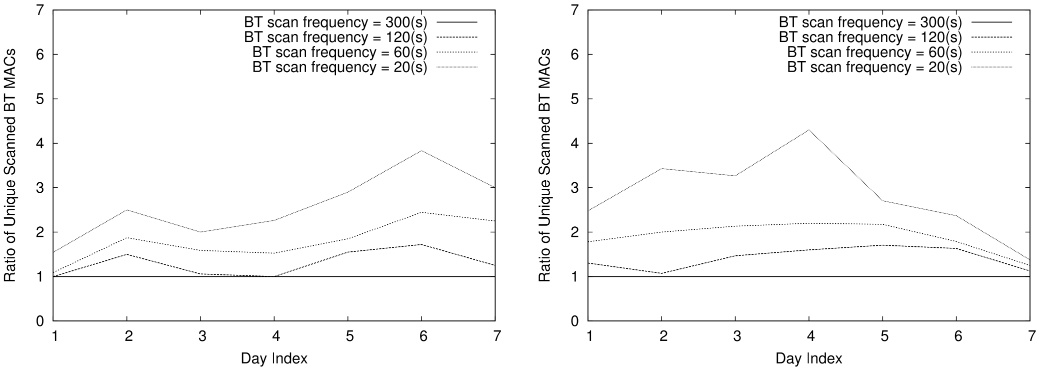
### 2.2.2. Bluetooth scanner

The Bluetooth scanner is shown in Fig. 2(b) with three decoupled components: a booter, a Bluetooth inquirer, and a Bluetooth receiver. Each component runs as a separate process and interacts with each other via the message passing mechanism within the Google Android phone OS. Similar to the WiFi scanner, the Bluetooth scanner is implemented as a background service. When the phone restarts, the phone OS triggers the booter, which starts the Bluetooth inquirer and the Bluetooth receiver. The inquirer and receiver work in an asynchronous fashion in which the inquirer uses a request timer to periodically (i.e., every 60 (s)) issue a Bluetooth scanning request to the phone OS. After sending the request, the inquirer makes the phone discoverable by other experiment phones (so that experiment phones can scan each other), goes to sleep, and wakes up for the next request when the request timer expires. The receiver, on the other hand, sleeps and is only waken up whenever a Bluetooth scan is returned by the phone OS and ready for collection. Upon receiving a Bluetooth scan that includes a set of MACs of Bluetooth-enabled devices in proximity of the experiment phone, the receiver writes the Bluetooth scan and a timestamp to a log file (see Table 2), and then goes to sleep. To conserve phone energy, the inquirer is configured to only issue scanning requests from 7AM of a day to 1AM of the next day. As a result, we can collect most of people movement while saving phone energy.

## 2.3. How does the scanning system meet design methodology?

UIM achieves the *design methodology* outlined in Section 2.1 as follows. First, the WiFi sensor and the Bluetooth sensor are the right choices since they provide location and contact contextual information. Other sensors (e.g., GPS coordinate scanner, cellular base station identity scanner) are available. However, the GPS scanner consumes much more phone energy while cellular base station identity scanner does not always provide accurate location information. Second, we set reasonable scanning periods. The WiFi scanner scans every 30 min since our system is deployed in a university campus where people usually stay long at one location. Meanwhile, our Bluetooth scanner has the shortest scanning period compared to previous works in university campus category as shown in Table 5. More importantly, our scanning periods conserve phone energy so that a recharge is needed only about in every two days, which is necessary for the participants to use experiment phones as their daily phones. Third, the WiFi sensing application and Bluetooth sensing application are implemented as separate software applications without any cross dependencies. Within each sensing application, three main components (i.e., the booter, the inquirer, the receiver) are decoupled and do not interfere with each other. This design provides reliability and robustness for our 6-month experiment. Fourth, not only the three main components of our sensing application do not interfere with each other, they all run in background and do not interfere with other applications or interrupt the usage of the phone users. Moreover, our sensing applications continue their scanning tasks if other phone applications crash.

Although we believe our design is clean, practical, and efficient, our methodology can be further improved. For example, the phone motion sensor can be used to detect body movement of carriers and thus a scanning request is made only when



(a) User 1: Bluetooth dataset.

(b) User 2: Bluetooth dataset.

**Fig. 3.** Impact of scanning period on collected Bluetooth data.

carriers are moving so that phone energy is further conserved. This feature requires an accurate body motion detection technique [17]. On the other hand, if the body movement is not accurately detected, the false positives may result in unnecessary scans and higher energy consumption.

### *2.4. Scanning period sensitivity analysis*

As we discuss in Section 2.1, the scanning period plays a pivotal role in any scanning systems on mobile phones. On one hand, the shorter scanning period provides more detailed collected trace that can lead to better knowledge of people movement. On the other hand, the shorter scanning period drains the phones faster and thus can result in broken traces. "Broken traces" mean traces where some sensing information is missing due to loss of phone power, failure of sensing application, temporal gaps in collected records, and so on. In this section, we study the tradeoff between scanning period and the quality of collected trace. Here, the quality of collected trace is measured by the number of collected WiFi MACs and Bluetooth MACs.

To this end, we did a focused experiment with two participants carrying experiment phones for a week, in which we set the scanning period of the Bluetooth scanner to 20 s and the WiFi scanner to 5 min. The participants are responsible to recharge the phone and keep the phones on during the experiment. Let $D_B$ and $D_W$ denote the collected Bluetooth and WiFi traces from the two participants. Notice that as of October 2010, the lower limit of Bluetooth scanning period is 12 s due to the hardware constraint on Google Android phones. Then, $D_B$ and $D_W$ are used to create other datasets of longer scanning periods as follows. For example, a dataset $D_{B1}$ with the period of 40 (s) is created by taking every other scanned record of the dataset $D_B$.

For each user, we create Bluetooth datasets for the scanning periods in the set $\Sigma_B = \{20\text{ (s)}, 60\text{ (s)}, 120\text{ (s)}, 300\text{ (s)}\}$. Let ${\sigma^b}_{300}$ be the number of unique Bluetooth MACs that is obtained per day with the scan period of 300 (s). ${\sigma^b}_{300}$ is the baseline in our analysis. For each period ${\sigma^b}_i \in \Sigma_B$, we calculate the ratio $\frac{{\sigma^b}_i}{{\sigma^b}_{300}}$ for each day (i.e., $i \in \{20, 60, 120, 300\}$). Fig. 3 shows that ${\sigma^b}_{20}$ can be up to 4 times of ${\sigma^b}_{300}$. Meanwhile, ${\sigma^b}_{60}$ is about double of ${\sigma^b}_{300}$.

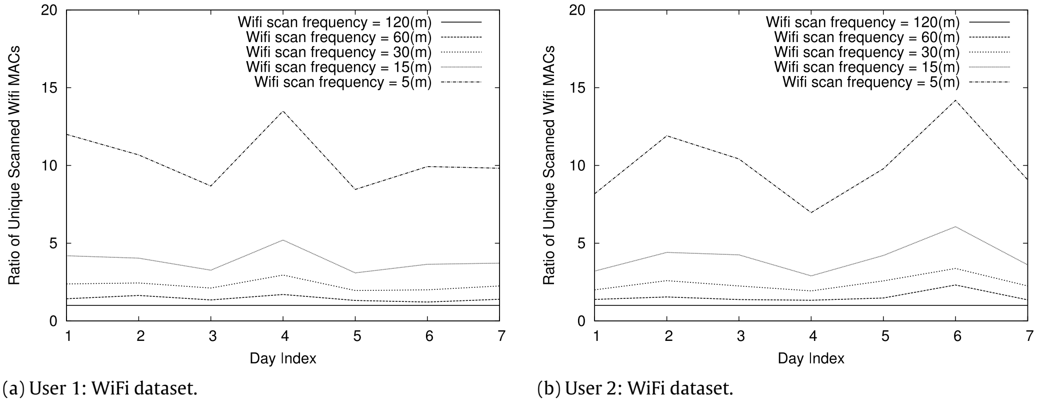
Likewise, we create WiFi datasets for the scanning periods in the set $\Sigma_W = \{5\text{ (m)}, 15\text{ (m)}, 30\text{ (m)}, 60\text{ (m)}, 120\text{ (m)}\}$. Let ${\sigma^w}_{120}$ be the number of unique WiFi MACs that is obtained per day with the scan period of 120 (m). ${\sigma^w}_{120}$ is the baseline in our analysis. For each period ${\sigma^w}_j \in \Sigma_W$, we calculate the ratio $\frac{{\sigma^w}_j}{{\sigma^w}_{120}}$ for each day (i.e., $j \in \{5, 15, 30, 60, 120\}$). Fig. 4 shows that ${\sigma^w}_5$ can be up to 14 times of ${\sigma^w}_{120}$. Meanwhile, ${\sigma^w}_{30}$ is about 2.5 times as much of ${\sigma^w}_{120}$.

In summary, our analysis shows a clear tradeoff between the scanning period and the amount of collected data. The scanning period should be set so that phones can perform prolonged experiment while collecting acceptable traces.

### *2.5. Learned lessons from a large-scale deployment*

We deployed our scanning system on Google Android phones carried by 123 participants for 6 months from March to October 2010 at the University of Illinois campus. From this large-scale deployment, we learn following lessons:

*Thorough application testing.* We learn that it is highly recommended to test the phones with installed scanning applications carefully before handling them to participants. This is because software failures occur on phones more frequently than one anticipates. For our work, software development took about three weeks while software testing took in total two months. In our very first experiment in February 2010, we tested the scanning applications on phones for one week and then assigned 15 phones to 15 participants. Within the first two days, most participants reported that their phones were "broken". These participants installed new applications and their uses of phones differed from how we tested these phones. We then had to re-develop and stress-test our scanning applications before the next experiment. It is also important to note that once phones are given to the participants, developers may not be able to take the phones back for bug fix or software upgrade since if the phone is taken back and re-assigned later, broken traces with temporal gaps may exist.



(a) User 1: WiFi dataset.

(b) User 2: WiFi dataset.

**Fig. 4.** Impact of scanning period on collected WiFi data.

**Table 3**
Overall characteristics of our collected UIM sensing traces.

| Overall characteristics | | | |
|---|---|---|---|
| Name of dataset | $D_1$ | $D_2$ | $D_3$ |
| Number of phones (participants) | 28 | 79 | 16 |
| Experiments | 03/01–03/20 | 04/08–05/15 | 05/24–08/16 |
| Number of days | 19 | 38 | 85 |
| Bluetooth scanning period (s, $\delta_B$) | 60 | 60 | 60 |
| WiFi scanning period (min, $\delta_W$) | 30 | 20 | 30 |
| Number of scanned BT MACs | 8508 | 17 080 | 7360 |
| Number of scanned WiFi MACs | 7004 | 29 324 | 6822 |

*Energy is critical.* A shorter scanning period provides more detailed traces but drains the phones faster. If the participant forgets recharging the phone (if the experiment phone is not her daily phone, the participant is likely to forget recharging it), we have broken and unusable traces. More importantly, a long-lived and energy-consuming application may not be preferable by the phone OS and may get killed unnoticeably. Setting the right scanning period thus become so important that it decides the success of the entire project.

*Handling privacy of participants.* As we got the phones back from the participants at the end of the prolonged experiment, a large amount of personal and sensitive information was available since our participants took personal photos, recorded videos/notes, made personal phone calls, etc. We promised our participants that only Bluetooth and WiFi sensing traces were collected and analyzed. Even with collected WiFi and Bluetooth sensing data, together with the associated timestamp, personal activities still could be inferred. To avoid information leak, we deleted all data on the phone immediately after the data was collected, and we anonymized WiFi and Bluetooth records prior to any analysis. Further, we signed the IRB paper work with our university and participants to guarantee that their personal information was protected.

## 3. Characterizing people movement

In this section, we compare our collected sensing traces with other traces collected at university campuses and workplaces. We then characterize our collected traces and present findings on contact analysis and location analysis.

### 3.1. Collected sensing traces

Table 3 summarizes major statistics of the sensing traces collected by the UIM system (i.e., UIM traces). Specifically, from March 2010 to August 2010, we conducted three rounds of experiments with 123 participants at the University of Illinois. Our participants included grads, undergrads, faculties, and staffs. The first experiment lasted 19 days, the second was 38 days, and the third was 85 days. The number of scanned WiFi MACs and Bluetooth MACs of the third experiment were fewer than the second experiment (although the third experiment was much longer) since the third experiment was conducted during the summer break with fewer classes and students on campus.

Tables 4 and 5 compare overall characteristics of our traces and other previously collected traces. Note that in these two tables, "# of In-contacts" means the number of contacts made between experiment participants while "# of Ex-contacts" means the number of contacts made between experiment participants and external people. Like ours, traces collected by the Reality Mining project at MIT [18] include both location and contact information while other traces only provide either location or contact information. For Reality MIT traces, the location information is inferred from the cellular base station

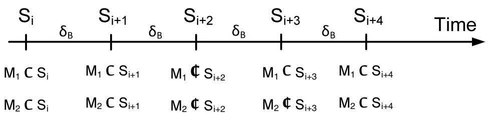


**Fig. 5.** Contact definition.

**Table 4**
UIM traces vs. previous traces in city, workplace, conference, corporation.

| | PMTR [20] | Intel [3] | Cam. city [21] | Info [22] | UIM |
|---|---|---|---|---|---|
| Environment | Work | Corp. | City | Conf. | UIUC |
| Duration (day) | 19 | 3 | 10 | 3 | 142 |
| # of devices | 49 | 8 | 36 | 41 | 123 |
| $\delta_B$ (s) | 1 | 120 | 600 | 120 | 60 |
| Contact trace | Yes | Yes | Yes | Yes | Yes |
| Location trace | No | No | No | No | Yes |
| Device type | PMTR | iMote | iMote | iMote | Phone |
| # of In-contacts | 11 895 | 1091 | 8545 | 22 459 | 171 812 |
| # of Ex-contacts | N/A | 1173 | 10 469 | 5791 | 285 637 |

**Table 5**
UIM traces vs. previous traces at other university campuses.

| | Cam. U. [3] | MIT [18] | Toron. [23] | UCSD [24] | UIM |
|---|---|---|---|---|---|
| Environment | University campus | | | | |
| Duration (day) | 5 | 246 | 16 | 77 | 142 |
| # of devices | 12 | 97 | 23 | 273 | 123 |
| $\delta_B$ (s) | 120 | 300 | 120 | N/A | 60 |
| Contact trace | Yes | Yes | Yes | No | Yes |
| Location trace | No | CellID | No | AP | AP |
| Device type | iMote | Phone | PDA | PDA | Phone |
| # of In-contacts | 4229 | 54 667 | 2802 | 195 364 | 171 812 |
| # of Ex-contacts | 2507 | N/A | N/A | N/A | 285 637 |

identity associated with experiment phones. However, since the cellular base station transmission range varies from meters (e.g., 500 (m)) to kilometers (e.g., 30 (km)), the inferred location information may not achieve the needed fine granularity of the physical location. For our collected traces, location can be inferred from WiFi MACs [19]. Since the transmission range of WiFi Access Point is from 100 to 200 (m), the locations inferred from WiFi trace provide a much finer granularity.

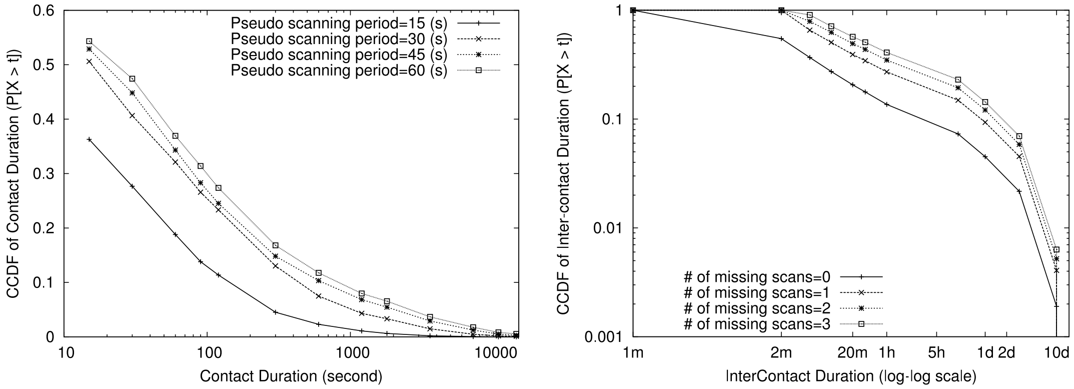
Table 4 shows that the PMTR project collected contact traces with scanning period of one second, which is the highest scanning period from all projects in Tables 4 and 5. However, due to its short scanning period, PMTR devices can only collect 19 days of contact traces (but without the location traces) before their pre-charged batteries drained. Our traces provide both location and contact information, with much more detailed information, which is clearly shown in the much higher number of internal contacts and external contacts captured by our scanning system.

## *3.2. Contact analysis*

### *3.2.1. Contact definition*

In our context, a phone $p$ and an ad hoc MAC $M$ are said to have a contact if $M$ exists in the Bluetooth scanned result of $p$. Let $T_C$ denote the contact duration between a phone $p$ and an ad hoc MAC $M$. $T_C$ could be calculated by using the scanning period $\delta_B$. For example, let $N$ be the number of $p$'s consecutive scans where $M$ appears in the scanned results, $T_C = N \times \delta_B$. However, due to the hardware limitation of the Bluetooth driver at the phone and the unreliable wireless communication channel, it is possible that $p$ does not receive $M$ in its scanned results even when $M$ is within Bluetooth sensing range of $p$. Therefore, in previous works [18,3,22], people accepted the missing scans in contact definition as follows: for $p$ and $M$, although $p$ does not see $M$ in its scanned result for a certain number of scans, $p$ and $M$ are still considered in contact if the number of missing scans is acceptable. Fig. 5 shows an example of contact definition. Let $S_i$ denote the scanned result of $p$ at time $t_i$. $M_1$ and $M_2$ are two ad hoc MACs scanned by the phone $p$. If the accepted number of missing scan for this figure is 1, from $t_i$ to $t_{i+4}$, $p$ and $M_1$ have one contact with the duration of $4\delta_B$ while $p$ and $M_2$ have two contacts with the durations of $2\delta_B$ and $\delta_B$ respectively.

The accepted number of missing scans depends on the trace collection procedures. For example, in [3,22], the number of missing scans is one (with $\delta_B = 120$ (s)); however, in [20], this number is 60 (with $\delta_B = 1$ (s)). To generalize this, we define $\Delta_B$ as the *accepted number of missing scans* in the definition of contact. The contact duration $T_C$ then depends on $\delta_B$ and $\Delta_B$. Notice that $\Delta_B$ defines the boundary between the two consecutive contacts.



(a) Contact duration.

(b) Inter-contact duration.

**Fig. 6.** Contact duration and inter-contact duration.

### 3.2.2. *Impact of $\delta_B$ on contact duration*

Fig. 6(a) shows the sensitivity of contact duration when we vary value of $\delta_B$. To obtain this plot, we have 6 students carry experiment phones for one week, we set $\delta_B = 15$ (s) for the Bluetooth scanner. Notice that the lower bound (hardware limitation) of Bluetooth scanning period for Google phone is 12 (s) [25]. We have tried the Bluetooth scan every 10 (s) and most of the time the scanned results are empty. Let $\Phi_1$ denote the dataset obtained from these 6 phones with $\delta_B = 15$ (s). Each element in $\Phi_1$ is the results obtained by one scan of one phone. Since $\delta_B = 15$ (s), we can derive $\Phi_2$ dataset from $\Phi_1$ using pseudo $\delta'_B \in [15, 30, 45, 60]$ as follows: if $\delta'_B = 30$ (s), $\Phi_2$ is the set of odd or even scans in $\Phi_1$. With $\delta'_B = 45$ (s), we take $i$th scan from $\Phi_1$ and put into $\Phi_2$, and skip the $(i + 1)$th and $(i + 2)$th scans. Fig. 6(a) is obtained from these $\Phi_2$ sets of corresponding pseudo $\delta'_B$ and $\Delta_B = 0$. Notice that $x$-axis of this figure is in log scale. The figure shows that different values of $\delta'_B$ lead to distinct curves. More importantly, although these curves look similar in shape, the difference among them is significant, ranging from 15% to 20%. That means, the Bluetooth scanning period $\delta_B$ has important impacts on calculating contact duration. This has not been investigated in previous studies [18,3,22,21].

Fig. 6(a) also shows that a large amount of contacts (from 35% to 55%) are short contacts (less than 15 (s)). Previous studies [18,3,22,21] have not studied the short contact distribution due to their long scanning periods (see Table 4). Except one study in a workplace environment [20], we are the first to study the distribution of short contact in university campus. As shown in [20], the short contact has an important role in data forwarding protocol in a workplace environment. In the future, we will investigate the impact of the short contact on data forwarding in the university campus environment.

### 3.2.3. *Impact of $\Delta_B$*

Besides $\delta_B$, $\Delta_B$ has an important role in defining the contact duration $T_C$. This section studies the impacts of $\Delta_B$ on inter-contact duration. Notice that the plots in this section are obtained from the dataset $D_1$ in Table 3.

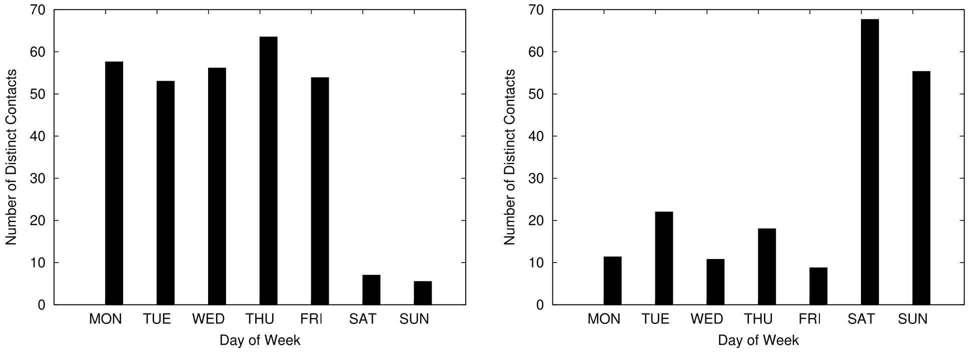
As defined in previous studies [3,22,21], inter-contact duration is the time duration between the two consecutive contacts of a given node pair. It is well-known from the previous studies that the inter-contact duration follows the power law [18,3,22]. Fig. 6(b) shows that overall the inter-contact duration follows the power law and about 60%–80% of inter-contact duration is less than 1 h. That means, if a pair of nodes meets at time $t$, this pair will meet again within one hour after time $t$ with high probability. This figure also shows that when $\Delta_B$ varies from 0 to 3, the inter-contact duration varies up to 15%, although the shapes of the curves are similar. So, the value of $\Delta_B$ has a clear impact on the inter-contact duration distribution.

In conclusion, the definition of contact depends on $\delta_B$ and $\Delta_B$. So, when using the (inter-)contact duration distribution reported in previous studies [3,22,21], the readers should carefully consider the corresponding values of $\delta_B$ and $\Delta_B$ since they have significant impacts on (inter-)contact duration distribution.

### 3.2.4. *Contact pattern*

Here, we study how contacts change between weekday and weekend. We select a set of 50 phones from three datasets $D_1, D_2, D_3$ in Table 3. Let $B_D$ be the Bluetooth traces collected by these 50 phones. Note that each of these phones collects from 19 days to 50 days. For each experiment phone $p$, we calculate the average number of unique Bluetooth MACs scanned by the Bluetooth scanner of $p$ for a particular day of a week such as Monday, ..., Sunday. We see four different contact patterns as shown in Figs. 7 and 8.

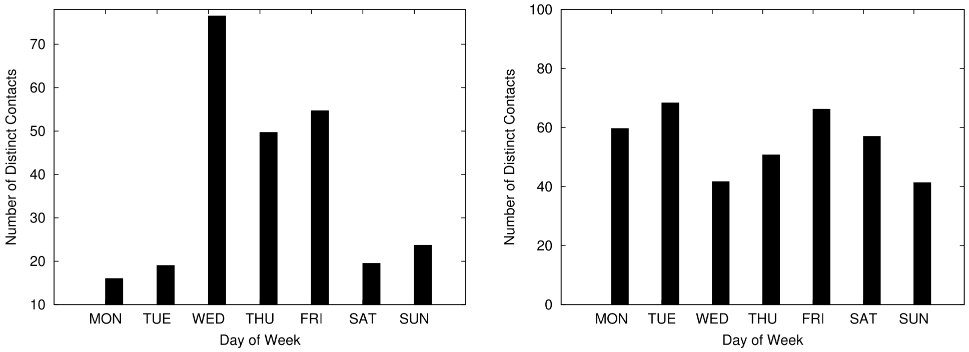
Fig. 7(a) shows the first contact pattern in which people usually have a considerably higher number of contacts during the weekdays than the weekends. This is the most common contact pattern found in our sensing traces since most people perform the casual routines at work for the weekdays when they make contacts with many more people. Working people usually spend time with family or for personal things at weekends and thus they meet fewer people. In contrast, Fig. 7(b) shows an opposite contact pattern in which people make more contacts during the weekends than the weekdays. This pattern does not appear frequently in our traces, however it exists.



(a) Fewer contacts at the weekend.

(b) More contacts at the weekend.

**Fig. 7.** The number of contacts decreases/increases at weekend.



(a) More contacts in midweek.

(b) Contacts stay in a small range.

**Fig. 8.** The number of contacts grows high during midweek or varies in a small range.

The third contact pattern exists for people who have the busiest schedule in the middle of the week as shown in Fig. 8(a). This figure looks similar to the bell shape and people with this pattern usually meet many more people during midweek. The last contact pattern is the most steady one as shown in Fig. 8(b). People, who belong to this pattern, make a similar number of contacts everyday, regardless of weekdays or weekends.

### 3.2.5. *Regular contact*

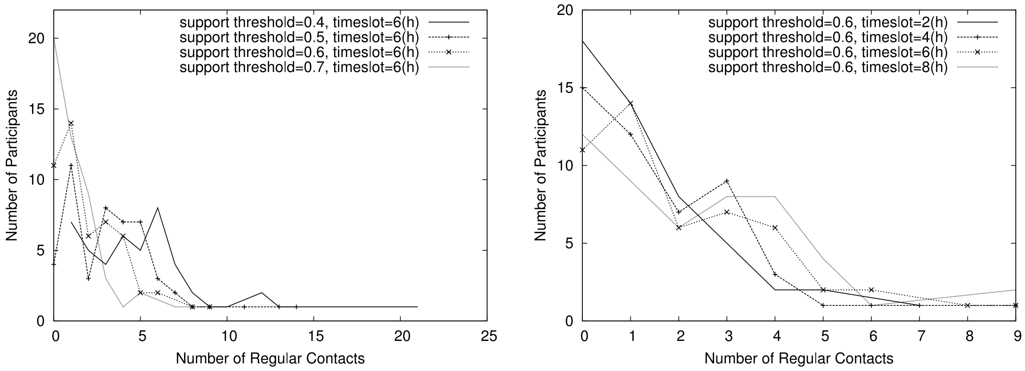
We study the regularity of contacts in the $B_D$ traces in Section 3.2.4. Given the Bluetooth traces of a phone $p$, we first find the set $U_p$ of all Bluetooth MACs scanned by the Bluetooth scanner of $p$. Then, we divide the course of a day into time slots of size $\tau$ hours. For example, with $\tau = 6$ (h), we have four time slots {[00:00,06:00), [06:00,12:00), [12:00, 18:00), [18:00,24:00)}.

For each phone $p$, let $D_p$ be the number of days in which $p$ collects the Bluetooth traces. For a scanned Bluetooth MAC $u \in U_p$, let $D_{up}$ be the number of days $p$ and $u$ have contacts, in which these contacts happen at the same time slot of these $D_{up}$ days. For example, $p$ and $u$ may have contacts every weekday at 8AM. In our context, *a contact between the phone p and a scanned Bluetooth MAC $u \in U_p$ is a "regular contact" if $D_{up} \geq D_p \cdot \epsilon_c$* (i.e., $0 \leq \epsilon_c \leq 1$), in which $\epsilon_c$ is a pre-defined threshold, denoted as the "regularity threshold". In other words, a contact is considered a regular contact if the phone $p$ and the Bluetooth MAC $u$ make contacts at the same time slot for at least $D_p \cdot \epsilon_c$ days during the experiment period of $D_p$ days.

Next, we set $\tau = 6$ (h) and vary $\epsilon_c$ from 0.4 to 0.7. Fig. 9(a) shows that when $\epsilon_c$ increases, the number of participants with more regular contacts decreases, which is intuitive. For $\epsilon_c = 0.5$, we have 45 participants with at least one regular contact. Then, we set $\epsilon_c = 0.6$ and vary $\tau$ from 2 to 8 (h). Fig. 9(b) shows that when $\tau$ increases, the number of participants with more regular contacts increases. This is intuitive since it relaxes the definition of regular contacts. When $\tau$ increases from 2 to 8 (h), we observe that from 33 to 40 participants (out of 50 participants) have at least one regular contact. In summary, our analysis shows that people make regular contacts during their daily activities.

### 3.2.6. *Contact group*

In this section, we study the contact made by a group of people and seek answers to the question: what is the grouping behavior of people at a particular location over a time interval (i.e., temporal flow of people at a certain location)?. As shown in Table 2, each Bluetooth scanned record $r_i$ has the format of $\langle t_i, U_i \rangle$, in which $U_i$ is a set of Bluetooth MACs scanned at time $t_i$. In order to find the contact groups from our collected sensing traces, we first divide the time dimension into equal-sized



(a) Regularity threshold $\epsilon_c$.

(b) Time slot $\tau$.

**Fig. 9.** Regularity of contact.

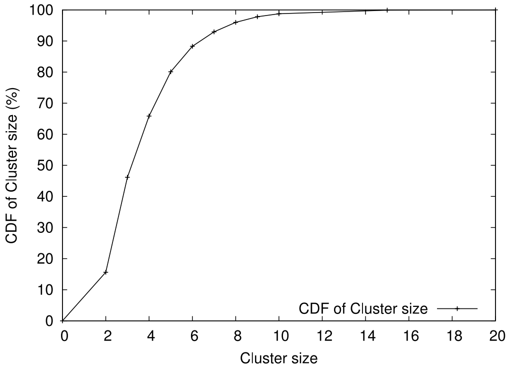


**Fig. 10.** Instant clusters (or instant contact groups).

time intervals of $\Psi$ (seconds). Within a particular time interval $\Psi$, the contact groups can be found as follows: given two Bluetooth scans $r_i = \langle t_i, U_i \rangle$ and $r_j = \langle t_j, U_j \rangle$ if $t_i \in \Psi$, $t_j \in \Psi$, and $U_i \cap U_j \neq \emptyset$, then Bluetooth MACs of $U_i$ and $U_j$ belong to the same contact group (or instant cluster). The intuition is that all Bluetooth MACs, that appear at the same location for the same time interval, are considered in the same instant cluster.

We then set $\Psi = 90$ (s) and find all clusters in Bluetooth traces of the dataset $D_1$ in Table 3. Here, we assume that the cluster remains unchanged during this 90 s time window. Fig. 10 shows that 90% of instant clusters (contact groups) have the size of 6 or smaller, 98% of instant clusters have the size of 10 or smaller, and the largest cluster has the size of 18. This means people rarely form large contact groups during their daily activities. Therefore, algorithms in multi-casting, content distribution, and congestion control in mobile peer-to-peer networks or DTNs [26–28] may need to take these results into consideration.

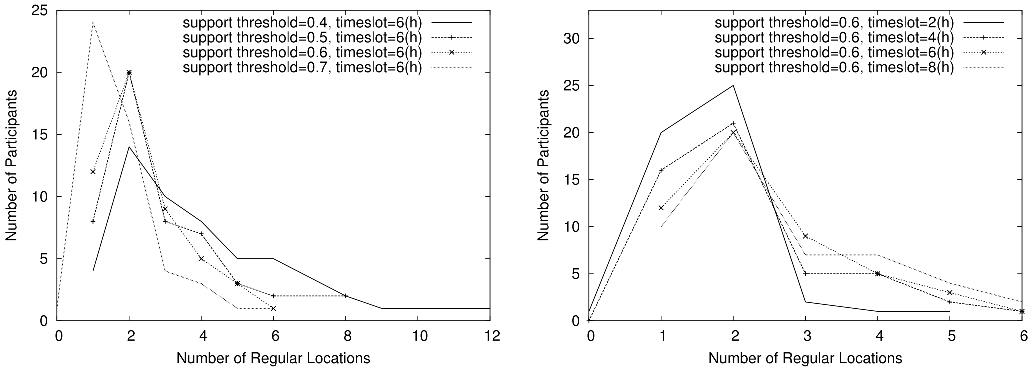
### 3.3. Location analysis

We define *location* as a unique set of WiFi MACs, which coexists **frequently** in the records of WiFi traces of all experiment phones. We apply UIM Clustering algorithm [19] to find locations from WiFi MACs. Then, we characterize regular location and location popularity.

#### 3.3.1. Regular location

Here, we seek answers for the question: do people visit locations regularly for their daily activities? For this analysis, we select a set of 50 experiment phones from three datasets $D_1, D_2, D_3$ in Table 3. Let $W_D$ be the WiFi traces of these 50 selected phones.

We divide the course of a day into time slots of size $\tau$ hours (i.e., $1 \leq \tau \leq 24$). For example, with $\tau = 6$ (h), we have following time slots ([00:00,06:00), [06:00,12:00), [12:00,18:00), [18:00,24:00]). For each phone (or participant) $p$, let $D_p$ be the number of days on which $p$ collects WiFi traces, and $L_p$ be set of locations the phone $p$ visits during the experiment period. For a location $l \in L_p$, let $D_{lp}$ be the number of days $p$ visits location $l$, in which these visits happen at the same time slot of these $D_{lp}$ days. For example, $p$ may visit location $l$ every weekday at 2PM. In our context, *a location $l \in L_p$ is a "regular location" of the phone $p$ if $D_{lp} \geq D_p \cdot \epsilon_l$ (i.e.,* $0 \leq \epsilon_l \leq 1$), in which $\epsilon_l$ is a pre-defined threshold, denoted as the "regularity



(a) Regularity threshold $\epsilon_l$.

(b) Time slot $\tau$.

**Fig. 11.** Regular location.

threshold". In other words, a location $l$ is considered a regular location if the phone $p$ visits $l$ at the same time slot for at least $D_p \cdot \epsilon_l$ days during the experiment period of $D_p$ days.

Next, we set $\tau = 6$ (h) and vary $\epsilon_l$ from 0.4 to 0.7. Fig. 11(a) shows that when $\epsilon_l$ increases, the number of participants with more regular locations decreases, which is intuitive. We also find that most people have at least two regular locations, which may be their home and work locations. Some participants have 12 regular locations, which indicates a highly repetitive movement behavior. With $\epsilon_l = 0.7$, 24 participants have one regular location, and the other participants have more than one regular locations.

To further understand the regularity of location visit, we set $\epsilon_l = 0.6$ and vary $\tau$ from 2 to 8 (h). Fig. 11(b) shows that for 2 (h) $\leq \tau \leq$ 8 (h), 10 to 20 participants have one regular location, 17–25 participants have two regular locations, others have more regular locations. Fig. 11(b) also shows that when $\tau$ increases, the number of participants with more regular locations increases. This is intuitive since it relaxes the definition of regular locations. In summary, our analysis shows that people usually visit regular locations during their daily activities.

### *3.3.2. Location popularity*

Popularity of location has been a focused research topic since it has a broad impact on numerous domains and applications such as urban planning, network resource scheduling, environmental control, location-based services [13,14].

Location popularity can be defined and interpreted differently. For example, given two locations $L_1$ and $L_2$, $L_1$ can be considered more popular than $L_2$ if more people appear at $L_1$ during *the same time window*. $L_1$ can also be considered more popular if more contacts are made at $L_1$ during *the same time window*. Further, $L_1$ can be considered more popular if more contact pairs are at $L_1$ during *the same time window*. Note that, more people appear at the location does not always mean more contacts are made since people may form small contact groups. Similarly, more contact pairs at the locations does not always mean more contacts at the locations. Here, we investigate how location popularity changes if we define location popularity based on: (1) number of people at locations, (2) number of contacts at locations, and (3) number of contact pairs at locations. In our context, a contact is made by a pair of devices (i.e., contact pair) and a contact pair may make many contacts at different times. For following plots, we use WiFi traces of the dataset $D_2$ in Table 3, which was collected by 79 participants for 38 days.

Fig. 12 shows location popularity in terms of number of experiment participants at locations. This figure shows that location popularity exhibits a heavy-tailed distribution. Especially, for location ranks greater than 30, location popularity follows a Zipf distribution.

Fig. 13(a) shows location popularity in terms of number of contacts made at locations. Again, location popularity is a heavy-tailed distribution. Specifically, only locations with ranks from 3 to 300 exhibits a Zipf distribution and the first two locations have a significant more number of contacts. Fig. 13(b) shows location popularity in terms of number of contact pairs at locations. Similar to these above results, location popularity exhibits a heavy-tailed distribution. However, only locations with ranks from 30 to 200 exhibit a Zipf distribution.

We also conduct two characterization studies on the impact of time on location popularity. For these studies, we use the definition: a location is more popular if it has more contacts. The first study investigates the change of location popularity between weekday and weekend. Fig. 14(a) shows that location popularity changes noticeably between weekday and weekend, and that popularity of locations decreases exponentially for weekdays (Note that we rank locations based on number of contacts during weekdays). Moreover, the most 100 popular locations during weekdays and 100 popular locations at weekends are not exactly the same. However, overlapping exists and several locations are popular both during weekdays and at weekends. These popular locations are university labs because many experiment participants are grad students, they usually work at labs both during weekdays and at weekends.

In the second study, we evaluate location popularity across time intervals of a day. We divide a day into 4 time intervals of {[12AM:6AM), [6AM–12PM), [12PM–6PM), [6PM–12AM)} and plot the last three time intervals (for a better presentation)

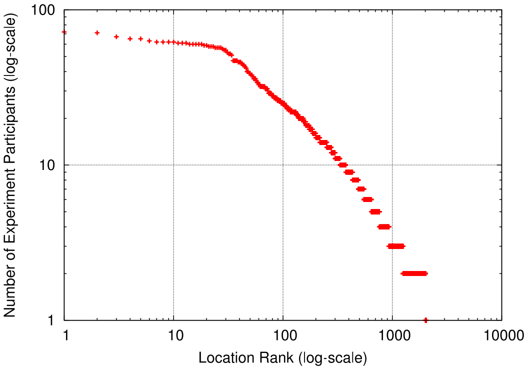
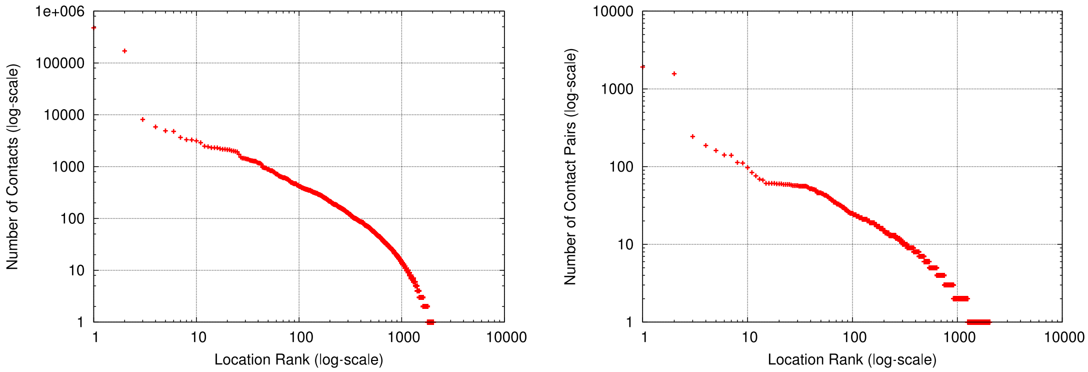


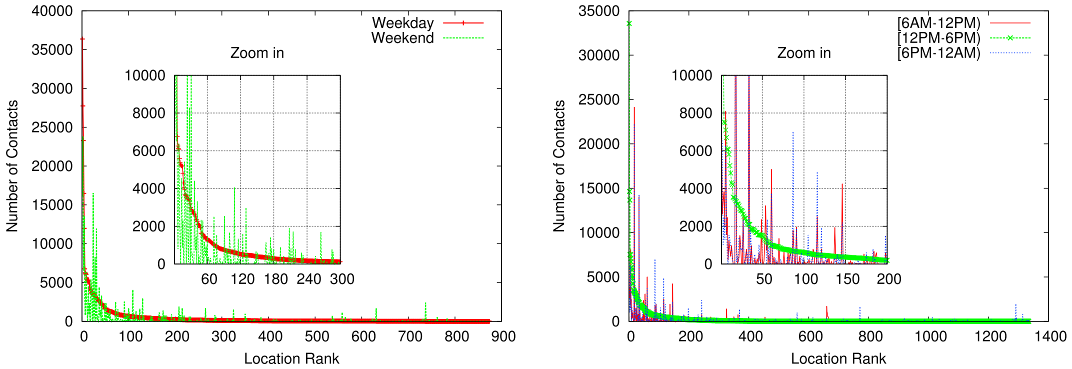
**Fig. 12.** More people at popular locations (log–log scale).



(a) More contacts at popular locations.

(b) More pairs at popular locations.

**Fig. 13.** Location popularity and contact (log–log scale).



(a) Weekday vs. weekend.

(b) Hour range.

**Fig. 14.** Location popularity changes across time slots or days of week.

in Fig. 14(b). We rank locations based on number of contacts during the [12PM–6PM) time interval since it is the most active time interval during a day of many people. Fig. 14(b) shows that location popularity changes across time intervals. Specifically, popularity of locations decreases exponentially for time interval [6PM–12PM) while other two time intervals have different location popularity patterns. In summary, our findings suggest that location popularity depends on: (1) how location popularity is defined, (2) at which time we measure the location popularity (e.g., weekday vs. weekend, or time intervals). These results on location popularity were not studied before and should be taken into account in future related studies.

## 4. Modeling people movement

In this section, we exploit joint location/contact traces to derive models for missing contact prediction and models for people movement prediction.

### 4.1. Context-aware predictive models of missing contacts

Crowd sensing has recently become an emerging research topic [29,30]. In crowd sensing settings, a set of sensing devices is carried by mobile carriers (i.e., human beings or machines) to collect (sense) contextual information such as location information and contacts. The objective of crowd sensing is to collect as much information as possible, given a limited number of sensing devices and a geographical area for sensing tasks. However, missing contacts always exist in collected sensing traces due to: (1) failure of scanning applications on sensing devices, (2) existence of physical obstacles that prevent the scanner to scan nearby devices, and (3) limited number of scanners in a highly crowded and mobile environment. Traces with missing contacts are less desirable since the models derived from these traces may not perform well. As a result, recovering and inferring missing contacts in the collected sensing traces become important.

Here, we propose a supervised machine-learning based model that combines location and contact contextual information to provide prediction of missing contacts. Our model incorporates new contact-based features that were not presented in previous studies [31] and novel location-based features into prediction of missing contacts. Given these features, we use Support Vector Machine (SVM) technique to train a supervised model. We evaluate our model with UIM traces and the evaluation results show that our proposed model outperforms the model that uses only contact-based features.

#### 4.1.1. Problem statement

For our crowd sensing settings, let $\Gamma_S$ be the set of sensing devices or scanners and $\Gamma_E$ be the set of non-sensing devices, so $\Gamma_S \cap \Gamma_E = \emptyset$. Let $\Gamma = \Gamma_S \cup \Gamma_E$. Given a scanner $u \in \Gamma_S$ and a device $v \in \Gamma_E$, we say $u$ and $v$ have a contact during a time interval $\tau$ if $v$ exists in scanning records of $u$ during the interval $\tau$. For a device $x$, either sensing or non-sensing, let $\Omega_\tau(x)$ be the set of contacts $x$ has during the interval $\tau$. If $x$ is a sensing device, $\Omega_\tau(x)$ can be obtained by aggregating all $x$'s sensed records for the interval $\tau$. If $x$ is a non-sensing device, $\Omega_\tau(x)$ includes sensing devices whose scanned records during the interval $\tau$ contain $x$. That is, if $u \in \Gamma_S,\ x \in \Gamma_E$, and $x \in \Omega_\tau(u)$, then we have $u \in \Omega_\tau(x)$.

For a pair of non-sensing devices $v,\ w \in \Gamma_E$, their contacts are not recorded in the sensing traces, *our objective is to infer contacts between $v$ and $w$ if these contacts exist*. These contacts are "missing contacts", which can be inferred from the observed contacts $\Omega_\tau(x)$ and location information of $x$ during the interval $\tau$. We formulate the missing contact prediction as a binary classification problem. That is, we derive a classification function for a pair of devices $v,\ w \in \Gamma_E$ and during a time interval $\tau$ so that $f(u, w, \tau) = \{1, 0\}$, where 1 means $v$ and $w$ have contacts during interval $\tau$, and 0 otherwise.

#### 4.1.2. Model construction

We construct a supervised model using SVM to represent the function $f(u, w, \tau)$. In order to construct the model, we use both location-based and contact-based features. We first include three features used by previous studies [31], such as: (1) "product of node degrees" of nodes (i.e., devices) in the graph formed by all devices $x \in \Gamma$, and an edge between $x, y \in \Gamma$ exists if $x$ and $y$ have contacts during $\tau$, (2) number of overlapped contacts between a pair of devices $x, y \in \Gamma$ for the interval $\tau$, and (3) duration (in minute) the two devices $x, y \in \Gamma$ have contacts with another device $z \in \Gamma$.

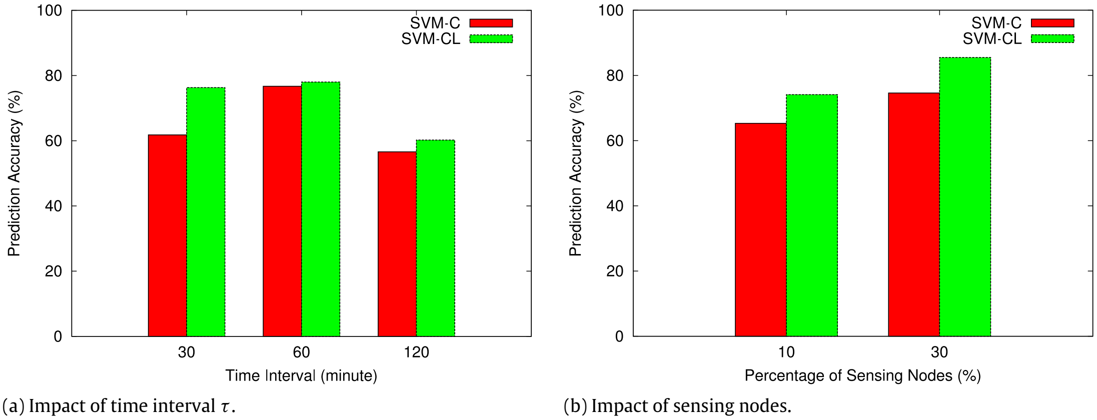
Besides, we proposed two new contact-based features and four location-based features. Contact-based features include: (1) probability that two devices have contacts during a weekday (i.e., Monday, Tuesday, etc.), and (2) probability that two devices have contacts during one of the four time slots of a day (i.e., each day is divided into four 6-hour time slots). Location-based features include: (1) location popularity ranks of locations where two devices appear together during interval $\tau$, (2) probability that two devices appear together at a particular location, (3) density of devices in the nearby area of two devices, and (4) number of locations the two devices appear together.

Given the set of above features, we train our SVM model with Weka machine learning toolkit. Next, we discuss the evaluation results of our model.

#### 4.1.3. Model evaluation

*Experiment settings.* We use the traces in the dataset $D_1$ in Table 3 for our evaluation. These traces were collected by 28 participants during a 19 day period. One of the main challenges for performance evaluation of predictive models in crowd sensing research is obtaining ground truth to compare against. To overcome this, we create a set of nodes that consists of 28 experiment phones in $D_1$ and use this set as $\Gamma$ in Section 4.1.1. Since these 28 nodes are our experiment phones, we can get their contacts from collected traces. For our evaluation, we randomly select a subset $\Gamma_S$ of sensing nodes from $\Gamma$ to build the SVM-based predictive models. So, we have $\Gamma_E = \Gamma \setminus \Gamma_S$ of non-sensing nodes. The objective is to infer contacts of nodes in $\Gamma_E$ (which we know from our collected traces) by using contacts of nodes in $\Gamma_S$.

We implement two models SVM-C and SVM-CL. SVM-C is constructed based on contact-based features while SVM-CL is constructed based on location-based and contact-based features in Section 4.1.2. We then compare the prediction accuracy of SVM-C and SVM-CL with respect to the two metrics: length of time interval $\tau$ and percentage of sensing nodes (i.e., size



(a) Impact of time interval $\tau$.

(b) Impact of sensing nodes.

**Fig. 15.** SVM-CL consistently outperforms SVM-C.

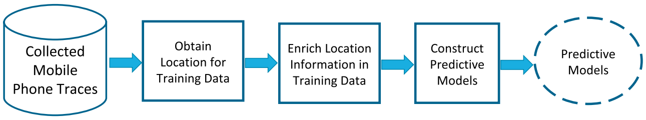

Collected Mobile Phone Traces
Obtain Location for Training Data
Enrich Location Information in Training Data
Construct Predictive Models
Predictive Models

**Fig. 16.** Predictive framework.

of $\Gamma_S$). On one hand, we need to understand how the two models perform under different values of $\tau$ since $\tau$ is a crucial control knob that impacts the topology of the contact graphs and the location popularity. On the other hand, reducing the number of sensing nodes while achieving acceptable prediction accuracy is desirable.

*Results*. Fig. 15(a) shows that the two models are sensitive to the value of $\tau$ and they both perform best with $\tau = 60$ (m) but worst with $\tau = 120$ (m). However, SVM-CL consistently outperforms SVM-C for different time intervals and SVM-CL performs much better than SVM-C with $\tau = 30$ (m). The time interval $\tau$ impacts the topology of contact graph and thus it has a direct impact on prediction outcomes. So, finding the optimal interval $\tau$ will be one of our future works. Fig. 15(b) shows that when the number of sensing nodes increases, both schemes perform better, which is intuitive since when we have more sensing nodes, we cover more contacts. However, SVM-CL always outperforms SVM-C. For example, with 30% of sensing nodes (or 8 participants), SVM-CL obtains 85.5% of accurate predictions while SVM-C only obtains 74% of accurate predictions.

In summary, we proposed a novel model that combines contact-based and location-based features and provides accurate missing contact prediction. We have demonstrated that by utilizing both location and contact information, one can improve the performance of the predictive models.

### 4.2. *Context-aware predictive models of people movement*

There have been previous research projects toward predictions of people movement since knowledge of people movement is vital to numerous domains and applications [32,33,19]. Our study in Section 3 shows that people movement exhibits repetitive patterns, i.e., people usually visit regular locations and make regular contacts. Here, we present a methodology that exploits these regular patterns to construct models of people movement. The constructed models are able to provide answers to fundamental questions: (1) where will the person be in the future (location)?, (2) how long will he stay at the location (stay duration)?, and (3) who will he meet at the location (social contact)?

#### 4.2.1. *Methodology*

Fig. 16 shows steps to construct the predictive models of people movement from mobile phone sensing traces, as we present in detail below.

First, we need to define the notion of "location". This definition is crucial since it decides the traces for inferring location information. The inferred locations could provide knowledge of the movement characteristics, which is fundamental for construction of predictive models. Note that granularity of location may depend on the target applications. For example, some applications require building-level locations while others need room-level locations. WiFi traces can provide room-level locations while cellular base station identity may provide coarser location. GPS signal can provide more precise and finer grained locations, however it is not available with indoor scanning devices. So, in this first step, we need to select the dataset that provides most accurate location from all collected datasets. Then, we have to assign location for all records in the selected dataset. Let $W$ be output dataset with assigned locations.

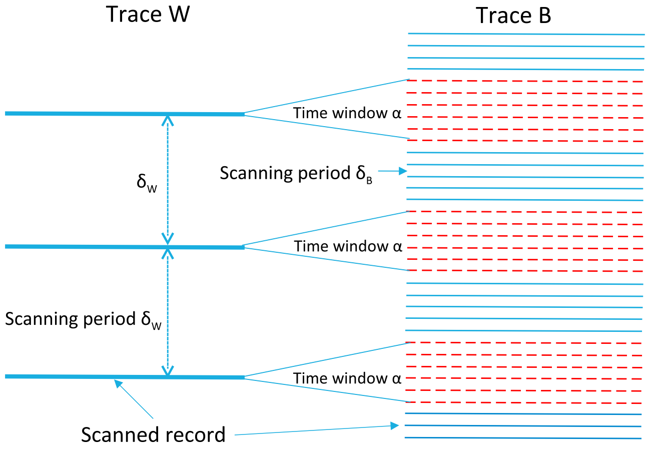


**Fig. 17.** Enrich location.

Second, we use $W$ to enrich location information for other collected datasets as shown in Fig. 17. This is a crucial step since its output provides a richer contextual data for better predictive models. Section 2.1 shows that each phone sensor collects different contextual information and consumes different amount of energy. Among the implemented sensors for sensing tasks, some may have shorter scanning period than others and thus collects more data. In our UIM system, the WiFi scanner scans every $\delta_W = 30$ min while the Bluetooth scanner scans every $\delta_B = 1$ min. More importantly, while WiFi traces can be used to infer location, Bluetooth traces themselves cannot be. So, this step is to label records of Bluetooth traces $B$ with locations obtained in WiFi traces $W$. The assumption is that if a person stays at location $L$ at time $t$, then he is likely to stay at that location during the time window $[t - \frac{\alpha}{2}, t + \frac{\alpha}{2}]$. Since scanned timestamps of Bluetooth and WiFi records are synced, we can use this technique to assign locations to all records of $B$ that belong to the time window $\alpha$ of an individual record of $W$. The records of $B$ with assigned locations form a set $B'$ where $B' \subset B$ (i.e., the dashed lines in Fig. 17). Then, we can construct a supervised model using a technique such as SVM, Naive Bayesian, and k-NN on $B'$. The trained model is used to predict locations for other records of $B$ (i.e., the solid lines in Fig. 17). Let $B''$ be the output dataset.

Third, given $B''$ dataset we use a supervised machine learning technique such as SVM, Naive Bayesian, k-NN to construct predictive models of people movement. Note that since $B''$ is enriched, it can provide better predictive models. For example, for our UIM system, $B''$ contains records with locations and records are every minute apart. $B''$ thus can be used to provide future information about location, stay duration, and social contact.

### 4.2.2. Jyotish

We apply the above methodology and implement a system named Jyotish to construct predictive models of people movement [19,15] using our collected Bluetooth $B$ and WiFi $W$ datasets. Our models provide answers to three questions: (1) where the person will stay in the future, (2) How long he will stay at the location, and (3) Who he will meet at the location.

We design the UIM clustering technique [15] to cluster WiFi MACs in $W$ into locations. Given the dataset $W$ with locations, we then use the time window $\alpha = 90$ (s) to create a set $B'$ of records with assigned locations. Specifically, we trained a supervised model using Naive Bayesian technique on $B'$ to label locations for other records of $B$, and obtain the enriched dataset $B''$. Given the enriched dataset $B''$, we trained three supervised machine learning based predictive models using Naive Bayesian technique, including location predictor, stay duration predictor, and contact predictor. We then evaluate these predictors with three datasets $D_1, D_2, D_3$ in Table 3. The experiment results show that our predictors perform well and provide accurate prediction on location, stay duration, and social contacts. Details of Jyotish design and implementation can be found in our previous works [19,15].

## 5. Conclusion

This paper presents methodologies and implementation about capturing sensing traces via mobile phones, measuring characteristics found in collected traces, and predicting missing contacts and future contextual information of people movement. Using selected case studies, such as (1) the UIM scanning system that captured WiFi and Bluetooth traces, (2) the extensive characterization study that measured fundamental aspects of people movement, and (3) the predictive model that predicted missing contacts and prediction framework that predicted location, stay duration and social contacts, we exemplified details of feasible solutions for characterizing and modeling people movements. From our work, we draw following lessons:

1. Traces of location sensor and at least one social contextual sensor (e.g., Bluetooth scan, photo capture, audio recording, etc.) are needed to derive mobility and density of people movement. So, a good scanning system must include at least

two scanners since collection of location information solely may not provide sufficient input for characterizing people movement.

2. Broken traces are major issues when analyzing traces and one gets false statements about mobility and people movement if certain characteristics (i.e., regular contacts, regular location visits, etc.) of sensing traces are not preserved. More importantly, traces are only usable if it is collected for a long enough period. Implementing and configuring the scanning system so that it can run in a reliable and robust manner for a prolonged trace collection task is critical.
3. Training datasets are crucial for prediction models of people movement. Enriching training data, by (1) inferring missing contacts, and (2) assigning location information for the traces collected by sensors with shorter scanning periods, is important since it provides better quality training datasets, which improves the final predictive models.

As phones are getting equipped with more and more sensors, ad hoc collaborations among groups of phones enable new interactions, and privacy-preserving schemes become integral parts of participatory sensing, we will experience an explosion of capturing and measuring of individual and group-based sensory traces, enabling even higher accuracy of characterizing and modeling people movement. We thus believe our presented methodologies and implementation in this paper will become useful and relevant.

## Acknowledgments

This research was supported by the National Foundation Grant NSF CISE CNS 1346782, the German Research Society (DFG) SFB 1053 MAKI Grant, and by the Deutsche Telecom Gift Grant. All results, findings and claims belong to the authors and do not reflect the opinions of the funding organizations.

## References

[1] L. Vu, K. Nahrstedt, M. Hollick, Exploiting schelling behavior for improving data accessibility in mobile peer-to-peer networks, in: Proceedings of Mobiquitous, 2008.
[2] F. Asgari, V. Gauthier, M. Becker, A survey on human mobility and its applications, 2013. Available: arXiv:1307.0814 [Online].
[3] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Impact of human mobility on the design of opportunistic forwarding algorithms, in: Proceedings of INFOCOM, 2006.
[4] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M.D. de Amorim, J. Whitbeck, The accordion phenomenon: analysis, characterization, and impact on DTN routing, in: Proceedings of INFOCOM, 2010.
[5] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, J. Laurila, Towards rich mobile phone datasets: Lausanne data collection campaign, in: Proceedings of ICPS, 2010.
[6] T.M.T. Do, D. Gatica-Pereza, Where and what: using smartphones to predict next locations and applications in daily life, Pervasive Mob. Comput. J. 12 (2014) 79–91.
[7] A. Sadilek, J. Krumm, Far out: predicting long-term human mobility, in: Proceedings of AAAI, 2012.
[8] W. Gao, G. Gao, Fine-grained mobility characterization: steady and transient state behaviors, in: Proceedings of MobiHoc, 2010.
[9] W. Hsu, T. Spyropoulos, K. Psounis, A. Helmy, Modeling time-variant user mobility in wireless mobile networks, in: Proceedings of INFOCOM, 2007.
[10] N.E. William, T. Thomas, M. Dunbar, N. Eagle, A. Dobra, Measurement of human mobility using cell phone data: developing big data for demographic science, in: Population Association of America Annual Meeting, 2013.
[11] L. McNamara, C. Mascolo, L. Capra, Media sharing based on colocation prediction in urban transport, in: Proceedings of MobiCom, 2008.
[12] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, A. Campbell, NextPlace: a spatio-temporal prediction framework for pervasive systems, in: Proceedings of Ninth International Conference on Pervasive Computing, 2011.
[13] T.M.T. Do, D. Gatica-Pereza, The places of our lives: visiting patterns and automatic labeling from longitudinal smartphone data, IEEE Trans. Mob. Comput. 13 (2014).
[14] L. Vu, K. Nahrstedt, S. Retika, I. Gupta, Joint Bluetooth/Wifi scanning framework for characterizing and leveraging people movement in university campus, in: Proceedings of MSWiM, 2010.
[15] L. Vu, Q. Do, K. Nahrstedt, Jyotish: a novel framework for constructing prediction model of people movement from joint Wifi/Bluetooth trace, in: Proceedings of PerCom, 2011.
[16] Y. Chon, E. Talipov, H. Shin, H. Cha, Mobility prediction-based smartphone energy optimization for everyday location monitoring, in: Proceedings of SenSys, 2011, pp. 82–95.
[17] E. Miluzzo, N.D. Lane, S.B. Eisenman, A.T. Campbell, CenceMe—injecting sensing presence into social networking applications, in: Proceedings of Second European Conference on Smart Sensing and Context, EuroSSC, 2007.
[18] N. Eagle, A. (Sandy), Reality mining: sensing complex social systems, Pers. Ubiquitous Comput. 10 (2006) 255–268.
[19] L. Vu, Q. Do, K. Nahrstedt, Jyotish: constructive approach for context predictions of people movement from joint Wifi/Bluetooth trace, Pervasive Mob. Comput. J. 6 (2011) 690–704.
[20] E.P.S. Gaito, G.P. Rossi, Opportunistic forwarding in workplaces, in: Proceedings of ACM WOSN, 2009.
[21] J. Leguay, A. Lindgren, J. Scott, T. Friedman, J. Crowcroft, Opportunistic content distribution in an urban setting, in: Proceedings of CHANTS, 2006.
[22] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, C. Diot, Pocket switched networks and human mobility in conference environments, in: Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking, 2005.
[23] J. Su, A. Chin, A. Popivanova, A. Goel, E. de Lara, User mobility for opportunistic ad-hoc networking, in: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications, 2004.
[24] Y.-C. Cheng, CRAWDAD trace ucsd/cse/jigsaw/wireless, 2008.
[25] Android development. http://developer.android.com/.
[26] P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: social-based forwarding in delay tolerant networks, in: Proceedings of MobiHoc, 2008.
[27] L. Vu, K. Nahrstedt, I. Rimac, V. Hilt, M. Hofmann, ishare: Exploiting opportunistic ad hoc connections for improving data download of cellular users, in: Proceedings of IEEE Globecom Workshops, 2010.
[28] L. Vu, Q. Do, K. Nahrstedt, 3R: Fine-grained encounter-based routing in Delay Tolerant Networks, in: Proceedings of IEEE WoWMoM, 2011.
[29] R. Kravets, H. Alkaff, A. Campbell, K. Karahalios, K. Nahrstedt, CrowdWatch: enabling in-network crowd-sourcing, in: ACM SIGCOMM Workshop on Mobile Cloud Computing, 2013.
[30] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, G.-S. Ahn, The rise of people-centric sensing, IEEE Internet Comput. 12 (2008).
[31] K. Jahanbakhsh, V. King, G.C. Shoja, Predicting missing contacts in mobile social networks, Pervasive Mob. Comput. 8 (2012) 698–716.
[32] Y. Chon, H. Shin, E. Talipov, H. Cha, Evaluating mobility models for temporal prediction with high-granularity mobility data, in: Proceedings of PerCom, 2012, pp. 206–212.
[33] J. Weppner, P. Lukowicz, Bluetooth based collaborative crowd density estimation with mobile phones, in: Proceedings of PerCom, 2013, pp. 193–200.