SocialProbe: Understanding Social Interaction Through Passive WiFi Monitoring

Hande Hong[†], Chengwen Luo[‡], Mun Choon Chan[†]
[†] School of Computing, National University of Singapore
[‡]College of Computer Science and Software Engineering, Shenzhen University
[†]{honghand,chanmc}@comp.nus.edu.sg, [‡]chengwen@szu.edu.cn

ABSTRACT

In this paper, we present an approach to extract social behavior and interaction patterns of mobile users by passively monitoring WiFi probe requests and null data frames that are sent by smartphones for network control/management purposes. By analyzing the temporal and spatial correlations of the Receive Signal Strength Indicators (RSSI) of packets from these low rate transmissions, we are able to discover proximity relationships, occupancy patterns, and social interactions among users.

We evaluate the SocialProbe system using commodity off-the-shelf smartphones and WiFi Access Points in two locations, a research lab and a public dining area. The result shows that the proposed approach is able to obtain reliable social relationships and interactions in a non-intrusive way.

CCS Concepts

•Networks → Wireless access points, base stations and infrastructure; •Human-centered computing → Social network analysis; Ubiquitous and mobile computing systems and tools; Smartphones;

Keywords

Social Relationship; Passive; WiFi Probe; Fingerprinting

1. INTRODUCTION

Understanding social behavior and interaction pattern plays an important role in many disciplines. Having this information is useful in a diverse range of applications, including psychology [22, 12], health care [23, 28] and urban planning [31, 8]. However, obtaining such information often either requires additional hardware support or users need to install software on their mobile devices. Getting this information in a non-invasive manner has been a long-standing challenge in the research community.

As the capability of mobile devices continues to grow, mobile devices, especially smartphones, will be even more ubiquitous. A recent study [10] suggests that negative psychological and physiological outcomes are associated with phone separation even just for a while. Thus, smartphone provides a good proxy to reason about a user's behavior and social interaction pattern.

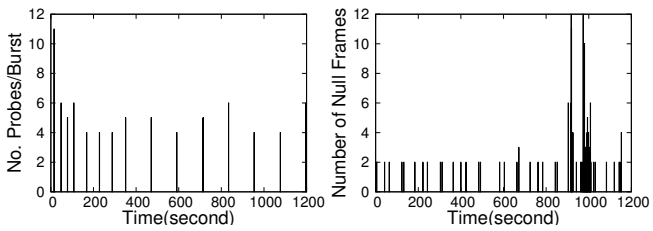
Due to the widespread deployment of WiFi networks and the availability of WiFi chipsets on smartphones, use of WiFi related information to extract context information has been both popular and shown to be effective. Researchers have proposed various ideas to utilize WiFi based information including indoor localization [3, 33], crowd counting [17, 30] and passive tracking [24, 25]. One limitation of these techniques is that they require users to install applications on their smartphones, which can be cumbersome and limits deployment. To make matter worse, Apple has recently removed the API for WiFi scanning on iOS platform [13] thus severely limiting the use of mobile apps that require active WiFi scanning on iOS.

In order to overcome such problems, researchers have proposed to use WiFi probe request frames. These probe request frames are constantly broadcast by smartphones to advertise their presence to nearby access points (APs). These frames contain useful information such as network identifier (SSID), MAC address, signal strength, and time stamp. Such approaches are *passive* because they require no change on the mobile devices. The monitoring is performed only by the APs with no impact on the operations of existing infrastructure. In the past, researchers have been able to uncover many interesting social relationships by looking at the SSIDs contained in the probe request [11, 14, 4, 7]. However, in a recent study [18], over 80% of the probe requests are broadcast with empty SSID field to reduce information leakage. This implies that such SSID based approaches will be much less effective.

In this work, we present an approach to extract social behavior and interaction pattern by using the spatial-temporal correlation of 802.11 frames and the signal strength information contained in these frames. Signal strength information is typically under-utilized because of the noise and non-uniform (among different phones) behaviors. In spite of these challenges, we show that we are able to discover users' daily behavior and social interaction pattern by passive WiFi monitoring. We summarize our contributions as follows:

- We make use of both probe request frames and null data frames transmitted by WiFi clients. While probe request frames have often been exploited, we are not aware of previous work that has utilized null data frames for relationship detection.
- Frames transmitted by different phones even in the same location have significantly different RSSIs. We compare different signal fingerprints normalization methods and use the maximum scheme to detect if two mobile devices are co-located.
- We design and evaluate SocialProbe, a system that can dis-

MOBIQUITOUS '16, November 28-December 01, 2016, Hiroshima, Japan
 ISBN 978-1-4503-4750-1/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2994374.2994387



(a) Probe request frames (b) Null data frames

Figure 1: The number of frames sent over time observed for a Nexus 5. The phone is not associated to an AP in (a) while connected to an AP in (b).

cover social interaction among users of mobile devices. We demonstrate the utility of SocialProbe by showing how it can be used to discover the underlying relationship and interaction patterns among different users in two different environments, namely a research lab and a public dining hall.

The rest of the paper is organized as follows. In section 2, we investigate how probe request and null data frame intervals and signal strength differ among different phones. In section 3 and 4, we present an overview of the system and introduce the four components of SocialProbe: devices filtering, fingerprint generation and normalization, co-location detection, and social relationship discovery. We show the evaluation result in section 5 and present related work in section 6. Finally, we discuss and summarize our work in section 7 and 8.

2. FEASIBILITY AND CHALLENGE

2.1 Probe Request and Null Data Frame

Probe Request Frame: In order to speed up the discovery of surrounding APs, a smartphone broadcasts probe request frames to trigger responses from nearby APs. These frames are management frames containing information such as network identifier (SSID), MAC address, signal strength, and time stamp. This information is used by devices and APs to make decisions related to association and authentication.

As probe requests frames are used for AP discovery, we expect these frames to be broadcast only when the smartphones are not associated with any AP. To our surprise, the mobile devices continue to broadcast probe requests even after they are already connected/associated to a WiFi network. We believe that this is done to prepare for handover if the WiFi signal from currently connected access point becomes weak, say due to mobility.

Figure 1(a) shows an example of how frequent a device, in this case, a Nexus 5 smartphone, sends out probe request frames. In this example, probe requests are sent in a burst of a few frames (4 to 11) each time. Probe interval increases from 30 seconds to 60s second and finally reaches a stable value of 120 seconds. The longest interval occurs when the smartphone cannot find available WiFi AP to connect to after repeated attempts. This behavior strikes a balance between energy consumption and discovery time.

Null Data Frame: Many existing works on WiFi passive monitoring rely only on probe request frames. In our work on discovering social interaction, it is reasonable to assume that the devices have access to the WiFi infrastructure. This is typically true in office and campus environments and is also likely in places such as malls, public libraries and transportation hubs (e.g airports and train stations).

Thus, in order to increase the amount of information we can gather, we also monitor additional IEEE 802.11 control frames. One control frame of interest is the null data frame. Null data frames are transmitted when a device is associated. It is used for power management purpose, such as to inform the AP that the device will be going offline into power save mode (PSM) or when it is online and ready to receive data.

Figure 1(b) shows an example of how frequent a device, in this case, a Nexus 5 smartphone, sends out null data frames. Null frames are sent every 20-30 second, with two frames sent each time. From time to time, the device will send out a burst of null frames when it hit the roaming threshold and is seeking out another access point to associate with. In Figure 1(b), we can also observe a large number of null frames transmitted around the 950 to 1000 second interval.

We found a similar pattern on 5 different smartphone phone models. The maximum interval between probes varies between 45 seconds to 5 minutes. The result we get from this controlled experiments is consistent with the findings by other researchers [16, 18, 24].

2.2 Inter-Frame Interval in the Wild

By combining probe request and null data frames, one can get information on a device when it is associated and not associated. However, the question of what are the (probe and null data) frame transmission intervals in the wild remains. To answer this question, we analyze data collected from a university environment covering eating areas, classroom, and research labs over a two months period. The data contains information from 4574 devices and 33 phone vendors. We show the results in Figure 2. For probe request frames, 89% of the inter-frame intervals are 120 seconds or less and 60% of the frames are separated by 20 seconds or less. On the other hand, 70% of the null frames are sent at intervals of 20 seconds or less. 30% of null frames are sent within 5 seconds which are the result of those burst of null frames shown in Figure 1(b) in the 950-1000 second interval. When we combine data from both probe request and null data frames, 95% of the frame intervals are less than 2 min.

Our measurement provides two valuable information. First, probe request and null data frames can provide up to minute-level granularity on user movement. Second, since 95% of probes and 97% of null frames have inter-frame intervals of less than 5 minutes, 5 minutes is a good threshold to decide whether a people has moved away based on frame reception.

2.3 Received Signal Strength

The signal strength information available in the probe request and the null data frame is known to be noisy and can be unreliable [6]. Many factors influence the stability of signal strength, including multi-path effect, antenna gain and phone placement. A phone can also transmit at different power depending on the specific IEEE 802.11 version used [15]. For example, Samsung Galaxy S4 sends at 13 dB using 802.11a but it sends at 12 dB using 802.11n.

In order to study the signal difference and stability from diverse phone models, we place six different phones very close together and collect probe requests from five WiFi monitors in a research lab. 5 different models installed with different versions of the Android operating system are used. We collected about 100 probe requests burst for each phone in all the 15 locations we have selected.

Figure 3 shows the results for one of the locations. From the figure, we see that even if the phones are placed close together, there can be substantial differences in the signal strength. While most of the signal strengths received from the same phone fluctuate within 4 dB, in some cases, the variation can be as much as 10 dB. Given such noisy and highly varying signals from different phones placed very close together, the challenge is thus on how to make sense of these noisy signals.

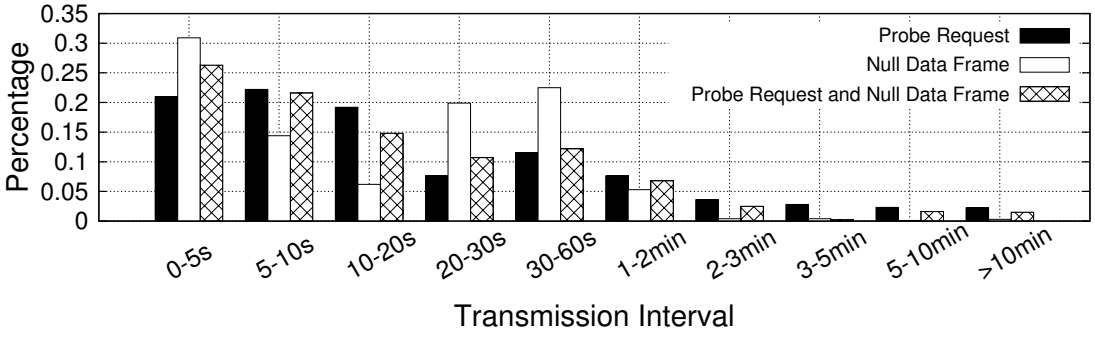


Figure 2: Probe request and null data frames interval distribution from 4574 devices over two months. 73% of the frames are sent by devices within 30 seconds interval and 95% within 2 minutes

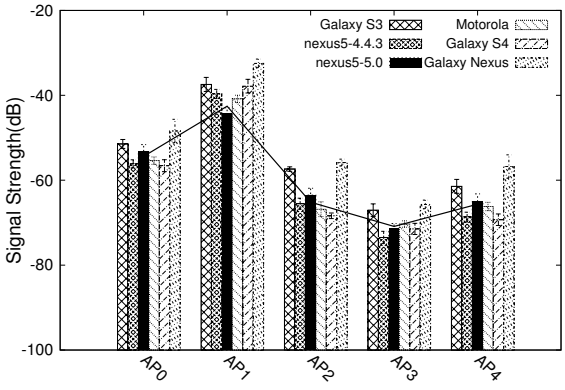


Figure 3: Signal strength of frames collected by 5 different WiFi monitors from 6 phones. Phones are kept in same location and settings: screen is ON but device is not connected to the WiFi.

3. SYSTEM OVERVIEW - SocialProbe

The overview of the proposed system, *SocialProbe*, is shown in Figure 4. Multiple WiFi monitors are deployed around the area of interest and each WiFi monitor scans for probe request and null data frames. When a user, carrying WiFi enabled mobile devices, enters the vicinity of the area to be monitored, the frames transmitted are captured by the monitors.

Data collected by the monitors are sent to the server for further processing. The server performs analysis of the data as follow:

- **Device Filtering:** only devices with valid MAC addresses and human-like present ratio are used.
- **Fingerprint Generation and Normalization** probe request and null data frames from multiple monitors are merged and then “normalized" to form the signal fingerprint.
- **Co-location Detection:** The above “normalized" WiFi fingerprints are fed into our algorithm to determine if two mobile devices are nearby.
- **Discovering Social Relationship:** Information on pair-wise device proximity enables us to discover short-term behavior like dining together in the canteen or a short gossip at tea breaks. As data was accumulated for a sufficiently long duration, social interaction pattern and the underlying relationship among mobile users can be obtained.

It is important to note that since we are relying on WiFi frames that are transmitted relatively infrequently, it is not possible to perform short time-scale tracking. Instead, we are looking at behaviors that last for a sufficiently long period, usually in the order of minutes.

4. METHODOLOGY

In this section, we will introduce each component of the SocialProbe system in detail.

4.1 Device Filtering

The first level of filtering is to make sure that the frames recorded belong to devices carried by human users. The processing is done as follows.

4.1.1 Filtering based on MAC Address

This filtering step is to make sure that the devices detected are from valid mobile device vendors. Note that we are mainly interested in smartphones carried by mobile users. Each frame received contains a plain text MAC address which contains a 3 byte Organizationally Unique Identifier (OUI) and a 3 byte Network Interface Controller Specific (NIC). By matching the first three bytes (OUI) to an online public database, we are able to identify vendors of the devices. This filtering process removes two categories of MAC addresses. The first category contains MAC addresses that are invalid or randomly generated. The second category are addresses that are not likely to be belonging to smartphones. Examples are laptops or devices that use WiFi dongles.

4.1.2 Filtering based on Present Ratio

While we are able to identify mobile devices that are likely to be smartphones, it is also important that these devices are carried by human users. Devices such as desktop or WiFi enabled sensor/gateways from smartphone vendors cannot be removed by MAC addresses alone. The next filtering step is based on temporal behavior. In particular, mobile devices carried by human users should exhibit daily routines similar to a human. After sufficient data is collected, we can use the data to check for consistency with human mobility patterns. We exclude devices with the two following patterns: very limited time or no mobility.

To perform this filtering step, we do an analysis of the detection duration and transmission frequency of these devices. We used a time slot duration of 5 min, a threshold determined based on the measurements from section 2. If consecutive frames from a device are detected within 5 min, we assume that the device is still in the vicinity of the monitor. Note that these observations are made after merging frames collected from all the monitors. Hence, as long as one of the monitors hears a probe request or the null data frame, the device is detected. Figure 5 shows a sample behavior of 4 devices detected over 5 days.

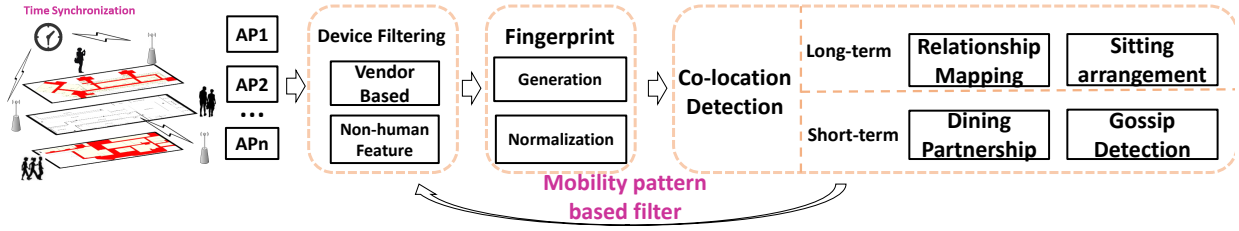


Figure 4: Overview of SocialProbe social interaction detection system

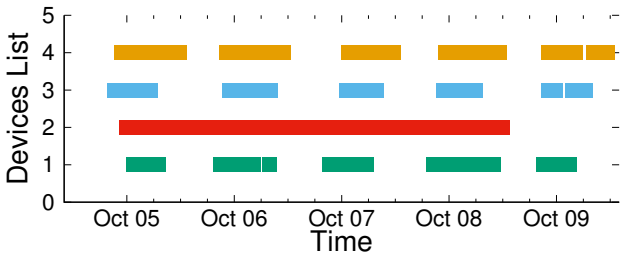


Figure 5: Active span for devices during weekday. Device No. 1, 3 and 4 show regular human activity patterns.

We calculate the Present Ratio (PR) of a device by dividing the active span (duration in which a device's frames can be detected in a 5 min interval) over the entire period monitored. If PR is less than T_{low}, we tend to think that the device is carried by humans who just walk past the monitored area or devices in deep sleep mode to conserve power. In both cases, either the stay time is too short or the frame transmission interval is too long for meaningful interpretation. If PR of a device exceed threshold T_{high}, then we tend to filter such devices as stationary machine not carried by the human. We empirically set T_{high} as 0.5, T_{low} as 0.01. For example, device 2 is a stationary WiFi-enable laptop left on the desk for 4 days which has high PR value of 0.8.

We used both OUI and PR methods to determine if a device is carried by a human user.

4.2 Fingerprint Generation and Normalization

4.2.1 *Fingerprint Generation*

After device filtering, the next step is to merge data that are collected from different monitors into a single signal fingerprint. The signal fingerprint of a device at a particular time is the collection of signal strengths extracted from frames collected at that time from different monitors. The fingerprint is represented as $\vec{f}$:$\{r_1, r_2, r_3, ..., r_n\}$, where r_i is the RSS captured by monitor i.

Note that frame detection is highly unreliable. It has been observed that about 40% of the frames transmitted may be missing [16]. In order to mitigate the effect of such losses, we search for fingerprints from nearby time periods to fill in the gaps. However, in order to reduce the error, we will fill in these measurement gaps only if the fingerprints before and after the gap do not differ too much. If we cannot find valid data from the nearby time periods, we use the value of -99 to denote missing data. To ensure that the local clocks are in sync, we run NTP in all the monitors.

4.2.2 *Fingerprint Normalization*

We have illustrates using Figure 3 the problem with utilizing RSS from different phone models. To compensate for such variations, we normalize the signal fingerprints obtained from monitors. Consider the Log-distance Path Loss (LDPL) model as shown in Equation (1):

$$RSS_d = RSS_0 - 10\gamma log_{10}(\frac{d}{d_0}) + \varepsilon \quad (1)$$

where RSS_d is the measured RSS value of a mobile device at a distance of d away. RSS_0 is the reference RSS at a distance of d_0. γ is the path loss exponent and ε is a Gaussian random variable with unknown parameters. Consider two co-located devices d meters from a monitor. If we compare the signal strength based on the LDPL model, we see that the term $10\gamma log_{10}(\frac{d}{d_0})$ relates to the transmission loss in the path and does not contain variable related to sending power. This term depends only on the distance d and the path loss component which should be similar for two devices in proximity. Hence, if we are able to normalize the fingerprint vector by reducing the effect due to differences in transmission power and emphasize the effect from distance path loss, the fingerprints of these two devices can be made more similar. We use the strongest signal to substitute RSS_0 in the LDPL model. That is to subtract the strongest signal strength from all signal strength and the new fingerprint vector becomes $\{r_1 - r_{max}, r_2 - r_{max}, ..., r_i - r_{max}, ..., r_n - r_{max}\}$. We compare the effectiveness of this normalization scheme in section 5 with using average and weakest signal strength to normalize fingerprint.

4.3 Co-location Detection

4.3.1 *Fingerprint Similarity*

To discover social interactions, an important step is to detect co-location. The key intuition we exploit is that we are not looking for localization but co-location detection. In co-location detection, we only need to know whether the devices are in proximity and do not need to know the exact location. Therefore, we want to know if two mobile devices and implicitly the persons carrying these devices are in proximity.

$$S_{ij} = \frac{\vec{f}_i \cdot \vec{f}_j}{\left\|\vec{f}_i\right\|^2 + \left\|\vec{f}_j\right\|^2 - \vec{f}_i \cdot \vec{f}_j} \quad (2)$$

To identify whether two devices are in the same place or in different places, one can compute the fingerprint similarity metric using the Tanimoto Coefficient as shown in Equation 2. This metric has also been used by many existing work [9, 20, 8] for co-location detection.

4.3.2 *Co-location Event Detection*

Fingerprint similarity gives us a metric to measure how close two devices, or two people, are. We can use this tool to detect co-location events, e.g. conversations, encounters, dining and so on. Figure 6 gives us a typical scenario where two people, Alice

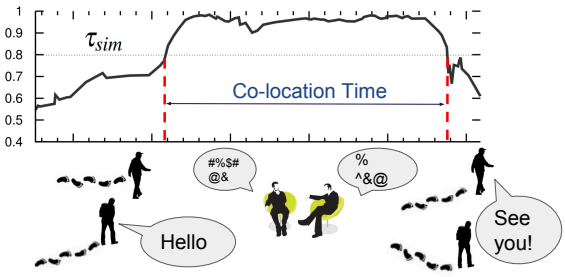


Figure 6: Fingerprint similarity trend during a conversation

and Bob (both carrying smartphones with them), met and had a short conversation. Initially, Alice and Bob are 10 meters apart. The fingerprint similarity is low, around 0.55 in the figure. As they approach each other closer, the fingerprints similarity of these two devices carried increases. When they sat down and talked, the similarity increases to values between 0.9 and 1.0.

This example motivates the use of fingerprint similarity to detect social interaction. However, in order to reconstruct such interactions much accurately, we need to derive the similarity threshold τ_{sim} that indicates sufficient proximity. The similarity threshold τ_{sim} to be used depends on the granularity of social events to be detected and the density of monitors deployed. For example, finding people dining together will require proximity detection of say 3 meters while people attending a lecture in the same classroom may be up to 10 meters away.

4.4 Discovering Social Relationship

Using fingerprint similarity over time enables us to identify co-location events and to estimate the intimate level between two people. The intuition is that if two people stay in the same vicinity, they are likely to interact or chat. Sustain number of co-location event over a long period between 2 users provides a hint that these 2 users are likely to have some form of social relationship. We define a relationship index over a period of time T:

$$RI = \sum_{1}^{n} (st_i + 1) * S_i / T \tag{3}$$

where n is the number of times that two users are in proximity. st_i is the time span of the related co-location event. S_i is the fingerprint similarity calculated using Equation 2. The algorithm of calculating the relationship index for two people is described in Algorithm 1.

There are three steps in this algorithm. First, we derive the similarity trace of two people. As frames are collected opportunistically and there is often frames loss, we use moving average to smooth fingerprint similarity. Next, we detect the starting and ending time points of interaction between two devices using the similarity threshold τ_{sim} derived from Figure 11. Based on these start and end points, we identify each social interaction event, its corresponding length and the people involved. Finally, interactions over time can be used to build up the relationship map.

5. EVALUATION

5.1 Implementation

SocialProbe system contains two parts: the frontend WiFi monitors and backend servers. WiFi monitors include Raspberry Pi 2B with two D-Link wireless USB adapters (DWA-137 and DWA-132).

Algorithm 1: Pairwise Relationship Estimation Algorithm

Input: Fingerprint Traces for two devices FT_1 and FT_2, Time Span T
Output: Relationship Index RI
S_{map} is map of timestamp and corresponding similarity, initially empty;
α is smooth factor, range in (0,1);
for *each fingerprint* ($\vec{f}_i$) *in* FT_1 **do**
  for *each fingerprint* ($\vec{f}_j$) *in* FT_2 **do**
    if *time stamp difference for* $\vec{f}_i$ *and* $\vec{f}_j$ *are within* t_s *gap* **then**
      Calculate Similarity S_{ij} and its timestamp t_{ij};
      if $t_{ij} - t_{(i-1)(j-1)} < t_s$ **then**
        $S_{ij} = \alpha * S_{ij} + (1-\alpha) * t_{(i-1)(j-1)}$;
      end
      add S_{ij} and t_{ij} into similarity map S_{map};
    end
  end
end
// identify co-location event and calculate Relationship Index RI_{base}
Initialize k and RI;
while k *not exceed the size of* S_{map} **do**
  Find next meeting time point I_{start} where $S_{start} > \tau_{sim}$ and $S_{start-1} < \tau_{sim}$;
  if *meeting point found* **then**
    Find leaving point I_{end}
    where $S_{end} > \tau_{sim}$ and $S_{end+1} < \tau_{sim}$;
  end
  calculate co-location time length st and
  mean similarity for S_m between I_{start} and I_{end};
  RI += $S_m * (st + 1)$;
  increase k to I_{end};
end
$RI = RI/T$;
return RI;

Figure 7(b) shows one of the monitors deployed in the research lab. In order to make sure that we have synchronized time in all the monitors, NTP runs every few hours. To reduce network traffic to the server, we only store a subset of the information in the frames including MAC address, timestamp, signal strength, monitor ID. After verifying that the MAC address is valid, we do not store the actual value but instead stored a hashed value of the MAC address.

We deployed the system in two environments: an 11m×13m research lab and 50m×29m dining hall. The solid triangle represents the monitors' positions on the map. The floor plan for the research lab is shown in Figure 7. The floor plan of the dining hall is shown in Figure 14. The dining hall is a dynamic and noisy environment with a lot of people moving around during the day. Figure 8 shows the number of unique mobile devices detected over 5 days, with the expected surge in traffic during lunch and dinner times.

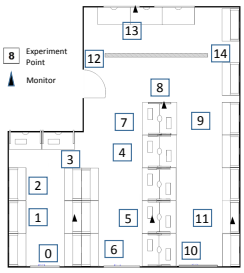
5.2 Device Filtering

In order to evaluate the effectiveness of the device filtering, we perform a small control experiment covering two research labs located side by side. We perform passive scanning for a week and use OUI and PR filter to identify valid devices. Since the experiment covers a relatively small area, we are able to get the ground truth by manually checking the devices present.

Figure 9 shows the performance of our filtering scheme with 56 devices that should be detected. Only use OUI gives us a low precision detecting valid device. Combining with PR filter, precision increases to 89% with only a little drop in recall rate.

5.3 Fingerprint Similarity

In this section, we do a measurement study to illustrate the effectiveness of our fingerprint normalization scheme, we plot fingerprint



(a) Research lab layout

(b) Raspberry pi monitor deployed with two dongles

Figure 7: Experimentation in 11m×13m Research lab. Square 0 to 14 label experiment positions.

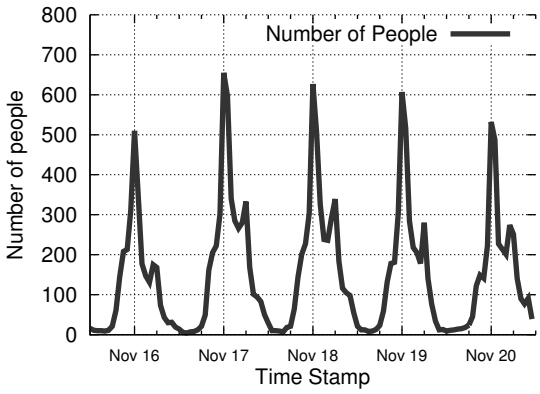


Figure 8: Mobile devices detected during 5 days in a dining hall

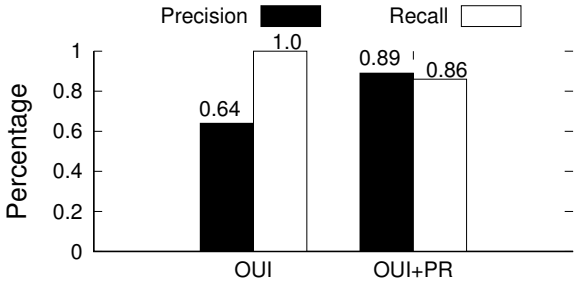


Figure 9: Performance for device filter method

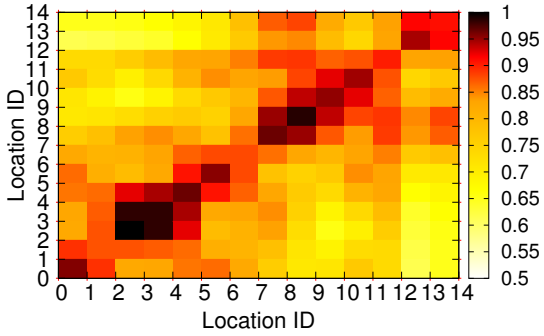


Figure 10: Fingerprint similarity comparison between 15 locations selected in research lab. Darker color represents higher similarity

similarity to distance relation in Figure 11. We can see that the similarity for normalized fingerprint drops significantly when the distance increases while the similarity for raw fingerprints remains high up to 12m in this experiment. We also compare the resolution improvement when we use mean, maximum or minimum signal strength as offset benchmark. We can see from Figure 11 that using maximum as offset standard can achieve a higher resolution.

We then do a site experiment in a 11m×13m research lab as shown in Figure 7. We select 15 locations evenly distributed in the lab which are marked 0 to 14 (in squares). We put 6 phones in each of the places and collect more than 500 frames per location. We calculate the similarity of these fingerprints collected in different places and in the same places using Equation 2 with maximum signal strength as offset benchmark. The results are shown in Figure 10.

As suggested in [9, 20, 8], if S_{ij} is larger than 0.7, we can conclude that the device has not moved. As shown in Figure 10, maximum similarity scheme can easily differentiate different locations. The overall co-location detection precision is 91% in the relative static lab environment and 84% from dynamic dining canteen. The high accuracy of co-location detection gives us more confidence on relationship inferring.

In the rest part of evaluation, we apply the information on co-location to discover behavior and interaction patterns.

5.4 Dining Hall

We deploy our system in a dinner hall on campus. The dining hall consists of food stalls, dining area, and tray collection area as shown in Figure 14. The positions of the monitors are located close to power outlets.

In this application scenario, we analyze the dining behavior. More specifically, we would like to discover clusters of diners dining together and how long each cluster took to finish their food. We calculated the stay time of people during dining time in two weeks and show the result in Figure 12. Around 90 percent of people finished their food in half an hour, while around 50 percent of people just stay in the canteen for less than 10 minutes. The reasons for a high percentage of short term stays are multi-folded. First, some people may just walk pass by without dining in the canteen. Further, some patrons could have a brief meal or order takeaway. Another possible factor is that the phones stop sending frame because of low battery or operating in power saving mode. Such durations do not provide sufficient information to infer dining group behavior. Thus in our system, we only analyze people who have stayed in the dining area for at least 10 min. We re-plot the dining curve after filtering these short duration presences.

To understand how the pair-wise similarity changes over time in the dining hall, we selected 5 users. Among these 5 users, 3 are people dining together (A,B,C) and 2 are randomly chosen from all the other users (a,b). Among these 5 users, we plot the similarity among 4 pairs. In two of these pairs, both users are from the same dining group and the similarity is expected to be high. In the other pairs, 1 user is from the chosen dining group and the other is a random user. We expect the similarity to be low. The result is shown in Figure 13.

In trace A-B and B-C, both users are from the same dining group. The similarity is maintained at a relatively high value of about 0.9 from 11:18am to 11:32am. In trace A-a, B-b, the users have a much lower value of similarity below 0.7. From the similarity to distance matching in Figure 11, we can easily draw the conclusion that A, B, C may be having lunch together but not with a and b.

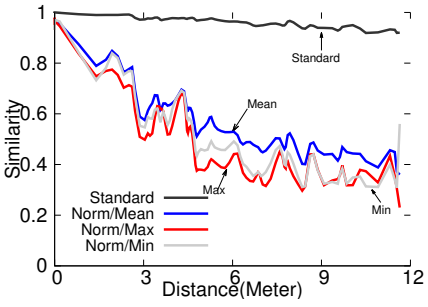


Figure 11: Estimation of fingerprint similarity to distance

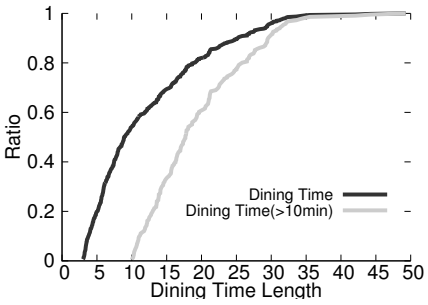


Figure 12: Dining Time CDF

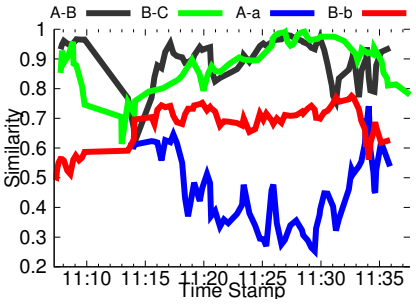


Figure 13: Similarity trend of people dining

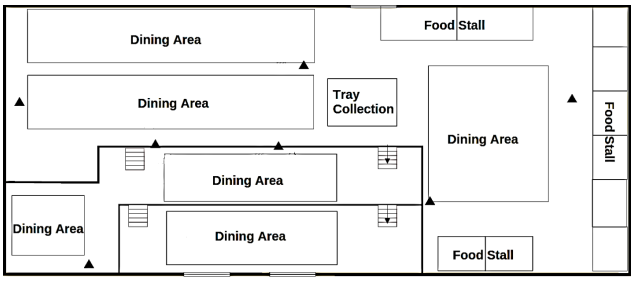


Figure 14: 50m×29m Dining hall. Solid triangles denoted monitors' positions.

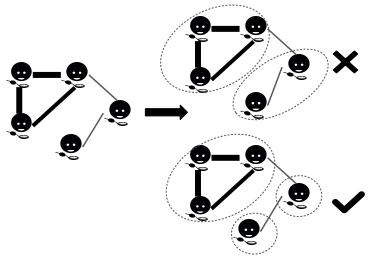


Figure 15: False positive case of MCL algorithm.

However, we also find similar valleys where similarity drops to a relatively low value in trace A-B and B-C. Recalling the experiment, this valley matches to the time when the diners disperse and order different food at their favorite counters. Thus, these traces perfectly describe the whole dining process. This experiment tells us the two people dining together should have high fingerprint similarity and synchronized time stamps. Based on these two facts, we derive the dining similarity index of person x and y as follow:

$$S_{dining} = 2 * RI * ST_{xy}/(ST_x + ST_y) \quad (4)$$

RI is the relation index we derive from Equation 3, ST_x and ST_y are the individual dining time of person x and y. ST_{xy} is the co-location time for the two persons. S_{dining} can be used to derive a dining graph of all the diners. Then we use the Markov Cluster algorithm (MCL) to detect dining groups. MCL works well when the cluster size is small as in our case. Such an approach has also been used in [27]. However, the initial result showed high recall rate for group clustering and also high false positive. Some diners with weak connections are unnecessarily grouped together as shown in Figure 15. Two persons in the right-bottom are falsely clustered as a group even though they have a weak connection. The problem lies in the normalization step for adjacency matrix. The weak connection can have a dominant effect on person with a low degree of connection, like the person at the bottom. Our case is different from pure group detection in that person can take a meal alone. Thus we improve the algorithm by filtering weak connection edges before we input the data to MCL. We analyze two weeks data of dining crowd and show the result of group clustering and dining length during four periods: breakfast, lunch, tea time, dinner in Figure 16.

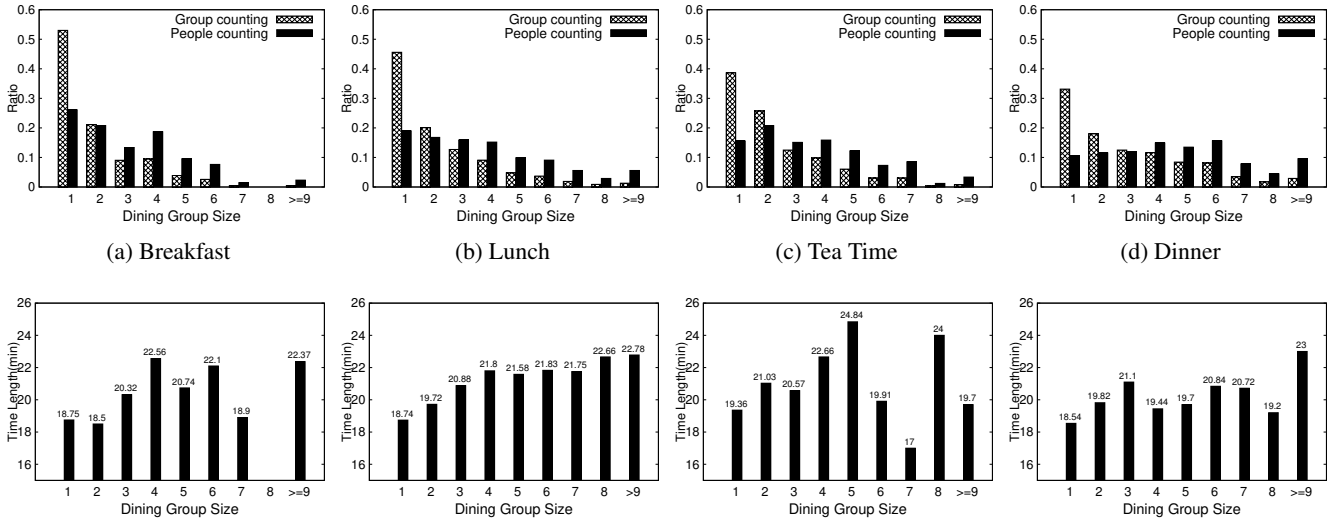
Figure 16 (a)-(d) show the group counting and people distribution during the four periods. Overall we can see that 80 percent of people tended to dine with their friends and 60 percent of people liked to form groups with less than 4 people. This may be explained by the fact that each table in the dining hall consists of four seats. Diners were more likely to dine alone during breakfast. During dinner time, more group behaviors were detected probably because there were less crowd. Figure 16 (e)-(h) show how the durations vary with different group sizes. In general, dining time increases when people are dining in a larger group. This confirms with our experience that people tend to talk more when dining with their friends and they wait for each other to finish the food.

Table 1: Dining behavior and partnership detection

Content/Day	1	2	3	4	5
Size of Group	6	5	6	7	8
Correct Number	6	5	5	5	6
False Positive	0	0	1	2	2
Estimated Time(min)	20	25	22	19	24
Ground Truth(min)	29	26	28	20	27

In order to gauge the accuracy of our dining detection, we performed an experiment with 6 people going for lunch together over 5 days.

The result of this experiment is shown in Table 1. During these five days, we have a good estimation of group size and its members compared to the ground truth. For example, in the fourth day, we identify 7 people as a group who may be having lunch together from around 500 diners. Among the 7 diners identified, 5 of them are correctly detected with 2 additional diners misclassified. All in all, the precision of group detection is 84% and the recall is 90%. The dining times we estimated are generally shorter than the ground truth. This is because the probe request and null data frames are sent opportunistically and we do not take into account the time needed for buying food. Even though the scale of this experiment is relatively small, it provides strong evidence that we can identify group of diners with high accuracy.



(a) Breakfast (b) Lunch (c) Tea Time (d) Dinner

(e) Breakfast (f) Lunch (g) Tea Time (h) Dinner

Figure 16: Dining group size distribution and dining time length

5.5 Research lab

Interaction in a research lab offers a different setup when compared to the dining hall. The number of people in the lab is relatively stable and the duration of stay is also much longer. Instead of analyzing short-term behavior like having lunch, we would like to infer their seating arrangement and build a relation map for members in the lab. Over 2 months, we detected 533 devices with monitors deployed in the research lab shown in Figure 7.

Infer Seating Arrangement In the previous experiment in the lab, we verified that fingerprint similarity can tell locations apart with high resolution. We would like to apply it to infer the sitting arrangement in this research lab. This process includes three steps: member identification, fingerprint extraction, and clustering.

First, we need to figure out who is working in this lab. This can be easily done by requiring a good quality signal strength since these monitors are physically close to the lab members. Next, to exclude users that walk pass the lab or visit the lab for a short time, we remove devices that stay for just for a short time. After these filtering, we can successfully identify 9 members from this lab.

Next, we need to extract the representative fingerprints at the users' actual seat locations. People do not spend all the time in the same place. From time to time, they move around, say to get a coffee or talk to another member. However, over a long time, they should spend most of the time working in their seats. We apply KMeans++ Cluster [2] to get the main fingerprint for each device. The use of KMeans++ is to avoid the poor initial center selection for kmeans.

Finally, we have to cluster these fingerprint to different groups to indicate their sitting arrangement. We still use MCL for it does not require a group number input and flexible on selecting the resolution. We show the result in Figure 17 with dotted circle. This result confirms with the ground truth marked in small black circle.

Relationship Map Using the relationship index *RI* defined in Equation 3, we can generate pairwise values to define the contact level between lab members from the same lab and from different labs. Figure 18 shows the relationship map generated from the data collected over two weeks. During this period, other than 9 members from lab A, we also identify 8 members from lab B. To evaluate

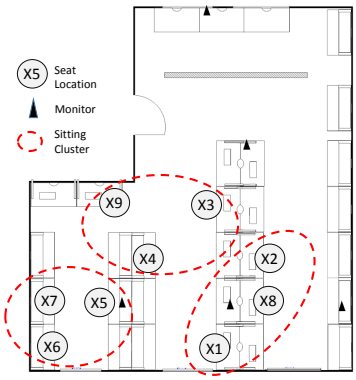


Figure 17: Sitting arrangement inferring result. Label X1 to X9 marked the ground truth location where they sit.

the accuracy of the relationship inferring, we collected ground truth only from lab A recording their daily behavior.

As we mentioned before, lab members generally spend most of their time working where they are seated. From the relationship map, we can see stronger links between people who sit closer, e.g. X5 and X6, X5 and X7, X4 and X9. Since each table is just 1.5 meters away, we expect such a result. Another relevant factor is that students who worked on the same project tend to interact more often. Such relationship can be shown in X1 and X4 who are seated farther apart but has a strong link.

Another observation that can be derived from the result and that is consistent with common sense is that members of the same lab show closer relation than those between the two labs. However, we can also derive working relationship between students between the two labs that was introduced as a result of this work, as shown by the connection between X5 and Y3.

6. RELATED WORK

Crowd Counting Variation in Bluetooth signal strength has been used to estimate crowd density [30]. The main intuition is that at high crowd density the signal strength variance would be higher than

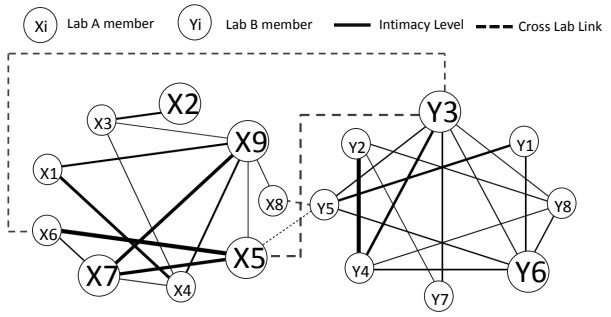


Figure 18: Relationship map based on *RI*. Members from lab A is marked by X1 to X9 with square shapes, lab B by Y1 to Y8 with circle shapes. The size of the node represents the degree of social activity. The stronger link means an intimate relationship between people.

at low crowd density. Similarly, analysis of probe request frame [17] and channel state information [32] can be used to estimate crowd size. Our work is different in that we do not focus on counting, but try to infer user behavior and interaction patterns.

Social Relationship Detection Social relationship detection and inference have long been discussed in the research community. Earlier methods use data of email, Facebook or phone call to generate dependency between people [14]. Recently, the use of probe frames to reveal user relationship has drawn attention from researchers. Cunche et al.[11] uses known SSIDs list in probe requests as the fingerprint to decide whether two devices are socially linked together. They applied modified Adamic-Adar, which was introduced earlier [1], as similarity metric over control data sets. Barbera et al.[4] uses an automated methodology to derive the underlying relationship graphs between the users. A similar method has been used to generate spatial-temporal similarity based on users' co-occurrence frequency to infer relationships between them [7]. However, as recent work has discovered that over 80% of the devices reply with empty (unknown) SSID list [18], approaches that rely on SSID information may not work well anymore.

WiFi-based Indoor Localization and Tracking Researchers have proposed a series of ideas to exploit the availability of WiFi networks including indoor localization[3, 33]. A key difference between our approach and localization is that we do not aim to localize a user. Instead, we focus on determining whether a user has moved and if two users are in proximity.

To improve accuracy, AP emulation and RTS injection to prompt additional probe frames have been used [24]. In other approaches, researchers have tried to analyze the crowd density and movement [25] and to optimize the facility planning from WiFi traces collected across buildings [26]. Our approach is different in that these approaches look at coarse level information such as whether users are on the same floor or building while we aim to discover more detailed and accurate social interaction on a personal level.

Human Communication and Group Detection Human communication such as conversation, encounter, and group activity is still an active research area because of its importance and difficulty. Mutual Information of human voice has been used to infer emerging conversation between people[5, 21]. GruMon [27] makes use of smartphone sensors and Markov Cluster algorithm to monitor closely related group activity. [29] is close to our job that targets at detecting short-lived human interaction. While they achieve high accuracy encounter detection, the modification of phones to schedule high frequency and synchronized probe request sending hinders its practical usage in real life. Instead, we try to detect co-location event with low-frequency frame and infer long-term relationship with unmodified mobile phone in a non-invasive way.

7. DISCUSSION

The sparse nature of frames transmission limits the performance of social interaction discovering. To get more frame transmission, the author in [24] propose to emulate the SSID of popular or previously visited AP. This technique can also be integrated into our system. But this technique triggers the use of WiFi interface of the mobile device which will interrupt the existing connection and drain the battery at a higher rate.

Even though we can detect social interactions and their pattern, we are unable to know whether they are having a conversation, greeting or quarreling. To further infer the detail of the emerging social event, we may need to deploy other sensors or install an application in the mobile device. Such methods extends the power of our social interaction system but contradicts with the non-intrusive nature of the probe scanning method.

From iOS 8, Apple introduced mac randomization to avoid passive scanning [1]. But it only works when smartphones are not connected to the network and the devices are in the sleep mode. In our evaluation, we are still able to hear probe request frames from Apple devices with valid MAC address. In addition, such technology has no effect on null frame detection. We are not sure about how a more effective mac randomization technology in the future will have an effect on our technology. But techniques have been proposed for monitors to track the WiFi devices with mac randomization[19].

In the two environments we deployed, the monitors were 5 and 15 meters apart. We used the same similarity threshold τ_{sim} as 0.8 rather than 0.7 in [9, 20, 8] to avoid high false positive. As we can see this threshold worked well for both environments. Based on the scenario of deployment, with monitors spacing 5-15 meter away, we believe this is a reasonable threshold for other settings as well.

Finally, it is possible for users to carry more than one device with them. Our device filter method will regard these two devices as two different users. We leave this problem for future work.

8. CONCLUSION

In this paper, we propose a system to discover social interaction based on opportunistic probe request and null data frames sent by mobile devices. We discuss the feasibility of using these frames to infer social events and using normalize fingerprint to figure out co-location events. This technique has been applied in a small office like environments as the research lab and public crowd area as the dining hall and we are able to discover relevant interaction patterns. We believe this is a good first attempt towards the understanding of human interactions using passive monitoring with infrastructure support.

Acknowledgements

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centre in Singapore Funding Initiative, Grant No. 61602319 from the National Natural Science Foundation of China and Natural Science Foundation of SZU (grant no. 2016048).

9. REFERENCES

[1] http://www.networkworld.com/article/2361846/wireless/ios-8-mac-randomizing-just-one-part-of-apple-s-new-privacy-push.html

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[3] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM*, volume 2, pages 775–784. Ieee, 2000.

[4] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa. Signals from the crowd: uncovering social relationships through smartphone probes. In *IMC*, pages 265–276. ACM, 2013.

[5] S. Basu. *Conversational scene analysis*. PhD thesis, MaSSachuSettS InStitute of Technology, 2002.

[6] B. Bonne, A. Barzan, P. Quax, and W. Lamotte. Wifipi: Involuntary tracking of visitors at mass events. In *WoWMoM*, pages 1–6. IEEE, 2013.

[7] N. Cheng, P. Mohapatra, M. Cunche, M. A. Kaafar, R. Boreli, and S. Krishnamurthy. Inferring user relationship from hidden information in wlans. In *MILITARY COMMUNICATIONS CONFERENCE, 2012-MILCOM 2012*, pages 1–6. IEEE, 2012.

[8] Y. Chon, S. Kim, S. Lee, D. Kim, Y. Kim, and H. Cha. Sensing wifi packets in the air: practicality and implications in urban mobility monitoring. In *UbiComp*, pages 189–200. ACM, 2014.

[9] Y. Chon, E. Talipov, H. Shin, and H. Cha. Mobility prediction-based smartphone energy optimization for everyday location monitoring. In *Proceedings of the 9th ACM conference on embedded networked sensor systems*, pages 82–95. ACM, 2011.

[10] R. B. Clayton, G. Leshner, and A. Almond. The extended iself: The impact of iphone separation on cognition, emotion, and physiology. *Journal of Computer-Mediated Communication*, 20(2):119–135, 2015.

[11] M. Cunche, M. A. Kaafar, and R. Boreli. I know who you will meet this evening! linking wireless devices using wi-fi probe requests. In *WoWMoM*, pages 1–9. IEEE, 2012.

[12] M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz. Predicting depression via social media. In *ICWSM*, page 2, 2013.

[13] A. developers. http://developers.apple.com/, 2014. [Online].

[14] C. P. Diehl, G. Namata, and L. Getoor. Relationship identification for social network discovery. In *AAAI*, volume 22, pages 546–552, 2007.

[15] S. Electronics. Sar evaluation report. In *SAR evaluation report*, pages 3–4. Samsung Electronics, 2013.

[16] J. Freudiger. How talkative is your mobile device?: an experimental study of wi-fi probe requests. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, page 8. ACM, 2015.

[17] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marrón, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and M. G. Fernández. Crowd density estimation for public transport vehicles. In *EDBT/ICDT Workshops*, pages 315–322, 2014.

[18] X. Hu, L. Song, D. Van Bruggen, and A. Striegel. Is there wifi yet? how aggressive wifi probe requests deteriorate energy and throughput. *arXiv preprint arXiv:1502.01222*, 2015.

[19] iMore. http://www.imore.com/closer-look-ios-8s-mac-randomization/, 2014. [Online].

[20] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 43–56. ACM, 2010.

[21] C. Luo and M. C. Chan. Socialweaver: collaborative inference of human conversation networks using smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 20. ACM, 2013.

[22] J. Mirowsky and C. E. Ross. Social pattern of distress. *Annual review of sociology*, pages 23–45, 1986.

[23] S. A. Moorhead, D. E. Hazlett, L. Harrison, J. K. Carroll, A. Irwin, and C. Hoving. A new dimension of health care: systematic review of the uses, benefits, and limitations of social media for health communication. *Journal of medical Internet research*, 15(4):e85, 2013.

[24] A. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*, pages 281–294. ACM, 2012.

[25] T. S. Prentow, A. J. Ruiz-Ruiz, H. Blunck, A. Stisen, and M. B. Kjærgaard. Spatio-temporal facility utilization analysis from exhaustive wifi monitoring. *Pervasive and Mobile Computing*, 16:305–316, 2015.

[26] A. J. Ruiz-Ruiz, H. Blunck, T. S. Prentow, A. Stisen, and M. B. Kjaergaard. Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning. In *PerCom*, pages 130–138. IEEE, 2014.

[27] R. Sen, Y. Lee, K. Jayarajah, A. Misra, and R. K. Balan. Grumon: fast and accurate group monitoring for heterogeneous urban spaces. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 46–60. ACM, 2014.

[28] S. Soto, E. M. Arredondo, M. T. Villodas, J. P. Elder, E. Quintanar, and H. Madanat. Depression and chronic health conditions among latinos: The role of social networks. *Journal of Immigrant and Minority Health*, pages 1–9, 2016.

[29] G. Vanderhulst, A. Mashhadi, M. Dashti, and F. Kawsar. Detecting human encounters from wifi radio signals. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, pages 97–108. ACM, 2015.

[30] J. Weppner and P. Lukowicz. Collaborative crowd density estimation with mobile phones. *Proc. of ACM PhoneSense*, 2011.

[31] T. Wiesenthal, G. Leduc, P. Cazzola, W. Schade, and J. Köhler. Mapping innovation in the european transport sector. *An assessment of R&D efforts and priorities, institutional capacities, drivers and barriers to innovation. JRC Scientific and Technical Report*, 2011.

[32] W. Xi, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang. Electronic frog eye: Counting crowd using wifi. In *INFOCOM, 2014 Proceedings IEEE*, pages 361–369. IEEE, 2014.

[33] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *NSDI*, pages 71–84, 2013.