MobiCamp: a Campus-wide Testbed for Studying Mobile Physical Activities

Mengyu Zhou[†][*], Kaixin Sui[‡], Minghua Ma[‡],
Youjian Zhao[‡], Dan Pei[‡], Thomas Moscibroda[§]
[†]Institute for Interdisciplinary Information Sciences, Tsinghua University
[‡]Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University
[§]Microsoft Research

ABSTRACT

Ubiquitous WiFi infrastructure and smart phones offer a great opportunity to study physical activities. In this paper, we present MobiCamp, a large-scale testbed for studying mobility-related activities of residents on a campus. MobiCamp consists of $\sim 2,700$ APs, $\sim 95,000$ smart phones, and an App with $\sim 2,300$ opt-in volunteer users. More specifically, we capture how mobile users interact with different types of buildings, with other users, and with classroom courses, *etc.* To achieve this goal, we first obtain a relatively complete coverage of the users' mobility traces by utilizing four types of information from SNMP and by relaxing the location granularity to roughly at the room level. Then the popular App provides user attributes (grade, gender, *etc.*) and fine-grained behavior information (phone usages, course timetables, *etc.*) of the sampled population. These detailed mobile data is then correlated with the mobility traces from the SNMP to estimate the entire campus population's physical activities. We use two applications to show the power of MobiCamp.

1. INTRODUCTION

People spend a significant part of their daily lives performing a variety of activities in the physical world at various locations and interacting with other people. These activities and interactions contain a wealth of information about user behavior, preferences, attitudes and interests, that, if harnessed, can benefit both users and consumer-facing businesses. The enterprise WiFi infrastructure and ubiquitous smart phones offer a great opportunity to study people's physical activities and interactions. In particular, there have been a long line of researches [15, 22, 14] that focus on university campus, where students study/work/live with good enterprise WiFi network (WLAN) coverage.

In this paper, we present MobiCamp, a large-scale testbed for studying people's mobility-related physical activities on Tsinghua campus. MobiCamp consists of $\sim 2,700$ APs, $\sim 95,000$ smart phones, and a popular App with $\sim 2,300$ opt-in volunteer users. More specifically, we capture how the entire population of mobile users interact with different types of buildings, with other users, and with classroom classes, *etc.* While previous work studied mobility using WLAN data [15, 13] or a small set of mobile data [22], MobiCamp uses the *combination* of enterprise WLAN data and mobile App user data *both at a large scale*. This key difference from previous work enables MobiCamp to estimate the entire campus population's physical activities for the first time. This is achieved by correlating the mobility traces from the SNMP with the sampled but detailed behavior data from App, including user attributes (grade, gender, *etc.*) and fine-grained behavior information (such as phone usages).

To achieve the above goal, the first major challenge is to obtain mobility traces with sufficient accuracy for our studies. First, traditional WiFi traces, including connectivity and RSSI of associated devices, can lead to bad estimation and missing pieces when devices are not yet associated to WLAN [6]. To improve coverage on time, space and population, we utilize four types of SNMP data from operational enterprise WLANs (see §2.1). Second, indoor positioning techniques are usually unavailable, immature or too costly to deploy in lots of WLANs. For scenarios that only require granularity at room-level, we propose a simple mobility detection algorithm that does not depend on manual location labelling of APs (*e.g.*, 3D coordinates, fingerprinting). Third, dirty noises can easily be introduced by MAC address spoofing that becomes emerging phenomenon in smart phones.

The second major challenge is how to link mobile data with mobility traces. A common practice is to match hardware identifiers with user identities. However, nowadays more and more mobile devices, including both Android and iOS, are blocking the access to hardware identifiers from mobile Apps. To tackle the problem, we implement an API for our Apps to query MAC addresses from the SNMP of enterprise WLAN.

With the large-scale mobility traces and the successful linking of rich and detailed mobile data, in §4 we further show two applications enabled by MobiCamp: 1) characterizing people's activities among heterogeneous buildings, and 2) social tie strengths *v.s.* distractions during people's co-location. We also sketch future work including optimization of network performance with device mobility, and educational measurements of courses on campus.

2. DATA COLLECTION

The campus of Tsinghua covers an area of $\sim 4.4km^2$ on which $\sim 45,000$ students and $\sim 12,000$ faculty and staff members are living. The ubiquitous enterprise WLAN on the campus allows us to track the devices of the large resident population. Meanwhile, popular App TUNet also allows us to learn from rich mobile data about detailed behaviors.

[*]Contact email: mengyu(dot)chou(at)gmail(dot)com.

WPA'16, June 26, 2016, Singapore.
© 2016 ACM. ISBN 978-1-4503-4328-2/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2935651.2935654

Table 1: Three Types of Packet Signal Strength Records.

RSSI type	description	timestamp
probe packet	RSSI of last scan probe request packet from a device heard by an AP	last heard
rogue packet	RSSI of last data packet sent from the device to rogue APs heard by an AP	last heard
connected	Average RSSI of data packets from the device to its associated AP	polling time

2.1 Data Collection on WLANs

Until January 2016, there are $2,786$ Cisco enterprise APs deployed in 114 buildings (including classroom, department, administrative buildings, apartments, gyms, libraries, restaurants, supermarkets, hotels, *etc.*) on the campus, providing dense deployment in most of the areas. At peak, there are $\sim$20, 000 devices concurrently connected to the campus WLAN. The total number of unique devices surpasses $60,000$ each day, which means on average everyone uses at least one wireless device. In MobiCamp, mobility traces of all client devices are derived from WLAN data provided by network administrators, including association events in SNMP trap messages, and device packet signal strengths derived from SNMP (Simple Network Management Protocol) object values. With the fast expansion of WiFi infrastructures, these data are readily available at the wireless controllers of most vendors [19, 20].

Device association events: SNMP trap messages report the AP association process of client devices[1], including `association`, `associated`, `disassociation`, `deauthenticated` and `disassociated` status, to our data collection server in real-time. Previous studies such as [6] have shown that the raw AP association trace $alone$ is a bad estimator of device location and mobility because of the following reasons. First, large part of mobility can be unknown if a device does not try to associate with WLAN. When moving at high speed, devices have no time to establish a WiFi connection. Besides, devices may connect to rogue APs[2] or the cellular network rather than the enterprise WLAN. Second, it is inaccurate to treat the associated AP of a device as its location estimator. Various strategies on different devices for AP selection and roaming influence the AP association. Aggressive devices can be wrongly considered moving due to the ping-pong effect of frequent AP switching when pausing. Or vice versa, conservative devices can be wrongly considered pausing due to the lazy roaming strategies of devices when moving.

Device packet signal strengths: Three types of packet signal strength, which can be derived from SNMP object values with MAC addresses of the sender device and the monitor AP, are listed in Table 1 and polled every $\sim 5min$. The device RSSI records in Table 1 allow MobiCamp to track any client device within a large area through long time spans. Firstly, monitoring the RSSI of probe[3] packets only requires the device to turn on WiFi. Some devices even send probe scans when WiFi is not turned on [4]. Secondly, monitoring probe and rogue packets does not require a device to connect to the WLAN. Rogue APs are prevalent [19] — more than $23,000$ rogue APs appeared in range of campus WLAN during one week, which outnumber the enterprise APs by about ten times. Thirdly, probe packets can reach further range than normal data packets because of its lower PHY rate. Fourthly, considering more surrounding APs can also help eliminate the ping-pong effect caused by solely association events. Finally, a conservative device may stop sending active probe scans when it has a stable connection or is in suspend mode. For these cases, the association events, connected and rogue packet RSSI records can fill up the gaps.

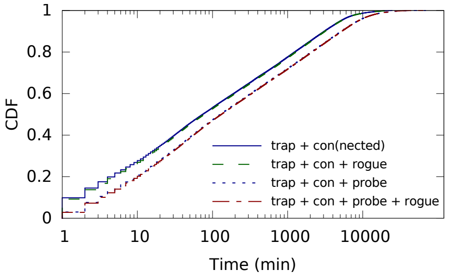


Figure 1: Distribution of Covered Minutes per Device.

As shown in Fig. 1, the minutes (covered by records) increase when adding more types of packet signal strength records. By utilizing probe and rogue packet RSSI, during a 11-week observation period from November 2015 to January 2016, there is a 71.0% boost in covered minutes compared to only utilize connectivity and connected packet RSSI — The minutes with records increase from $228,039,086min$ to $389,942,434min$ in total for all devices. (Adding only rogue packet RSSI yields a 1.7% boost while adding only probe packet RSSI yields a 69.3% boost.)

2.2 MAC Address Spoofing

MAC (Media Access Control) address is a unique identifier of network interfaces for communications at link layer. MAC address is usually assigned by the manufacturer of a NIC (Network Interface Controller) and stored in its hardware, thus providing an identifier for tracking devices. However, a host device can issue commands to the NIC to use an arbitrary address instead of the burned-in one, leading to the problem of MAC spoofing that fake MAC addresses become noises in mobility tracing.

For laptops and desktops, there are tools[4] readily available to change MAC address manually. Meanwhile, on mobile phones, MAC spoofing is relatively rare, but emerging. Currently manual changes of MAC on Android and iOS devices require rooted permission and are rarely being done. But there is a trend for mobile phone operating systems to eliminate unintended exposure of MAC address within probe requests. *E.g.*, as of iOS version 8, Apple changed the way mobile devices (*e.g.*, iPhone 5S and 6) send probe requests. Under certain strict conditions [2], iOS phones will use randomized MAC addresses in probe scan packets.

Both built-in randomization of iOS [2] and Windows [12] follow the standard to use locally administrated MAC addresses[5]. As shown in Table 2, from all the $209,723$ MAC addresses that appeared in the campus WLAN during the 11 weeks, only 486 are locally administrated addresses. Although there are methods to defeat randomization [11] through mobility patterns, network

[1] Some vendors use *syslog* events to report association events [15].

[2] Rogue APs are access points which are not installed by network operators (*e.g.* by students) and not accessible for data collection.

[3] For energy saving concerns, active probe request is widely adopted on mobile devices (in complimentary with passive listening to beacon frames) for AP discovery [10].

[4] Softwares such as Technitium MAC Address Changer [5] and SpoofMAC [1] can quickly change MAC on operation systems such as Windows, Mac OS and Linux. Windows 10 even provides a complex built-in generator of randomized MAC addresses [12].

[5] MAC addresses can either be universally administered (UA) or locally administered (LA). A UA address is uniquely assigned to a device by its manufacturer. The first three octets (in transmission order) identify the organization that issued the identifier and are known as the Organizationally Unique Identifier (OUI). A LA address is assigned to a device by a network administrator or device itself, overriding the burned-in address, and do not contain OUIs. UA and LA addresses are distinguished by setting the second-least-significant bit of the most significant byte of the address.

Table 2: Device Classification and Randomized MAC.

Administrated	Visitor	Stationary	Laptop	Phone	$\sum$
Locally	371	3	20	92	486
Universally	$102,062$	339	$11,555$	$95,281$	$209,237$
$\sum$	$102,433$	342	$11,575$	$95,373$	$209,723$

authentication and deep packet inspection, we consider it as a sign of opt-out and do not track their mobility. In Table 2, one may notice that most of locally administrated addresses are classified as "visitor"s (other three categories will be defined in §3.3). We define a MAC address is representing a "visitor" device if it appeared for < 5 days in 11 weeks. These visitors are also filtered out from mobility tracing for two reasons: 1) MAC spoofing (includes non-built-in methods which can use universally administrated addresses) can generate visitor-like MAC addresses that appear shortly in WLAN, and 2) In MobiCamp we mainly focus on longitudinal studies on residents of the campus.

2.3 Data Collection on Mobile Apps

Until January 2016, our mobile client WLAN tool application **TUNet** (developed by a student group led by authors) is installed on more than $8,600$ Android and $6,500$ iOS client devices. $\sim 2,300$ Android volunteers contribute their mobile data to MobiCamp testbed. Data of pedometer sensor, WiFi scan results and connectivity events are collected and used for cross validation in §3.2. Phone interactive states, demographic data, educational data, *etc.* are also collected for further analyses in §4.

2.4 Hardware Identifiers *v.s.* User Identities

To link WLAN and mobile data, we need to fill the gap between unique hardware identifiers (MAC addresses from WLAN) and user identities such as campus account or student ID from mobile apps (TUNet in MobiCamp testbed). There are two obstacles to fill the gap using mobile apps: 1) More and more Android and iOS devices block the access to hardware identifiers, especially MAC address, for mobile applications [3]. 2) The raw data of each login of an account on a device does not directly yield a one-to-one relation. Multiple accounts can be used duplicately on multiple devices. For the first problem, we develop a web API for devices to query their MAC address through SNMP according to the SSID and BSSID of connected AP (of campus WLAN) and the obtained IP address. For the second problem, we assign each MAC address to the most frequently used account on the device. Thus for a user that obtains new devices during the observation period, old and new devices will all be mapped to the same account.

3. MOBILITY DETECTION

In this section we show how mobility of client devices is derived from WLAN data in MobiCamp testbed.

Measurement of mobility is at the fundamental core of many studies and applications. Event detection [14] and prediction of social links [21] based on co-location, optimization and communication opportunities [17] of wireless networks, *etc.* all depend on observations, assumptions and models of human mobility. With mobility, further observations and applications are then available for other domains. *E.g.*, behaviors of users of location-based social networks [8], correlation between study performance and student mobility [22], legal analyses of the dangers of "guilt by association", healthcare and epidemics [7], *etc.*

The "**mobility**" of a device is defined as the stay interval (start time, end time) and the corresponding appeared location (RSSI fingerprint). A device is considered to stay in a region as long as its RSSI fingerprint remains stable.

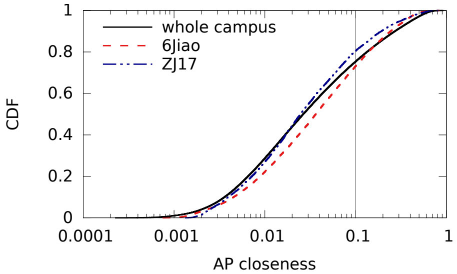


Figure 2: Distributions of AP closeness.

3.1 Derive Mobility from WLAN Data

There are multiple ways to derive mobility traces of devices. However, there is no need to use excessively accurate algorithms such as complex and costly indoor localization methods for lots of tasks, *e.g.*, social networks [8], opportunistic networks [17], content distribution, educational studies, *etc.*. We propose an easy-to-adopt mobility detection algorithm based on relatively dense-deployed APs. Considering that location labels (*e.g.*, 3D coordinates, nearby rooms and manual fingerprinting) of APs are hard to collect and prone to variations, the algorithm does not assume such labels. The algorithm takes the following two steps:

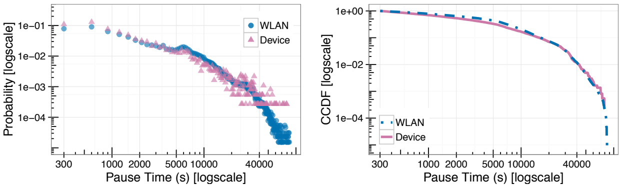
1) To smooth the fluctuations in raw WiFi data, fingerprint snapshots of each device are generated from its packet RSSI records and association events. The snapshots are generated by using a sliding window and continuously kick out deprecated records and records of "far-away" APs.

2) Successive "similar" fingerprint snapshots are merged into large fingerprint to represent the appeared location of the device. The earliest and latest time of records in each merged fingerprint defines the stay interval.

We determine that two APs are "far-away" using an intuitive AP closeness metric based on the fact that nearby APs can hear probe scans of a device at almost the same time. Devices regularly send probe request packets with some intervals or after specific events like wakeup and connectivity changes. We re-construct the *scan group*s of a device as continuous scan packet RSSI records of the device. Based on all historical scan groups SG from the WLAN data, the *closeness* from AP_i to AP_j is defined as $C(AP_i, AP_j) = \frac{|\{sg|AP_i \in sg, AP_j \in sg, sg \in SG\}|}{|\{sg|AP_i \in sg, sg \in SG\}|}$ where $C(AP_i, AP_j) = 0$ if two APs almost never appear in the same scan groups. The numerator is the number of scan groups in the range of common accessible area of AP_i and AP_j, and the denominator is the number of scan groups appearing in the range of AP_i. $C(AP_i, AP_j)$ approximates the probability that AP_j can see a device given that AP_i has already seen it. Note that the closeness maybe asymmetric when AP_i and AP_j have different size of accessible areas. The higher $C(AP_i, AP_j)$ is, the less movements are needed for a device to transit from the range of AP_i to AP_j. Therefore we further define AP_j is "**close**" to AP_i if $C(AP_i, AP_j)$ is large enough (a threshold is chosen based on the overall distribution) that a inter-region movement is not likely to happen. Further, two RSSI fingerprints[6] are considered "similar" if most APs in the records are "close" to each other and the fluctuation on RSSI values of each AP is limited.

There are several parameters — such as sliding-window length, AP closeness threshold and fingerprint similarity percentage — which need to be adapted to the environment. In MobiCamp

[6] A fingerprint of a device is a set of APs, corresponding packet RSSI (see Table 1) and heard time: $\{(AP_i, RSSI_i, t_i)\}$. A device connectivity event can be converted into the form of packet RSSI record, taking the nearest packet RSSI.



(a) Probability (bin width = 300) (b) CCDF

Figure 3: Pause Time Distribution Comparison.

testbed, we choose the sample rate of $5min$ as the sliding-window length to avoid fluctuations. As shown in Fig. 2, first quartile of AP closeness ~ 0.1 is set as the threshold. For fingerprint comparison, we consider two fingerprints are similar only when $\geq 75\%$ pairs of APs are close. We enumerated several reasonable values for the parameters and found no major qualitative differences in §4.

Note that the granularity of the mobility detection depends on the density of APs. At Tsinghua, on average a client device can be heard by 3.59 APs at the same time, which means that there are enough APs to continuously monitor the location of each device. For most of the buildings on the campus, APs divide the indoor space into room-level regions with finer than $10 \sim 17m$ (diameter) granularity. As a consequence, for scenarios that only require granularity at room-level, it is more practicable to adopt our simple algorithm which does not depend on location labelling of APs.

Finally, in some analyses we need to distinguish whether a device is paused or in-transit. We consider stay intervals $\geq 5min$ as "pause" modes and others as "transient" modes.

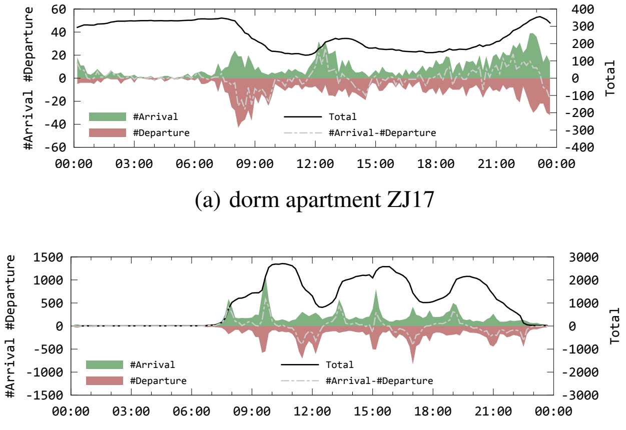
3.2 Evaluation and Cross Validation

For the evaluation of the mobility detection algorithm, we compare the pause intervals derived from WLAN data with that from mobile data and manual mobility logs.

Comparison with mobility derived from mobile data: Pause intervals of TUNet (see §2.3) volunteer devices are also derived by the following steps. Firstly, the step counter sensor provides accurate ground truth of whether the device is carried by a walking human, *i.e.* whether it is in "transient" mode or "pause" mode. Secondly, with the help of scan results (each scan result record includes BSSID, channel frequency, RSSI and the seen time of a detected AP) reported by Android, rare miss-classification cases such as short pauses in elevators and running on treadmills are calibrated by looking at RSSI fluctuations.

We compare the pause modes derived from WLAN data with that derived from mobile data of same devices in Fig. 3 where probability and CCDF of pause time are plotted. It shows that the detected pause time distribution from WLAN and that from devices match with each other.

Comparison with manual logs: We further compare the pause intervals derived from both WLAN data and mobile data with manually collected daily pause modes logs from 5 devices. For all seconds with known results, pause modes derived from both WLAN data and mobile data show high accuracy (84.5% and 92.0%) and low conflict (0.73% and 1.31%) with manual logs. We find that mobile data generates more transient intervals due to its sensitive step counter. WLAN data sometimes loses traces when a device is outside the WLAN signal range, especially outdoor.



(a) dorm apartment ZJ17

(b) classroom building 6Jiao

Figure 4: Flows in different buildings.

3.3 Laptops *v.s.* Phones

To approximate the indoor human mobility, mobile phones are better tracking targets rather than laptops. We roughly categorize non-visitor devices (visitors are filtered out in §2.2) into three types — mobile (phone and laptop), and stationary. The stationary devices that are located at only one location stablely ($\geq 7hrs$) for most (60%) days are filtered out. A mobile device is likely to be a laptop if it moves between places but shows no intermediate transient locations [13]. Thus if a mobile device is never in transient mode for most (60%) appeared days, we consider it as a laptop. Classification results are shown in Table 2. From all the $107,290$ non-visitor devices, 342 were stationary devices, $11,575$ were laptops and $95,373$ were phones. The misclassification of TUNet client phones as laptops was only 1.09%.

4. MOBICAMP APPLICATIONS

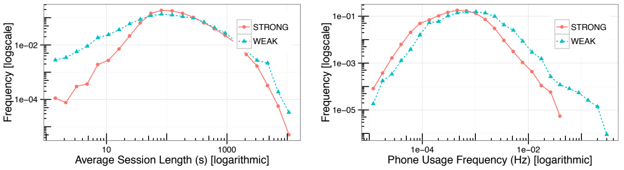
In this section, we demonstrate how multiple studies are made possible by MobiCamp testbed. Not only the large-scale mobility traces on the diverse campus town is interesting, but also the rich and detailed mobile data help us to better understand how people use their mobile phones, how they interact with each other, how students act in their courses, *etc.*

4.1 Activities among Heterogeneous Buildings

The diverse set of 114 buildings in MobiCamp allows us to investigate how human physical activities differ among buildings. *E.g.*, in Fig. 4, flows of arrival, departure and total number of devices (derived from mobility traces) on a weekday in ZJ17 and 6Jiao are shown. People aggregate in apartments (*e.g.* ZJ17) during noon and night. Students attend classes during the five sessions of a day in classroom buildings (*e.g.* 6Jiao). Buildings at Tsinghua are manually categorized into 8 major types: administrative, cafeteria, classroom, department, dorm, gym, library and others.

To further understand similarity and differences of activities among buildings, we try to cluster buildings based on the similarities between their flows. To compare two time series of flow, we calculate cross correlation between them with lag $\leq 30min$. Cross correlation (also known as a sliding dot product or sliding inner-product) is widely used in signal processing as a measure of similarity of two series as a function of the lag of one relative to the other. It can be defined as $\rho_{XY}(\tau) = E[(X_t - \mu_X)(Y_{t+\tau} - \mu_Y)]/(\sigma_X \sigma_Y)$ for series X and Y (μ is mean and σ is standard deviation). We define the flow similarity of two buildings as the summation of maximum cross correlation (where lag $|\tau| \leq 30min$) on each type of flow. Then we apply a hierarchical clustering on all the buildings.

By further looking at the clustering results in Fig. 6 (building types are in parentheses), we find that flow similarity can be a good metric to compare building usages. *E.g.*, all dorms are clustered



(a) Length of All Sessions (b) Frequency of Short Sessions

Figure 5: Phone Usage *v.s.* Strong/weak Tie.

together cleanly since they have distinct flows which are almost complementary with other work and study areas opened during the day. The slight differences between ZJ14-18 and ZJ19-23 apartments reflects the different lifestyles of their residents — PhD students live in ZJ14-18, while visiting and foreign students live in ZJ19-23. Similarly, all cafeterias are nicely clustered together by their distinct bursts during foraging time. Most frequently used classrooms, namely 3Jiao, 4Jiao, 5Jiao and 6Jiao also share similar flows. These results provide a new way to look at human interactions with buildings, and shed lights on modelling of new buildings from similar old ones.

4.2 Distractions during Co-location

Co-location history can tell a lot about social ties [9, 21] and group events [14]. Based on the mobility traces, we characterize the social-physical relationships using co-location networks. Each edge in the network is attached with contact intensity weights to present the strength of companionship. By comparing social tie strengths with mobile phone usages (derived from phone interactive and doze/asleep states collected from TUNet), we find that *people tend to interact with phones frequently and shortly when they stay with weak tie friends*. In Fig. 5(a) it seems that whether the tie is strong or weak shows major differences at short sessions, specifically those under $\sim$30s length. In Fig. 5(b) of only short sessions (with length $\leq 30s$), we can see a clear shift between the log-normal-like phone usage frequency distributions of strong and weak ties. Under weak ties, people tend to use phones in a bursty way, indicating more distractions. In other words, the burstiness of phone usage can be an indicator to the strengths of social-physical relationships of co-located people.

Other social-related topics, such as location-based social services [9, 8], prediction of new social links [21], *etc.* can also be better understood through co-location networks.

4.3 Other On-going Projects

Network performance: In our WifiSeer project [20], WLAN data, active ping and TUNet app are used to characterize WiFi latency and guide the devices to associate to APs with low latency when they are paused. Our past approach did not consider the mobility patterns of each device. Better AP association and switching mechanisms can be implemented based on the next place location prediction. Other network related topics can also be studied more in MobiCamp testbed. *E.g.*, WLAN-first building design, contact opportunities for wireless communications, privacy and uniqueness in the crowd, *etc.*

Educational Measurements: Classroom based education is hard to measure at large scale. Based on mobility traces and mobile data (including course timetables and final scores from volunteers) of MobiCamp testbed, we are able to measure multiple course metrics for near 800 courses. On the other hand, students' physical activities highly correlates with their study performances [22]. The activity patterns and demographic attributes of the large group of mobile users also help us better understand what characterize a good student and related factors for performance prediction.

5. RELATED WORK

There are generally three ways to collect device traces [6]: *monitoring location* (device-based localization using GPS, RFID based, Bluetooth, GSM and 802.11 beacons, *etc.*) [16, 22], *monitoring communication* (connectivity events and signal strengthes sensed by based station/access point) [15, 14] and *monitoring contacts* (use Bluetooth, WiFiDirect, *etc.* on mobile devices to sniff other nearby devices) [18, 17]. To get relatively accurate real traces within limited cost, it is common to monitor communication using WiFi networks and then approximate traces based on the WLAN data [6]. WLAN monitoring is non-intrusive, device-free and easy-to-scale, especially for centralized enterprise WLANs. However, based on operational enterprise WLANs, most past work, such as LiveLabs [14, 13], mainly focus on associated devices and adopt complicated indoor positioning techniques. In MobiCamp, we improve the coverage in time, space and population by utilizing association events and three types of packet signal strengths, which allows us to track residents when their devices are not yet associated with the WLAN. Furthermore, we propose a simple mobility detection algorithm which avoids costly labelling process in positioning systems. More importantly, comparing to the previous smaller-scale mobile studies [22, 13], our combination of WiFi monitoring and large-scale mobile App data makes the analysis of people movement richer and more accurate.

6. CONCLUSION

To our knowledge, MobiCamp is one of the largest-scale testbed for studies on WiFi network, mobile usages and human behaviors including mobility, education, *etc.* By linking mobility traces (from enterprise WLANs) with rich and detailed mobile data (from mobile Apps), MobiCamp sheds lights on how to better understand mobility though various perspectives. Together with related experiments such as StudentLife [22] and LiveLabs [14, 13], we can push forward the frontier of analytics on human physical activities.

7. ACKNOWLEDGEMENTS

We thank the network center[7] of Tsinghua University for their kind support to our researches on the campus WLAN. Zimu Li *et al.* help us a lot on building the testbed. We strongly appreciate team members of the student interest group Lab μ[8] at Tsinghua University, including Yangkun Zhang, Guang Chen, Haoyu Hu, *et al.*, who put great amount of efforts into the development of mobile Apps for the testbed. Thorough comments and valuable feedbacks from Fan Ye and other reviewers also help us improve the work.

This work was partly supported by the National Basic Research Program of China (973 Program) under grant 2011CBA00300 & 2011CBA00301 & 2013CB329105, the National Natural Science Foundation of China (NSFC) under grant 61033001 & 61361136003 & 61472214 & 61472210, the Key Program of the National Natural Science Foundation of China under grant 61233007 & the National High Technology Research and Development Program of China (863 Program) under grant 2013AA013302, the Tsinghua National Laboratory for Information Science and Technology key projects, the Global Talent Recruitment (Youth) Program, and the Cross-disciplinary Collaborative Teams Program for Science & Technology & Innovation of Chinese Academy of Sciences-Network and system technologies for security monitoring and information interaction in smart grid.

[7] http://www.itc.tsinghua.edu.cn

[8] http://www.lab.mu

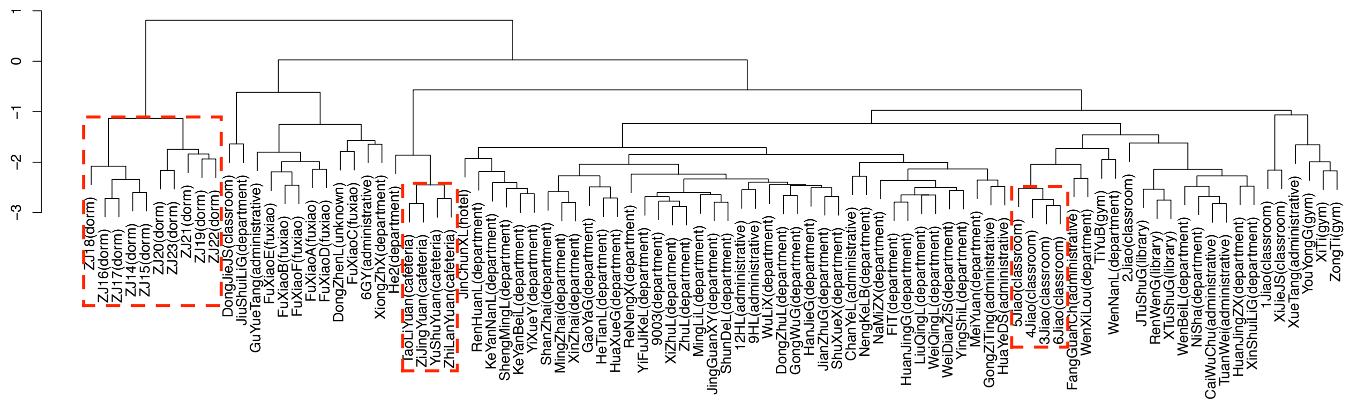


Figure 6: Dendrogram of Hierarchical Clustering Based on Building Flows. (Distance as negative flow similarity.)

8. REFERENCES

[1] Announcing spoofmac spoof your mac address. http://feross.org/spoofmac, 2011. Accessed: 2016-04-20.

[2] ios8 mac randomization - analyzed! http://blog.mojonetworks.com/ios8-mac-randomization-analyzed, 2014. Accessed: 2016-04-10.

[3] Android 6.0 changes on access to hardware identifier. http://developer.android.com/about/versions/marshmallow/android-6.0-changes.html#behavior-hardware-id, 2016. Accessed: 2016-03-29.

[4] Android enable scans to be available even with wi-fi turned off. http://developer.android.com/reference/android/net/wifi/WifiManager.html#ACTION_REQUEST_SCAN_ALWAYS_AVAILABLE, 2016. Accessed: 2016-03-29.

[5] Technitium mac address changer. https://technitium.com/tmac, 2016. Accessed: 2016-04-20.

[6] N. Aschenbruck, A. Munjal, and T. Camp. Trace-based mobility modeling for multi-hop wireless networks. *Computer Communications*, 34(6):704–714, 2011.

[7] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, and A. Vespignani. Multiscale mobility networks and the spatial spreading of infectious diseases. *Proceedings of the National Academy of Sciences*, 106(51):21484–21489, 2009.

[8] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[9] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 119–128. ACM, 2010.

[10] L. Demir, M. Cunche, and C. Lauradoux. Analysing the privacy policies of wi-fi trackers. In *Proceedings of the 2014 workshop on physical analytics*, pages 39–44. ACM, 2014.

[11] 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.

[12] C. Huitema. Experience with mac address randomization in windows 10. https://www.ietf.org/proceedings/93/slides/slides-93-intarea-5.pdf.

[13] K. Jayarajah, Y. Lee, A. Misra, and R. K. Balan. Need accurate user behaviour?: pay attention to groups! In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 855–866. ACM, 2015.

[14] K. Jayarajah, A. Misra, X.-W. Ruan, and E.-P. Lim. Event detection: Exploiting socio-physical interactions in physical spaces. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 508–513. ACM, 2015.

[15] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *INFOCOM*, volume 6, pages 1–13, 2006.

[16] I. Leontiadis, A. Lima, H. Kwak, R. Stanojevic, D. Wetherall, and K. Papagiannaki. From cells to streets: Estimating mobile paths with cellular-side data. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 121–132. ACM, 2014.

[17] S. Liu and A. D. Striegel. Exploring the potential in practice for opportunistic networks amongst smart mobile devices. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 315–326. ACM, 2013.

[18] V. Sekara and S. L. Jørgensen. The strength of friendship ties in proximity sensor data. *PL o S One*, 9(7), 2014.

[19] K. Sui, Y. Zhao, D. Pei, and L. Zimu. How bad are the rogues' impact on enterprise 802.11 network performance? pages 361–369, April 2015.

[20] K. Sui, M. Zhou, D. Liu, M. Ma, D. Pei, Y. Zhao, Z. Li, and T. Moscibroda. Characterizing and improving wifi latency in large-scale operational networks. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016.

[21] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1100–1108. ACM, 2011.

[22] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell. Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 3–14. ACM, 2014.