



CISTER

Research Centre in
Real-Time & Embedded
Computing Systems

Journal Paper

An Experimental Study for Tracking Crowd in Smart Cities

Kai Li*

Chau Yuen

Salil S. Kanhere

Kun Hu

Wei Zhang

Fan Jiang

Xiang Liu

*CISTER Research Centre

CISTER-TR-181201

2018

An Experimental Study for Tracking Crowd in Smart Cities

Kai Li*, Chau Yuen, Salil S. Kanhere, Kun Hu, Wei Zhang, Fan Jiang, Xiang Liu

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: kaili@isep.ipp.pt, yuenchau@sutd.edu.sg, hk19900116@163.com, jungle.zw@gmail.com, jf864430830@163.com, xliu@ss.pku.edu.cn

<http://www.cister.isep.ipp.pt>

Abstract

Knowledge about people density and mobility patterns is the key element towards efficient urban development in smart cities. The main challenges in large-scale people tracking are the recognition of people density in a specific area and tracking the people flow path. To address these challenges, we present SenseFlow, a lightweight people tracking system for smart cities. SenseFlow utilizes off-the-shelf sensors which sniff probe requests periodically polled by user's smartphones in a passive manner. We demonstrate the feasibility of SenseFlow by building a proof-of-concept prototype and undertaking extensive evaluations in real-world settings. We deploy the system in one laboratory to study office hours of researchers, a crowded public area in a city to evaluate the scalability and performance "in the wild", and four classrooms in the university to monitor the number of students. We also evaluate SenseFlow with varying walking speeds and different models of smartphones to investigate the people flow tracking performance.

An Experimental Study for Tracking Crowd in Smart Cities

Kai Li , *Member, IEEE*, Chau Yuen , *Senior Member, IEEE*, Salil S. Kanhere , *Senior Member, IEEE*, Kun Hu, Wei Zhang, Fan Jiang, and Xiang Liu

Abstract—Knowledge about people density and mobility patterns is the key element toward efficient urban development in smart cities. The main challenges in large-scale people tracking are the recognition of people density in a specific area and tracking the people flow path. To address these challenges, we present *SenseFlow*, a lightweight people tracking system for smart cities. *SenseFlow* utilizes off-the-shelf sensors that sniff probe requests periodically polled by user's smartphones in a passive manner. We demonstrate the feasibility of *SenseFlow* by building a proof-of-concept prototype and undertaking extensive evaluations in real-world settings. We deploy the system in one laboratory to study office hours of researchers, a crowded public area in a city to evaluate the scalability and performance “in the wild,” and four classrooms in the university to monitor the number of students. We also evaluate *SenseFlow* with varying walking speeds and different models of smartphones to investigate the people flow tracking performance.

Index Terms—Mobile computing, sensor systems and applications, system analysis, system performance, wireless application protocol.

I. INTRODUCTION

IN RECENT years, the popularity of mobile and pervasive computing stimulates many research efforts on wireless people tracking for smart cities. An increasingly common requirement of people tracking is to extract information regarding the people density and moving trajectories [1]–[3]. In smart cities, many questions could be asked, e.g., how many customers visit a shopping mall everyday and which shops get more customers than the others in the mall? How many people are waiting in

a subway station, and how the flows of people move inside interchange stations? How long do the people walk from an entrance to an exit? Different from individual movement tracking, crowd tracking focuses on discovering proximity relationships, occupancy patterns, and social interactions among people based on overall crowd density and people flow movement. Generally, crowd tracking is useful in a diverse range of applications, including urban planning, psychology, and health care. Specifically, such crowd tracking information can help the smart city service provider understand public space usage patterns so as to improve their resource allocation, which makes the city a desirable place to work and live [4], [5]. Moreover, it is also a significance to understand pedestrian flows [6] and human mobility [7], e.g., for social psychology studies to sense citizen's mood based on their attitude toward joining flocks or for epidemiological studies to consider how often people join flocks [8]. Furthermore, flock detection can also enable new tools for emergency research studies, which is concerned with the size of flocks and how they form, dissolve, or are slowed down by constraints, e.g., door passages [9].

The integration of wireless sensing techniques and mobile devices, such as smartphones is enabling next generation lightweight people tracking applications for smart cities [10]. A possible way of people tracking is to utilize probe requests that are broadcast by smartphones for Wi-Fi connection [11]–[13] as a proxy for the people present in the area. People's trajectory can be tracked only when these spatial-temporal probe requests at different locations are fully collected by the sensors. Unfortunately, translating this broad idea into a practical people tracking system entails a range of challenges. First, a large number of probe requests (i.e., hundreds of people and smartphones in a crowded city area) are generated in real time. Forwarding all the probe requests generates a peak of traffic in the network, which poses a challenging problem on the data collection and processing. Second, the probe request generating interval highly depends on operational mode of a mobile device (shown by our experiments in Section IV-A). As a result, the smartphone will not be tracked if its probe request is missed when the user moves across the sensing range of the sensor. Third, tracking people flow in a spacious area is non-trivial since multiple trajectories are available between any two locations, and the exact people movement pattern is often unknown. Additionally, the probability of probe request detection decreases with pedestrian walking speed since the sensor node has a limited sensing coverage. However, no existing work studies how the probe request

Manuscript received April 16, 2018; revised July 21, 2018 and October 10, 2018; accepted November 4, 2018. This work was supported in part by the International Design Center and NSFC under Grant 61750110529, and in part by the National Funds through FCT (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (CEC/04234). This paper was presented in part at the Proceedings of the 6th ACM Workshop on Real World Wireless Sensor Networks, The Plaza Hotel, Seoul, South Korea, November 2015. (*Corresponding author: Chau Yuen.*)

K. Li is with the Real-Time and Embedded Computing Systems Research Centre (CISTER), Porto 4200-072, Portugal (e-mail: kaili@isep.ipp.pt).

C. Yuen is with the Singapore University of Technology and Design, Singapore, 138682 (e-mail: yuenchau@sutd.edu.sg).

S. S. Kanhere is with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: salil.kanhere@unsw.edu.au).

K. Hu, W. Zhang, F. Jiang, and X. Liu are with the School of Software and Microelectronics, Peking University, Beijing 100871, China (e-mail: hk19900116@163.com; jungle.zw@gmail.com; jf864430830@163.com; xliu@ss.pku.edu.cn).

Digital Object Identifier 10.1109/JSYST.2018.2880028

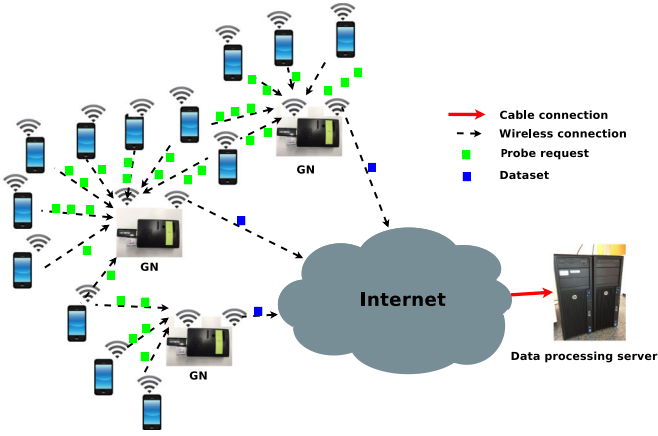


Fig. 1. System architecture of *SenseFlow*. Each sensor node, i.e., GN, covers a specific area and the datasets of GN are uploaded to the sever through public network.

frequency and human walking behavior effect people flow tracking performance.

To address the above-mentioned issues, we propose *SenseFlow*, a lightweight sensing testbed to monitor people quantity in a given area and track people flow movement, based on a passive collection of the probe requests from their smartphones, without knowing the environmental feature or fingerprint. Specifically, *SenseFlow* uses a number of time-synchronized wireless gateway sensor nodes (GNs) to collect probe requests broadcasting by users' smartphones at different locations, as shown in Fig. 1. In particular, GNs are dispersedly deployed to cover a specific interesting area, e.g., offices in a building, shops in a shopping mall, bus/metro interchange station, etc. A user's presence is detected when the probe request of the smartphone is received by the GN the user passes by. Therefore, the trajectory of the user is obtained by tracking the probe requests on a series of GNs according to time. In *SenseFlow*, data packets of GNs, named as dataset, are uploaded to a backend server through existing public network facilities, e.g., Wi-Fi access points. Essentially, the server amalgamates the data from all GNs so as to investigate when and which GNs collect the probe requests. In our preliminary work [1], [14], we demonstrate a proof-of-concept prototype of a simplified *SenseFlow*, where probe request intervals with regard to smartphone statuses and human walking behavior are measured. In the simplified system, GNs transmit all probe requests in real time generates a large network traffic. To address this issue, in this paper, we propose a novel probe request interval based data collection scheme (PRI-DCS) to provide a real-time people flow tracking. Specifically, the GN extracts source media access control (MAC) address and timestamp from probe request, and uploads the extracted data (dataset) according to the probe request interval of smartphone. The datasets transmitted from all GNs are amalgamated to monitor people density of the area by counting the amount of MAC addresses during a certain time interval.

Furthermore, we find that the people tracking system based on smartphone monitors people density inaccurately due to a probe request overhearing problem, where the nodes deployed at adjacent locations can receive probe requests from the same

smartphone. To address this issue, in *SenseFlow*, we extend the PRI-DCS by selecting the GN that has the maximum received signal strength indicator (RSSI) of probe requests as the location where the smartphone presents. We formulate the trajectory of individual user as a sequence of GN IDs in the datasets. Next, we implement a longest common subsequence (LCS) algorithm to recognize the user's trajectory, and track the flow of people from one specific starting point to an ending point.

The rest of this paper is organized as follows. Section II presents the related work on different types of people tracking system in smart cities. We then present the details of the proposed *SenseFlow* system in Section III. Section IV illustrates the implementation and evaluation results in both controlled and real-world experiments. Finally, we conclude this paper in Section V.

II. RELATED WORK

In this section, we present a brief overview of previous work of people tracking systems covering the people density monitoring and movement tracking.

A. Sensor-Based People Tracking Systems

The GPS-based localization system is widely used for outdoor position determination and this technology is currently implemented in many mobile devices [15]. Unfortunately, the main challenge in indoor environments is the unavailability of GPS signals since the technology requests for line of sight when connecting to satellites. In addition, such system requires the user to install an application on the mobile device in order to enable GPS localization, which does not track people in a passive way.

Camera-based system was proposed to address the people tracking using thermal infrared, stereo, and time-of-flight camera [16], [17]. The vast majority of human-detection approaches currently deployed in camera-based systems rely on background subtraction, pattern matching, and face recognition, which can process the conventional images from the camera. However, these systems are affected by lighting variations and shadows. Moreover, camera-based system has limited coverage due to a fixed location and angle [18].

Apart from cameras, other devices used for people tracking are range finders, such as radar and sonar. In [19], Mozos *et al.* propose people tracking by using multiple layers of two-dimensional laser range scans. Premebida *et al.* [20] present a valuable analysis of pedestrian detection in urban scenario using exclusively lidar-based features. Unfortunately, the impressionable wave and laser signal lead a large number of false negatives [3].

Sensor fusion approaches build upon the use of multiple sensors, such as camera and microphone [21], camera and range finder [22], camera and motion sensor [23], etc. The idea is to combine their advantages while cancelling out their disadvantages as much as possible. Unfortunately, sensor fusion systems require the installation of a complex infrastructure, which causes a large capital investment in setup [3]. In addition, the state-of-the-art sensor fusion systems can hardly meet the accuracy and delay requirement for large-scale people tracking.

TABLE I
COMPARISON OF PEOPLE TRACKING SYSTEMS

	Camera	GPS	Range finders	Sensor fusion	SenseFlow
Indoor tracking	Yes	No	Yes	Yes	Yes
Large-scale people tracking	No	Yes	No	Yes	Yes
Trajectories recognition accuracy	Low	High	Low	Low	High
Tracking latency	High	Low	Low	High	Low
System Complexity	High	Low	Low	High	Low

The comparison of people tracking systems in the literature is summarized in Table I.

B. People Tracking With Smartphone

RSSI samples of Wi-Fi signal [24] and ambience fingerprinting [25] have been researched for mobile device based indoor localization. In [24], RSSI samples captured on a smartphone with Wi-Fi connection are utilized to detect user's presence in the area. It is found that sampling rate and distance between the smartphone and Wi-Fi gateway can affect the detection accuracy. In addition, user's activities, such as standing, sitting, lying, and walking, can be recognized by classifying features of RSSI samples. By combining optical, acoustic, and motion attributes from various sensors of the mobile phone, *Surround-Sense* system constructs an identifiable fingerprint, which includes ambient sound, light, color, RF, and user movement [25]. This fingerprint is then used to identify the user's location.

In [26], the authors present studies of people's behavior in large building complexes utilizing measurements of WiFi signals from peoples' mobile devices. Several analysis methods are provided to estimate when and where users (respectively, their mobile devices) enter and leave buildings. Moreover, the people at the hospital are differentiated into roles, such as staff and visitors so as to obtain information about behavior of respective groups inside the buildings.

A tracking system, which consists of a number of road-side Wi-Fi monitors and a central server, is presented in [13]. They propose a probabilistic method to estimate smartphone trajectories for single user from Wi-Fi detections. It is shown that the accuracy of Wi-Fi tracking depends to a large degree on the density and geometry of monitors' deployment.

A low-cost tracking system for pedestrian flow estimations based on Bluetooth and Wi-Fi captures is proposed in [27]. The system tracks a pedestrian's movement through an area of interest by capturing the device specific MAC address at different monitor nodes located at the entrances/exits to this area of interest. Furthermore, they propose a hybrid tracking approach that considers both the RSSI value and the time when a MAC address was captured.

In [28], the social links in a venue of large political and religious gatherings are studied from the probe requests. A database

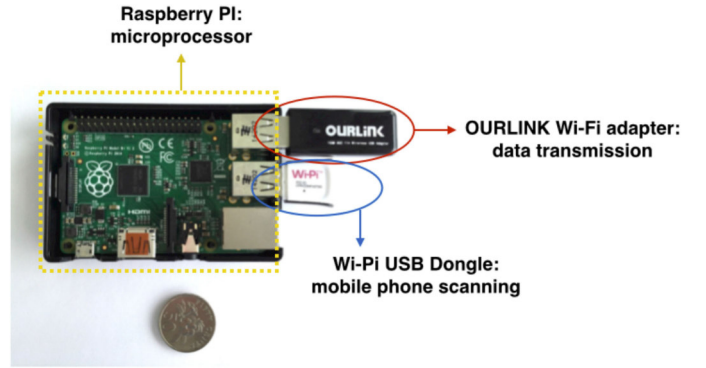


Fig. 2. GN is a Raspberry PI connecting with Wi-Fi (white color) and OURLINK (black color) wireless interfaces.

that associates each device is built, as identified by its MAC address, to the list of service set identifiers (SSIDs) derived from its probe requests. Moreover, an automated methodology is proposed to learn the social links of mobile devices given that two users sharing one or more SSIDs indicate a potential social relationship between the two.

Compared with the existing solutions and concepts, we experimentally measure the probe request interval with different operational modes of three typical smartphone operating systems and the effect of human walking behavior. With these studies, our people tracking system collects dataset packets from GNs based on the probe request interval to mitigate network traffic and tracking latency. Moreover, our approach is able to address the probe request overhearing problem to improve the tracking accuracy.

III. SENSEFLOW SYSTEM

In this section, we first present the design of gateway node and system architecture with a new data collection scheme. Next, we study the probe request overhearing problem in the people tracking system. We then outline a people tracking algorithm based on the datasets from GNs.

A. Probe Request Collection and Overhearing

We implement the GN (shown in Fig. 2) based on Raspberry PI, which connects with a Wi-Fi USB dongle for probe request collection, and a OURLINK Wi-Fi adapter so that the dataset of GN can be transmitted to the server through existing Wi-Fi network. Both of them work in 2.4 GHz. The reason of using two wireless interfaces is that dual wireless interface system achieves probe requests collection while transmitting the datasets simultaneously. Significantly, using GNs in *SenseFlow* has the following merits. First, GN is a portable plug and play device, thereby, *SenseFlow* can be extended to a large-scale deployment and easy to be maintained. Second, GN can wirelessly upload its data through existing Wi-Fi network without changing the network infrastructure. Third, Raspberry PI that is a programmable single-board computer allows users to develop different applications in the people tracking system. Fourth, thanks to low price of the Raspberry PI, setup and maintenance costs of GNs are both cheap.

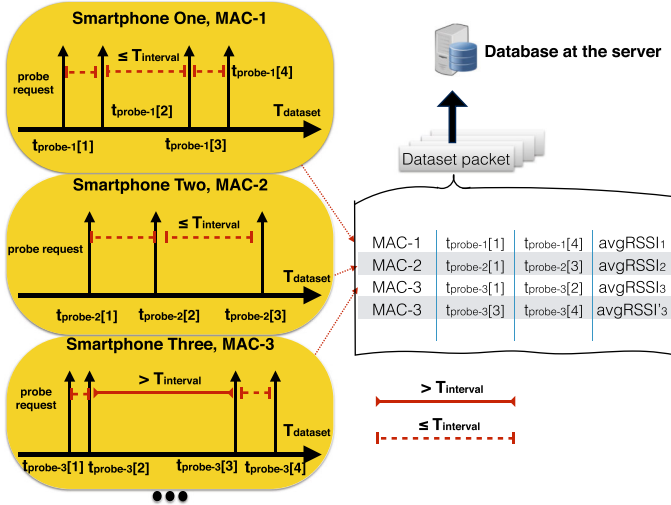


Fig. 3. PRI-DCS of *SenseFlow*. In each T_{dataset} , MAC address is provided by the probe request from the smartphone.

Different smartphone models may have different probe request definitions. Generally, the probe request contains the type of frame, duration, source MAC address, BSSID, SEQ, etc., [29], [30]. We propose a PRI-DCS for *SenseFlow*. The PRI-DCS is shown in Fig. 3 and Algorithm 1. T_{dataset} is the time to transmit the dataset packet. $t_{\text{probe-}i}[\cdot]$ denotes the time of the probe request from the smartphone in T_{dataset} , e.g., $t_{\text{probe-1}}[2]$ is the second probe request from smartphone one. We define T_{interval} as a time threshold to merge two probe requests of the smartphone. Specifically, if the time interval between two consecutive probe requests is smaller than T_{interval} , the smartphone is assumed to be not moving, and only one record that contains timestamp of the first probe request and the last one is kept by the PRI-DCS. Otherwise, both timestamps are kept in two independent records as the smartphone may leave and return to the coverage of the GN. Our aim to do so is to reduce the packet size to be uploaded. Finally, the MAC address of the smartphone, timestamps, and average of RSSI of the probe requests are added to the dataset packet that is transmitted to the server for people tracking (details are presented in Section III-B). Moreover, time and space complexity of PRI-DCS algorithm is $O(S_j)$, where S_j is the number of records in dataset packet $D_j(t)$.

Most of people tracking systems are based on the probe requests received by the distributed sensor nodes from the smartphones. However, the sensor node at one location can receive the probe requests of a smartphone at an adjacent location due to an overlapping coverage area. As a result, those systems are not able to monitor accurate people density and track people flow since some of smartphones are captured at multiple locations at the same time, which we name *probe request overhearing problem*. In *SenseFlow*, we extend the PRI-DCS by utilizing RSSI deviation at different locations to address this probe request overhearing problem. Although the RSSI of the probe request does not depict a precise location of the smartphone, it implies how far the smartphone is away from the GN since the RSSI measurements attenuate in distance with a power decay factor. Therefore, when multiple GNs receive the probe request

Algorithm 1: PRI-DCS Algorithm.

- 1: The timestamp of the latest probe request of smartphone i is $t_{\text{probe-}i}$.
 - 2: **if** A new probe request x_i is received by GN j . **then**
 - 3: $x_i = x_i + 1$, the timestamp of x_i is $t'_{\text{probe-}i}$.
 - 4: The connection time in T_{dataset} is: $T(i, j) = t'_{\text{probe-}i} - t_{\text{probe-}i}$.
 - 5: **if** $T(i, j) < T_{\text{interval}}$ **then**
 - 6: Update the latest timestamp of smartphone i as $t'_{\text{probe-}i}$ in $D_j(t)$.
 - 7: **else**
 - 8: Create a new record for smartphone i with timestamp $t'_{\text{probe-}i}$ in $D_j(t)$.
 - 9: **end if**
 - 10: **if** $\text{SystemClock} == T_{\text{dataset}}$ **then**
 - 11: Calculate average of RSSI for each MAC address.
 - 12: GN j uploads $D_j(t)$ to the server.
 - 13: **end if**
 - 14: **else**
 - 15: The GN keeps listening.
 - 16: **end if**
-

from the same smartphone at the same time, the GN that is closest to the smartphone has the highest RSSI value. Specifically, in *SenseFlow*, the GN calculates an average value of RSSI for each smartphone in T_{dataset} . The RSSI values are appended to the corresponding MAC address in the dataset packet (shown in Fig. 3). On the server, if any MAC address is captured in dataset packets from multiple GNs, we select the GN that has the maximum RSSI value as the location where the smartphone presents.

B. *SenseFlow* People Tracking Algorithm

In order to monitor people density and track people flow, a *SenseFlow* people tracking (SFPT) algorithm is proposed. Details are provided in Algorithm 2. Specifically, the SFPT algorithm amalgamates dataset packets to monitor people density and track people movement on the server. Based on the spatial-temporal dataset $D_j(t)$ of all GNs, the MAC address of smartphone i with timestamp is extracted. To address the probe request overhearing problem, each MAC address only keeps one GN who has the highest RSSI at any time. Therefore, people density at any location is known by counting the unique MAC addresses that connect to the GN j at that location.

The trajectory of smartphone i can be known by tracking the GNs that receive its probe requests according to time. Specifically, we formulate the trajectory as a sequence of GN IDs, which is denoted as $\vec{X}_i = (x_1, x_2, \dots, x_n)$, where x_n is ID of the GN that has the strongest RSSI of probe request. Given a targeting trajectory \vec{J} , we decide whether smartphone i has ever travelled along \vec{J} based on the LCS of \vec{X}_i . If all GNs in \vec{J} detect the probe requests from smartphone i , the LCS of \vec{X}_i and \vec{J} is \vec{J} . The number of people moving along \vec{J} is obtained by counting the users who fulfil $\vec{J} = \text{LCS}(\vec{J}, \vec{X}_i)$. Note

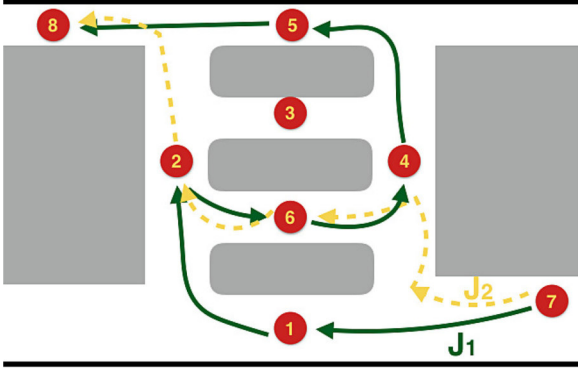


Fig. 4. Example for trajectory recognition: Eight GNs are deployed along the walking path in an area.

Algorithm 2: SFPT Algorithm.

```

1: if  $D_j(t)$  is received from the GN  $j$  then
2:    $D_j(t)$  is sorted by unique MAC address of the
     smartphone  $i$ .
3:   for  $i \leq$  the total number of users do
4:      $[RSSI_i, j'] = \max(RSSI_i[1], RSSI_i[2], \dots,$ 
        $RSSI_i[j, \dots])$ .
5:      $PeopleDensity[j'] = PeopleDensity[j'] + 1$ .
6:     if  $\vec{J} = \text{LCS}(\vec{\mathcal{X}}_i, \vec{J})$  then
7:       The number of people in  $\vec{J}$  increases by 1.
8:     else
9:       Continue.
10:    end if
11:  end for
12: end if

```

that LCS in an SFPT algorithm only evaluates two trajectories regardless of the total number of GNs in the system, i.e., the targeting trajectory and the trajectory containing the GNs that receive the probe request from the smartphone with the strongest RSSI. In terms of computational efficiency, as the LCS problem is solvable in polynomial time by dynamic programming [31], time complexity and space complexity of an SFPT algorithm are $O(|\vec{\mathcal{X}}_i| \cdot |\vec{J}| + S_j \log(S_j))$ and $O(|\vec{\mathcal{X}}_i| \cdot |\vec{J}| + S_j)$, respectively.

For trajectory recognition, as an example shown in Fig. 4, we have eight GNs along the walking path in an area. From GN (7) to (8), we consider two targeting trajectories, $\vec{J}_1 = (7, 1, 2, 6, 4, 5, 8)$ and $\vec{J}_2 = (7, 4, 6, 2, 8)$. Assume that the system detects a smartphone moving from GN (7) to (8), and the trajectory is given by $\vec{\mathcal{X}}_1 = (7, 1, 6, 5, 8)$. Then, we have $\text{LCS}(\vec{J}_1, \vec{\mathcal{X}}_1) = (7, 1, 6, 5, 8)$ and $\text{LCS}(\vec{J}_2, \vec{\mathcal{X}}_1) = (7, 6, 8)$. Therefore, by using LCS, the system recognizes that $\vec{\mathcal{X}}_1$ travels along \vec{J}_1 , not \vec{J}_2 since $\text{LCS}(\vec{J}_1, \vec{\mathcal{X}}_1)$ has more nodes than $\text{LCS}(\vec{J}_2, \vec{\mathcal{X}}_1)$. Additionally, the more GNs detect the smartphone, the more accurate trajectory recognition *SenseFlow* achieves. Consider an extreme case that only three GNs detect a smartphone, $\vec{\mathcal{X}}_2 = (7, 3, 8)$. Then, we have $\text{LCS}(\vec{J}_1, \vec{\mathcal{X}}_2) = (7, 8)$ and $\text{LCS}(\vec{J}_2, \vec{\mathcal{X}}_2) = (7, 8)$. The trajec-

TABLE II
AVERAGE PROBE REQUEST INTERVAL OF SMARTPHONES IN
DIFFERENT WI-FI AND SCREEN MODE

Smartphones	non-registered Wi-Fi		registered Wi-Fi	
	screen on	screen off	screen on	screen off
iOS	70.6s	109.8s	1200.8s	1204.4s
Android	0.8s	1s	2.11s	2.15s
Windows	10.9s	13.9s	1200.8s	1204.4s

tory of the smartphone is not able to be tracked since it is hardly detected by GNs. To address this issue, one practical solution could be deploying multiple duplicated GNs around a specific location to increase the probe request receiving probability. The smartphone is deemed to traveling the location as long as one of the GNs at the location receives its probe request.

IV. EXPERIMENTS ON TESTBED AND EVALUATION

In this section, we first present the experiments to show the effect of probe request transmitting interval on *SenseFlow*. Then, the extensive experiments for monitoring people density are conducted in four classrooms and one laboratory on the SUTD University campus and a crowded public area in Singapore. Next, we measure how people walking behavior effects the probe request detection under different smartphone's operational modes. We evaluate the performance of *SenseFlow* for tracking people flow on our testbed in the university.

A. Probe Request Interval Measurement

Generally, on any smartphone, probe requests are transmitted in bursts, the interval of which depends on the OS and Wi-Fi chipset driver and varies greatly according to status of the Wi-Fi connection and screen mode [32], [33]. Some of smartphones may not be captured since the probe request interval ($t'_{\text{probe}-i} - t_{\text{probe}-i}$) (in Algorithm 1) is longer than the time when the people move across the sensing range of GN, which degrades the tracking accuracy of the system. To understand how smartphones affect the performance, we evaluate the system with four operational modes, (Wi-Fi registered, screen ON), (Wi-Fi non-registered, screen ON), (Wi-Fi registered, screen OFF), and (Wi-Fi non-registered, screen OFF). Specifically, "Wi-Fi non-registered" implies that the phone is not authorized to connect to a Wi-Fi network. Note that the phone in Wi-Fi non-registered mode still broadcasts probe requests to search Wi-Fi access points for network connection from time to time. We also note that the Wi-Fi non-registered mode does not disable smartphone's cellular connection, in other words, the phone can connect to Internet over 3G or 4G cellular network.

In this experiment, we choose three typical smartphone models, two iOS phones (one iPhone 4 and one iPhone 4S), five Android phones (one Samsung Galaxy Nexus, one ASUS MeMo pad, three Sony Xperia Z5), and one Windows phone (Nokia Lumia 520). Table II presents average probe request interval of smartphones in different Wi-Fi and screen conditions. Specifically, in screen OFF and Wi-Fi non-registered mode, the smartphones increase the probe request interval to conserve battery power comparing to the mode of screen ON. Moreover, iOS

	$T_{\text{interval}} = 5\text{mins}$	$T_{\text{interval}} = 10\text{mins}$	$T_{\text{interval}} = 20\text{mins}$
$T_{\text{dataset}} = 10\text{mins}$	506KB	470KB	476KB
$T_{\text{dataset}} = 30\text{mins}$	255KB	216KB	192KB
$T_{\text{dataset}} = 60\text{mins}$	208KB	158KB	131KB
$T_{\text{dataset}} = 120\text{mins}$	186KB	119KB	100KB

Fig. 5. Data traffic with different T_{dataset} and T_{interval} .

phone and Windows phone have a long interval around 1200 s when the smartphone has connected to Wi-Fi network. However, Android phone still keeps a short interval of 2.11 and 2.15 s. The different probe request interval is caused by a differentiated energy-saving design of smartphones in Wi-Fi registered mode. In addition, the results of Table II also indicate the longest T_{interval} in *SenseFlow*.

B. T_{dataset} and T_{interval} Characterization

The GN in *SenseFlow* transmits the dataset to the public network wirelessly. A practical question is how much data traffic will the GN generate everyday? This issue is crucial when the data are forwarded to the server via cellular network since more data traffic causes higher data bill. Therefore, we next study the impact of T_{dataset} and T_{interval} on the daily data size. In this experiment, we deploy 12 GNs with different T_{dataset} and T_{interval} configuration in a laboratory on the SUTD University campus. We run the experiment for one day (1440 min), and analyze the total data size collected from the GN.

The experimental results are shown by Fig. 5. The maximum data traffic is 506 KB when T_{dataset} is 10 min and T_{interval} is 5 min. With an increase of T_{dataset} , the GN uploads data in a long transmission interval where the unique MAC address is merged and the number of records is reduced significantly. For example, given $T_{\text{interval}} = 5$ min and $T_{\text{dataset}} = 120$ min, the daily data size is reduced to 186 KB. Moreover, Fig. 5 shows that increasing T_{interval} also reduces data traffic. The reason is the records of the MAC address that fulfils $(t'_{\text{probe}-i} - t_{\text{probe}-i}) < T_{\text{interval}}$ are merged to one record.

C. People Density Monitoring

To monitor people density in a public area, we deploy *SenseFlow* in three applications: One laboratory room in the university to observe office hours of researchers, a crowded area in city to learn people density in public, and four closely located classrooms on the SUTD University campus to study the number of students in the classrooms.

1) *Laboratory in University*: We deploy one GN in a laboratory room on campus for 7 d \times 24 h continuous people tracking, as shown in Fig. 6. A commercial off-the-shelf gateway device, Meshlium model [34], produced by Libelium, is deployed at the same location. Libelium node can work as a passive smartphone scanner, where smartphones are detected without the need of being connected to a specific access point. Typically, Libelium node is equipped with a Wi-Fi 2.4 GHz radio interface, and its scanning range and transmit power are around 500 m and 20 dBm, respectively. Furthermore, we personally visit the lab-

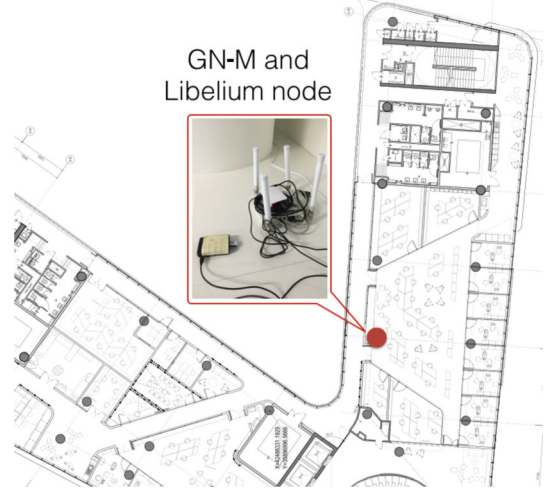


Fig. 6. GN-M and Libelium device are deployed at the same location in a laboratory in the university.

oratory at several specific time, e.g., 10:00A.M., 11:00A.M., and 1:00P.M., and record headcount from 10:00A.M. to 6:00P.M. every weekday as the ground truth of this experiment.

Fig. 7(a) and (b) shows the number of unique smartphones detected by the GN-M and Libelium node. From Monday to Friday, both of the nodes detect more people in the daytime, from 8:00A.M. to 5:00P.M., than the time before dawn and midnight. This result is reasonable because most of researchers use the laboratory during office hour. Moreover, the amount of smartphones detected by Libelium node is larger than the detections of GN-M for around 10 phones. The reason is that Libelium node has higher packet receiving sensitivity and larger signal coverage than our GN. Therefore, Libelium node is able to detect the smartphones in other rooms. In Fig. 7(a), we also see that the people density decreases significantly on weekends. Around 20 smartphones are detected between 1:00P.M. and 5:00P.M. on Saturday. During the other time, only 7 smartphones are detected. This can be explained by the fact that most of researchers did not present in the laboratory out of the office hour. Fig. 7(c) shows the number of people recorded by the weekday head count at specific time between 10:00A.M. and 6:00P.M. Based on Fig. 7(a) and (c), we can see that the amount of smartphones detected by our GN is closer to the number of people recorded by head count during daily office hours. For example, our GN detects 29 smartphones at 4:00P.M. on Friday while our recorder observes 27 researchers in the laboratory.

2) *Crowded Public Area in City*: To test system scalability, GN-A and GN-C are placed along the walking path in a crowded city area in Singapore, as shown in Fig. 8. The experiment was carried out from 11:00P.M. October 28, 2014 to 4:00P.M. October 29, 2014 (30 h in total). The people counting performance is shown in Fig. 9. It can be observed that there are three peaks of people quantity at 9:00A.M., 12:00P.M., and 6:00P.M. in one day. The results indicate that they are rush hours and many people go through the public area. Moreover, more smartphones connecting to GN-A than GN-C indicate that the location of GN-A is more popular than the one of GN-C in the monitoring

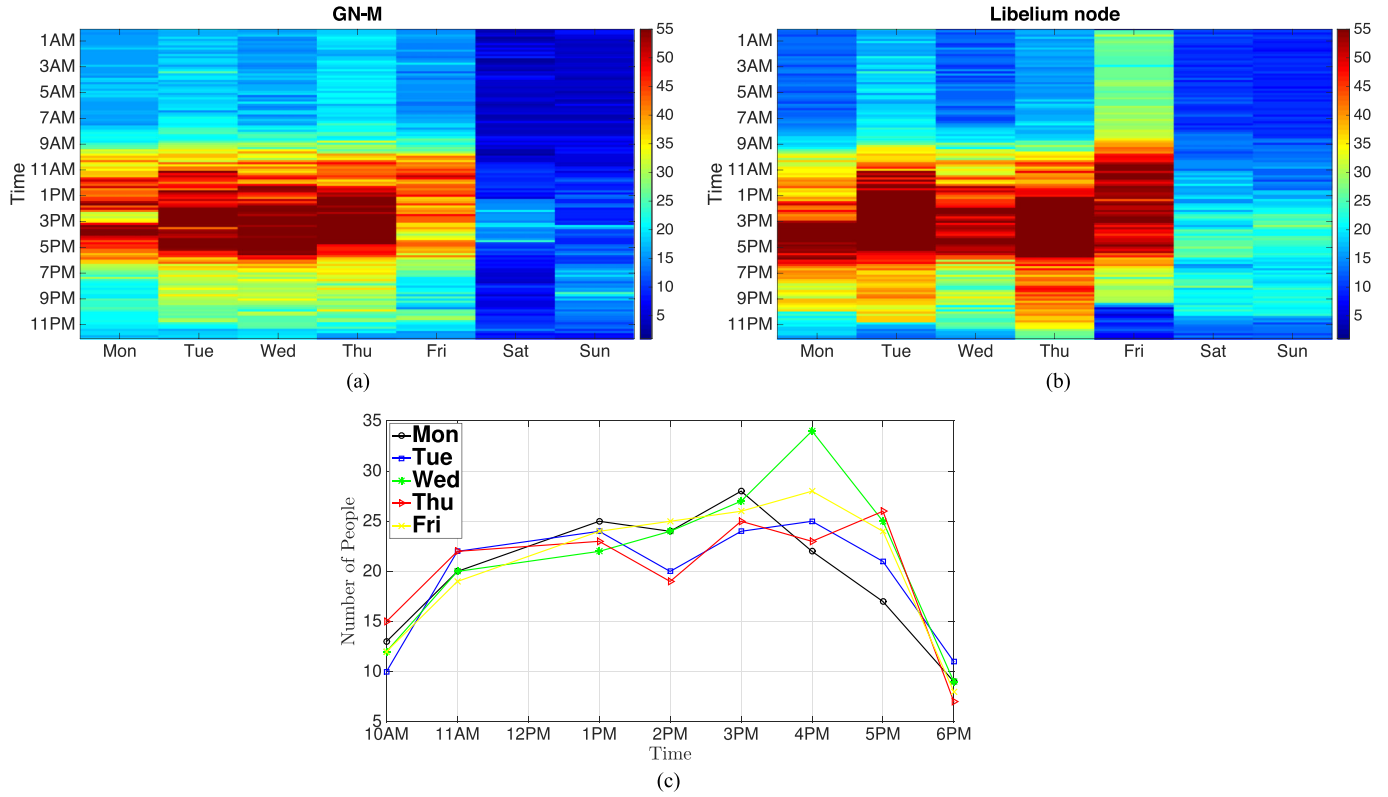


Fig. 7. People density monitoring in a laboratory in the university from Monday to Sunday (from January 4, 2016 to January 10, 2016). (a) Smartphones detected by GN-M. (b) Smartphones detected by Libelium node. (c) Number of people at specific time recorded by head count.

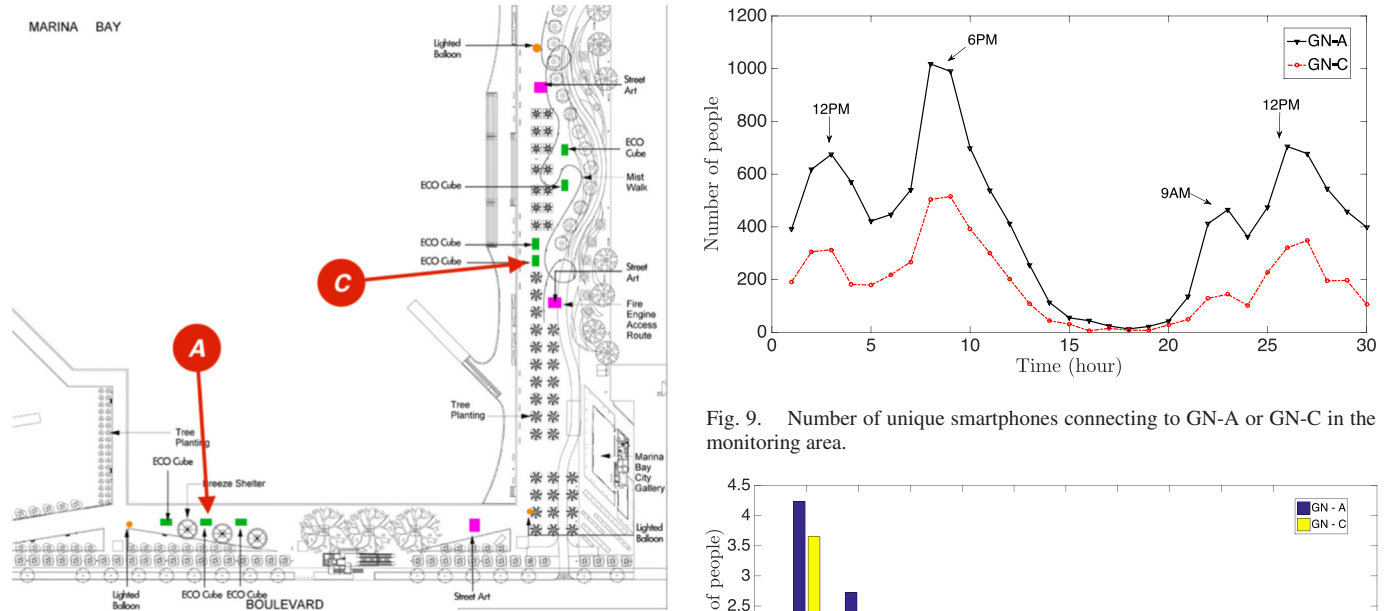


Fig. 8. GN-A and GN-C are deployed in a public area.

area. This result can be used for guidance of city planning and promotion of coordinated development of the public area.

Based on the SFPT algorithm in Section B, how long each people stays in the monitoring area can be known. Due to a large variance on the people, Fig. 10 shows the number of devices connecting to GN-A or GN-C during one day in *log* scale.

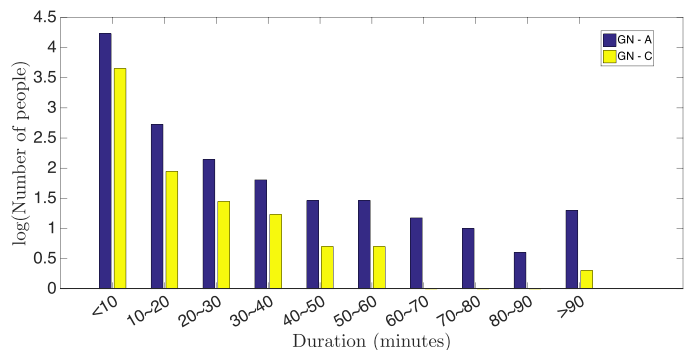


Fig. 10. Number of devices connecting to GN-A or GN-C in a specific duration.

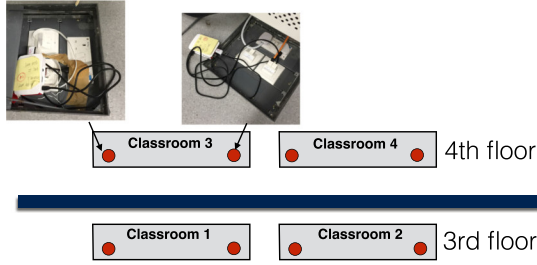


Fig. 11. Eight GNs are deployed in four adjacent classrooms in the SUTD. Each classroom has two GNs.

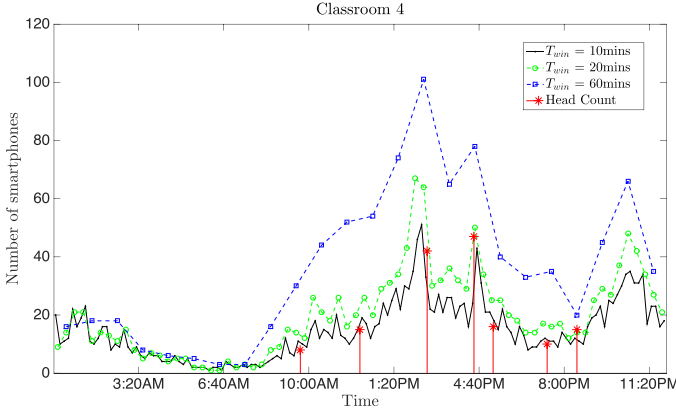


Fig. 12. Number of smartphones detected by *SenseFlow* compares with ground truth. The figure shows the number of smartphones in the Classroom 4 on November 18, 2015 (from 0:00:00A.M. to 11:59:59P.M.).

Specifically, the amount of devices whose connection time to GN-A and GN-C is less than 10 min is about 17 008 and 4462. A few smartphones connect to the two GNs for more than 20 min. This indicates that most of people could just pass by the two GNs. Moreover, there are only 20 and 2 smartphones connecting with GN-A or GN-C for longer than 90 min. Those people could be the staff who works in the nearby shops. The significance of this experiment and result is that the building planner can make a more efficient plan based on information of the people quantity and flow movement.

3) *Classrooms in University*: To monitor people density, eight GNs are deployed in four adjacent classrooms. Classrooms 1 and 2 are in the 3rd floor of the building, and classrooms 3 and 4 are right over 1 and 2 in the 4th floor. The GNs are closely located, as shown in Fig. 11.

Fig. 12 shows the number of unique smartphones in the Classroom 4 for one day (November 18, 2015) when the data analyzing time window T_{win} is 10 min, 20 min, or 60 min. Namely, we calculate the number of unique smartphones every T_{win} .

It is observed that the number of smartphones from daytime to midnight is generally more than the ones before dawn. In this experiment, we personally visit the classrooms at different time to count the number of students. We use these recorded head counts to compare with *SenseFlow* when $T_{win} = 10$ min since the ground truth is based on the discrete time point. Generally, the difference between *SenseFlow* and ground truth is less than 2. It is also observed that *SenseFlow* with $T_{win} = 10$ min undercounts 6, 7, and 3 people comparing to the ground truth at 2:00P.M.,

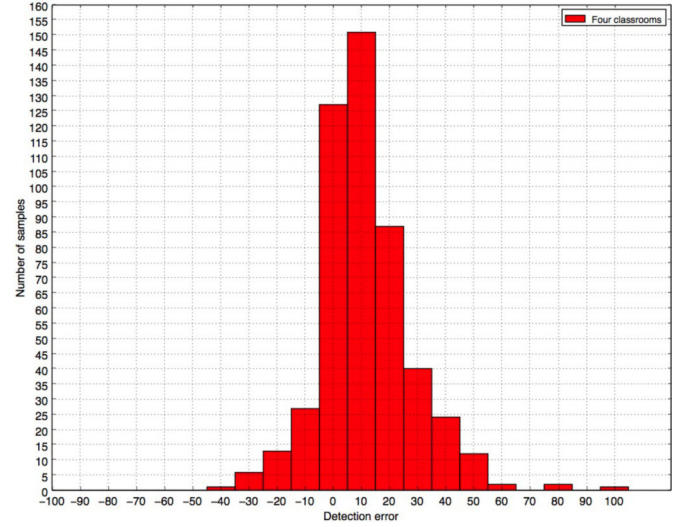


Fig. 13. Detection error of *SenseFlow* in the four classrooms. The figure shows the performance for 33 d.

4:30P.M., and 8:30P.M.. The reason is that there is an event being observed in the classroom where some students switch OFF their mobile devices. Additionally, it is observed from $T_{win} = 60$ min that around 100 different smartphones are in the classroom from 1:00P.M. to 2:00P.M., which indicates the peak hour in the Classroom 4.

The data of *SenseFlow* are collected over 33 d in total, and we record 130 samples for the ground truth. To compare the number of mobile phones detected by *SenseFlow* with the actual number of people in classrooms, we define system detection error at a specific time as follows:

Detection error

$$= \frac{(\text{The phones detected by SenseFlow}) - (\text{ground truth})}{\text{ground truth}} \quad (1)$$

where the data of *SenseFlow* are based on $T_{win} = 10$ min, which is comparable to the ground truth. Moreover, we select the data to compare given the non-zero ground truth value. Therefore, we have three following possible results of detection error.

- 1) The detection error > 0 : The number of phones detected by *SenseFlow* is more than the number of people in the four classrooms.
- 2) The detection error $= 0$: The number of phones detected by *SenseFlow* is the same as the number of people in the four classrooms.
- 3) The detection error < 0 : The number of phones detected by *SenseFlow* is lower than the number of people in the four classrooms.

Fig. 13 shows the histogram of detection error over the four classrooms. As observed, 127 samples are as accurate as the ground truth, 151 samples have 10% detection error, and 87 samples have 20% detection error.

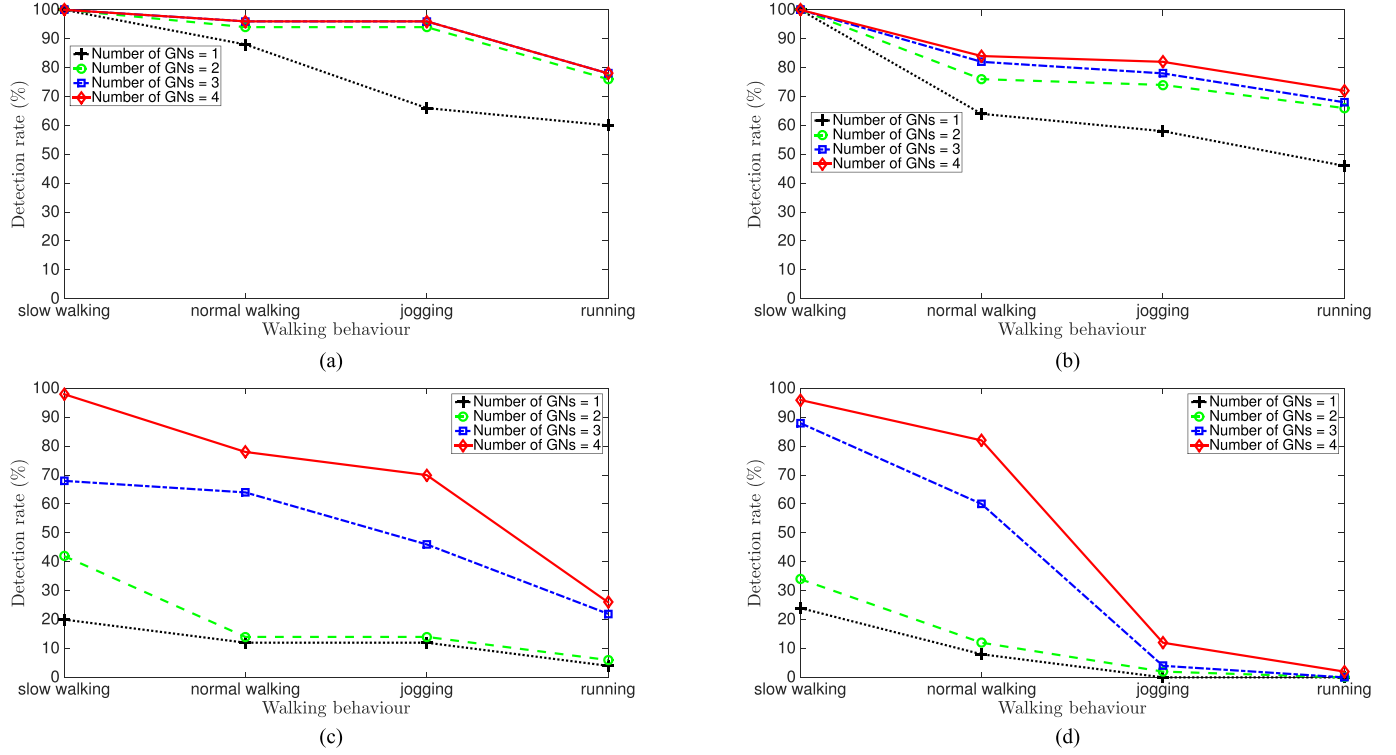


Fig. 14. Detection rate of the smartphones under different human walking behaviors and operational modes. (a) Screen ON, Wi-Fi non-registered. (b) Screen OFF, Wi-Fi non-registered. (c) Screen ON, Wi-Fi registered. (d) Screen OFF, Wi-Fi registered.

D. Human Walking Behavior Effect

From Table II, it is observed that a longer contact time between the GN and smartphone can increase the probe request receiving probability. Namely, the slower the people move, the higher chance that the GN can capture the probe requests from the smartphone. Therefore, we further measure the effect of people walking behavior on *SenseFlow*.

In this experiment, the GNs are deployed in a monitoring area to capture the user's presence. One person carrying five smartphones, i.e., two iOS phones, two Android phones, and one Windows phone, goes through the area with different walking speed. We employ four human walking behaviors, *slow walking*, *normal walking*, *jogging*, and *running* with referred speeds of 1.25, 2.25, 2.6, and 4.5 m/s, respectively. The four operational modes (described in Section IV-A) are also considered. In each experiment, we increase the number of GNs from 1 to 4. For each operation mode and walking behavior, we repeat the experiment for ten times, therefore, detection rate is equal to the average number of smartphones that are detected by any one of GNs.

Fig. 14 shows the performance of detection rate under different operational modes and walking behaviors. Generally, the detection rate grows with an increase in the number of GNs. However, with an increase of walking speed, the detection rate of smartphones decreases. Because the probe request is unable to be detected when its interval is longer than the contact time of smartphone and the GN. Moreover, *SenseFlow* achieves the highest detection rate when the smartphone is in (screen ON, Wi-Fi non-registered) mode. The reason is the smartphone's

energy-saving design prolongs the probe request interval when the screen is turned OFF or Wi-Fi is connected.

E. People Flow Tracking Experiments

We evaluate the performance of *SenseFlow* by deploying a proof of concept testbed on the SUTD University campus. The testbed consists of 14 GNs that are evenly deployed at seven locations in four buildings of the university so that the probe requests of the mobile device are not overheard at adjacent locations. Each location contains two GNs that are 1 m away from each other. The reason is to increase the chance of probe request reception especially when one of the GNs malfunctions. The location sequence is from 1 to 7. Three people move from location 1 to 7 at normal walking speed and each of them carries one smartphone. The smartphones in this experiment are one iOS, one Android, and one Windows phone. We choose two targeting trajectories from the same starting point (location 1) to the same destination (location 7), $\vec{\mathcal{J}}_1 = (1, 2, 3, 4, 5, 6, 7)$ and $\vec{\mathcal{J}}_2 = (1, 2, 5, 6, 4, 3, 7)$. The starting and ending time are denoted as t_1 and t_7 , respectively.

Given the trajectory of smartphone i is $\vec{\mathcal{X}}_i$ and the length of $\vec{\mathcal{X}}_i$ is presented by $\ell(\vec{\mathcal{X}}_i)$. Moreover, the number of locations successfully detects the smartphone is $\ell(\vec{\mathcal{X}}'_i)$. Therefore, the tracking accuracy can be given by

$$\delta = \frac{\ell(\vec{\mathcal{X}}'_i)}{\ell(\vec{\mathcal{X}}_i)}. \quad (2)$$

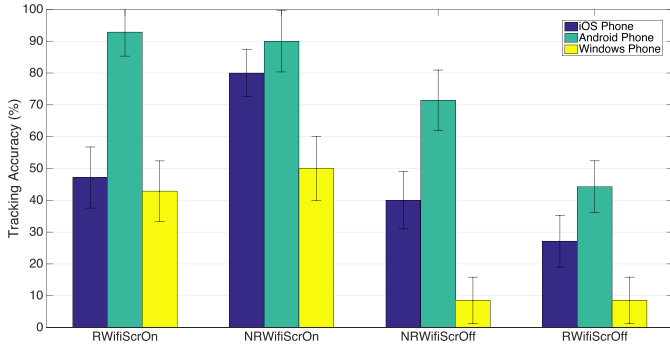


Fig. 15. Tracking accuracy of iOS, Android, and Windows phones in four modes, (screen ON, Wi-Fi registered), (screen ON, Wi-Fi non-registered), (screen OFF, Wi-Fi non-registered), and (screen OFF, Wi-Fi registered). The confidence interval is based on 10 experiments.

Fig. 15 shows the performance of *SenseFlow* with different smartphone models and operational modes. In RWifiScrOn (screen ON, Wi-Fi registered), Android phone achieves the highest δ , about 92.9%. iOS and Windows phone achieve $\delta = 47.2\%$ and $\delta = 42.9\%$ on average. In NRWifiScrOn (screen ON, Wi-Fi non-registered), the average δ of Android, iOS, and Windows phone is 90%, 80%, and 50%, respectively. The tracking accuracy of iOS and Windows phone in this case is higher than the one in RWifiScrOn. The reason is that when the Wi-Fi of the smartphones is not connected, they broadcast probe requests more frequently in order to search the Wi-Fi connection. NRWifiScrOff (screen OFF, Wi-Fi non-registered) and RWifiScrOff (screen OFF, Wi-Fi registered) present the walking trajectories tracked by *SenseFlow* when the screen of smartphone is OFF. Specifically, the δ of Android phone is 71.4% and iOS is 40% in NRWifiScrOff. The δ of Android and iOS go down to 44.3% and 27.2% in RWifiScrOff. This is because the smartphone increases the probe request interval when Wi-Fi is connected. Moreover, in both experiments, the trajectory of Windows phone is not tracked since its wireless transceiver is turned OFF when its screen is OFF.

The recognition of the two targeting trajectories $\vec{\mathcal{T}}_1$ and $\vec{\mathcal{T}}_2$ depends on $\vec{\mathcal{T}}_1^* = (3, 4, 5, 6)$ and $\vec{\mathcal{T}}_2^* = (5, 6, 4, 3)$. Furthermore, since the locations 5 and 6 are in the same order in $\vec{\mathcal{T}}_1$ and $\vec{\mathcal{T}}_2$, for simplicity, we denote them as one combined location (5, 6), then we have, $\vec{\mathcal{T}}_1^* = (3, 4, (5, 6))$ and $\vec{\mathcal{T}}_2^* = ((5, 6), 4, 3)$.

To evaluate the trajectories recognition of *SenseFlow*, we define successful trajectory recognition rate, which is a ratio of number of phones on the targeting trajectory and the total number of phones. In this experiment, five people walk from location 1 to location 7 along the trajectories $\vec{\mathcal{T}}_1$ and $\vec{\mathcal{T}}_2$, and each of them carries one smartphone. The types of smartphone we use include two Android phones, two iPhones, and one Windows phone. We find that the highest successful recognition rate of the two targeting trajectories, $\vec{\mathcal{T}}_1^*$ and $\vec{\mathcal{T}}_2^*$, is 100% when the phones are in the mode, (screen ON, Wi-Fi non-registered). The reason is the smartphone transmits probe requests frequently in this operational mode. The GNs at the locations in $\vec{\mathcal{T}}_1^*$ and $\vec{\mathcal{T}}_2^*$ detect the smartphone successfully. In (screen

OFF, Wi-Fi non-registered) mode, *SenseFlow* recognizes 40% of trajectories on $\vec{\mathcal{T}}_2$, and 60% of trajectories on $\vec{\mathcal{T}}_1$. In (screen ON, Wi-Fi registered) mode, the recognition rate achieved by *SenseFlow* is 60%. However, for the phones with (screen OFF, Wi-Fi registered), the trajectories are hardly recognized, the recognition rate is only 20%. The reason is the probe request interval is increased when the Wi-Fi connection of smartphone is set up and the screen is locked (shown in Table II).

V. CONCLUSION

In this paper, we have investigated *SenseFlow* system to monitor people density and track people flow in smart cities by using wireless sensors to passively collect probe requests from people's smartphones. We explored the lightweight sensing system architecture and data collection scheme based on the probe request interval. In *SenseFlow*, the probe request overhearing problem is addressed to improve people tracking performance. We measured the effect of smartphone's operational modes and human walking behavior, and evaluated the tracking accuracy of *SenseFlow* in four typical application scenarios.

ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous reviewers for their constructive comments on this paper. This paper features a complete system and data collection architecture, and people tracking experiments.

REFERENCES

- [1] K. Li, C. Yuen, and S. Kanhere, "SenseFlow: An experimental study of people tracking," in *Proc. 6th ACM Workshop Real World Wireless Sensor Netw.*, 2015, pp. 31–34.
- [2] G. Cardone, A. Cirri, A. Corradi, and L. Foschini, "The participact mobile crowd sensing living lab: The testbed for smart cities," *IEEE Commun. Mag.*, vol. 52, no. 10, pp. 78–85, Oct. 2014.
- [3] T. Teixeira, G. Dublon, and A. Savvides, "A survey of human-sensing: Methods for detecting presence, count, location, track, and identity," *ACM Comput. Surv.*, vol. 5, pp. 59–69, 2010.
- [4] A. Petrenko, S. Bell, K. Stanley, W. Qian, A. Sizo, and D. Knowles, "Human spatial behavior, sensor informatics, and disaggregate data," in *Proc. 11th Int. Conf. Spatial Inf. Theory*, 2013, pp. 224–242.
- [5] D. Doran, S. Gokhale, and A. Dagnino, "Human sensing for smart cities," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, 2013, pp. 1323–1330.
- [6] M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Troster, "Mobile sensing of pedestrian flocks in indoor environments using WiFi signals," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2012, pp. 95–102.
- [7] C. Xu, B. Firmer, Y. Zhang, R. Howard, and J. Li, "Exploiting human mobility trajectory information in indoor device-free passive tracking," in *Proc. IEEE/ACM 11th Int. Conf. Inf. Process. Sensor Netw.*, 2012, pp. 121–122.
- [8] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas, "EmotionSense: A mobile phones based adaptive platform for experimental social psychology research," in *Proc. 12th ACM Int. Conf. Ubiquitous Comput.*, 2010, pp. 281–290.
- [9] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried, "Evacuation dynamics: Empirical results, modeling and applications," in *Extreme Environmental Events*. New York, NY, USA: Springer, 2011, pp. 517–550.
- [10] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 140–150, Sep. 2010.

- [11] B. P. L. Lau, T. Chaturvedi, B. K. K. Ng, K. Li, M. S. Hasala, and C. Yuen, "Spatial and temporal analysis of urban space utilization with renewable wireless sensor network," in *Proc. IEEE/ACM Int. Conf. Big Data Comput. Appl. Technol.*, 2016, pp. 133–142.
- [12] K. Fujinami, Y. Xue, S. Murata, and S. Hosokawa, "A human-probe system that considers on-body position of a mobile phone with sensors," in *Proc. Distrib., Ambient, Pervasive Interact.*, 2013, pp. 99–108.
- [13] A. Musa and J. Eriksson, "Tracking unmodified smartphones using wi-fi monitors," in *Proc. 10th ACM Conf. Embedded Netw. Sensor Syst.*, 2012, pp. 281–294.
- [14] K. Li *et al.*, "Understanding crowd density with a smartphone sensing system," in *Proc. IEEE World Forum Internet Things*, 2018, pp. 517–522.
- [15] G. Deak, K. Curran, and J. Condell, "A survey of active and passive indoor localisation systems," *Comput. Commun.*, vol. 35, no. 16, pp. 1939–1954, 2012.
- [16] L. Chen, H. Wei, and J. Ferryman, "A survey of human motion analysis using depth imagery," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 1995–2006, 2013.
- [17] A. Fernández-Caballero, J. C. Castillo, J. Martínez-Cantos, and R. Martínez-Tomás, "Optical flow or image subtraction in human detection from infrared camera on mobile robot," *Robot. Auton. Syst.*, vol. 58, no. 12, pp. 1273–1281, 2010.
- [18] M. A. Guvensan and A. G. Yavuz, "On coverage issues in directional sensor networks: A survey," *Ad Hoc Netw.*, vol. 9, no. 7, pp. 1238–1255, 2011.
- [19] O. M. Mozos, R. Kurazume, and T. Hasegawa, "Multi-part people detection using 2D range data," *Int. J. Social Robot.*, vol. 2, no. 1, pp. 31–40, 2010.
- [20] C. Premebeda, O. Ludwig, and U. Nunes, "Exploiting LIDAR-based features on pedestrian detection in urban scenarios," in *Proc. 12th Int. Conf. Intell. Transp. Syst.*, 2009, pp. 1–6.
- [21] M. J. Beal, N. Jojic, and H. Attias, "A graphical model for audiovisual object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 828–836, Jul. 2003.
- [22] N. Hu, G. Engleblenne, and B. J. Kröse, "Bayesian fusion of ceiling mounted camera and laser range finder on a mobile robot for people detection and localization," in *Human Behavior Understanding*. Berlin, Germany: Springer, 2012, pp. 41–51.
- [23] Q. Zhai *et al.*, "VM-tracking: Visual-motion sensing integration for real-time human tracking," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 711–719.
- [24] S. Sigg *et al.*, "Passive, device-free recognition on your mobile phone: Tools, features and a case study," in *Proc. Mobile Ubiquitous Syst., Comput., Netw., Serv.*, 2014, pp. 435–446.
- [25] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 261–272.
- [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 *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2014, pp. 130–138.
- [27] L. Schauer, M. Werner, and P. Marcus, "Estimating crowd densities and pedestrian flows using Wi-Fi and bluetooth," in *Proc. 11th Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Serv.*, 2014, pp. 171–177.
- [28] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa, "Signals from the crowd: Uncovering social relationships through smartphone probes," in *Proc. Conf. Internet Meas. Conf.*, 2013, pp. 265–276.
- [29] J. Freudiger, "How talkative is your mobile device?: An experimental study of Wi-Fi probe requests," in *Proc. 8th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, 2015, Art. no. 8.
- [30] T. Chaturvedi, K. Li, C. Yuen, A. Sharma, L. Dai, and M. Zhang, "On the design of MAC protocol and transmission scheduling for Internet of Things," in *Proc. IEEE Region 10 Conf.*, 2016, pp. 2000–2003.
- [31] D. S. Hirschberg, "Algorithms for the longest common subsequence problem," *J. ACM*, vol. 24, no. 4, pp. 664–675, 1977.
- [32] L. Demir, "Wi-Fi tracking: What about privacy," Ph.D. dissertation, M2 SCCI Security, Cryptology and Coding of Information-UFR IMAG, Grenoble Univ., Grenoble, France, 2013.
- [33] J. Little and B. O'Brien, "A technical review of Cisco's wi-fi-based location analytics," white paper, 2013.
- [34] "Meshlium xtreme," Jun. 2018. [Online]. Available: <http://www.libelium.com/products/meshlium>



Kai Li (S'09–M'14) received the B.E. degree from Shandong University, Jinan, China, in 2009, the M.S. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2010, and the Ph.D. degree in computer science from the University of New South Wales, Sydney, NSW, Australia, in 2014.

He is currently a Research Scientist and Project Leader with the Real-Time and Embedded Computing Systems Research Centre (CISTER), Porto, Portugal. From 2014 to 2016, he was a Postdoctoral Research Fellow with the SUTD-MIT International Design Centre, Singapore University of Technology and Design, Singapore. From 2012 to 2013, he was a Visiting Research Assistant with ICT Centre, CSIRO, Australia. From 2010 to 2011, he was a Research Assistant with the Mobile Technologies Centre, The Chinese University of Hong Kong. His research interests include vehicular communications and security, resource allocation optimization, cyber-physical systems, Internet of Things, human sensing systems, wireless sensor networks, and UAV networks.



Chau Yuen (S'02–M'08–SM'12) received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2000 and 2004, respectively.

He was a Postdoc Fellow with Lucent Technologies Bell Labs, Murray Hill, in 2005. He was a Visiting Assistant Professor with Hong Kong Polytechnic University in 2008. From 2006 to 2010, he was with the Institute for Infocomm Research (Singapore) as a Senior Research Engineer. He has been with the Singapore University of Technology and Design as an Assistant Professor since June 2010.

Dr. Yuen serves as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and awarded Top Associate Editor for three consecutive years. He was the recipient of the IEEE Asia-Pacific Outstanding Young Researcher Award in 2012. He has held positions on several conference organizing committees, and is on technical program committees of various international conferences.



Salil S. Kanhere (M'01–SM'11) received the M.S. and Ph.D. degrees in electrical engineering from the Drexel University, Philadelphia, Pennsylvania, in 2001 and 2003, respectively.

He is currently an Associate Professor with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW, Australia. He is currently a contributing research staff with the National ICT Australia, Sydney, NSW, Australia, and a Faculty Associate with the Institute for Infocomm Research, Singapore. His current research interests include pervasive computing, crowdsourcing, embedded sensor networks, mobile networking, privacy, and security. He has authored/coauthored more than 140 peer-reviewed articles and delivered more than 15 tutorials and keynote talks on these research topics.

Dr. Kanhere regularly serves on the organizing committee of a number of IEEE and ACM international conferences (e.g., IEEE PerCom, ACM MobiSys, ACM SenSys, ACM CoNext, IEEE WoWMoM, IEEE LCN, ACM MSWiM, IEEE DCSS, IEEE SenseApp, ICDCN, and ISSNIP). He currently serves as the area editor for the *Pervasive and Mobile Computing*, *Computer Communications*, *International Journal of Ad Hoc and Ubiquitous Computing*, and *Mobile Information Systems*. He received the Humboldt Research Fellowship in 2014.



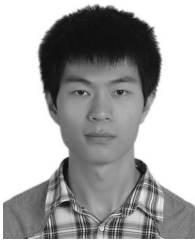
Kun Hu received the B.S. degree in computer science and technology from Jilin University, Changchun, China, in 2012, and the M.S. degree in software engineering from Peking University, Beijing, China, in 2015.

He was a Visiting Student Researcher with the Singapore University of Technology and Design in 2014. His research interests include Internet of Things, pervasive computing, and smart home.



Fan Jiang received the bachelor's degree in engineering from the China University of Geosciences, Wuhan, China, in 2014, and is currently working toward the master's degree at Peking University, Beijing, China.

He is currently a Visiting Researcher with the Singapore University of Technology and Design, Singapore. He was a software development intern with the Intel Corporation, from 2014 to 2015. His research interests include IoT system, smart city, and data visualization.



Wei Zhang received the bachelor's degree from Beijing Jiaotong University, Beijing, China, in 2012, and the master's degree from Peking University, Beijing, China, in 2015. He studied digital media technology from 2010 to 2012, mainly focuses on three-dimensional render and game engine.

In 2015, he joined the High Performance Embedded Computing Lab held by Dr. X. Liu, studied smart space and inference engine, it lasts for three years. During his master's degree, he visited the Singapore University of Technology and Design as a Research

Assistant to work for Dr. C. Yuen on smart city. He joined Schlumberger in July 2015 as a Software Engineer.



Xiang Liu received the B.Eng. degree in electrical engineering from Hunan University, Changsha, China, in 1984, the Ms. Eng. degree in pattern recognition and intelligent control from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1987, and the Ph.D. degree in automation from Bordeaux University 1, Bordeaux, France, in 1993.

In 1994, he joined the Global Software Group, Motorola. In 2001, he had been elected as a member of the Motorola Science Advisory Board Associates for his technical contribution. In March 2003, he joined the School of Software and Microelectronics, Peking University as a Professor and the Chair of Department of Embedded System Engineering. His current research interests include high-performance embedded computing, ubiquitous computing, and software engineering.