T-Cove: An exposure tracing System based on Cleaning Wi-Fi Events on Organizational Premises

Yiming Lin, Pramod Khargonekar, Sharad Mehrotra, Nalini Venkatasubramanian
University of California, Irvine, USA.
{yiminl18,pramod.khargonekar}@uci.edu, {sharad,nalini}@ics.uci.edu

ABSTRACT

WiFi connectivity events, generated when a mobile device connects to WiFi access points can serve as a robust, passive, (almost) zero-cost indoor localization technology. The challenge is the coarse level localization it offers that limits its usefulness. We recently developed a novel data cleaning based approach, *LOCATER*, that exploits patterns in the network data to achieve accuracy as high as 90% at room level granularity making it possible to use network data to support a much larger class of applications. In this paper, we demonstrate one such application to help organizations track levels of occupancy, and potential exposure of the inhabitants of the buildings to others possibly infected on their premises. The system, entitled T-Cove, is in operational use at over 20 buildings at UCI and has now become part of the reopening procedure of the schools. The demonstration will highlight T-Cove functionalities over both live data and data captured in the past.

PVLDB Reference Format:
Yiming Lin, Pramod Khargonekar, Sharad Mehrotra, Nalini Venkatasubramanian . T-Cove: An exposure tracing System based on Cleaning Wi-Fi Events on Organizational Premises. PVLDB, 14(12): 2783 - 2786, 2021.
doi:10.14778/3476311.3476344

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/yiminl18/contactExposure.git.

1 INTRODUCTION

This paper explores data cleaning challenges that arise in using WiFi connectivity data to support location-based applications such as exposure tracing. WiFi connectivity data consists of sporadic connections between devices and nearby WiFi access points (APs), each of which may cover a relatively large area within a building. Our recent work *LOCATER* [7] studies the concept of *semantic localization* based on such connectivity logs; here, we associate a person's location to a semantically meaningful spatial domain such as a floor, region, or a room. WiFi connectivity data is vital and unique for indoor sensing due to the following crucial properties. First, since WiFi connectivity is ubiquitous in modern buildings, using this infrastructure for semantic localization does not incur any additional hardware costs or time to deploy either to users or to the built infrastructure owner. This is a significant benefit, unlike other solutions, e.g., if we were to retrofit buildings with technologies such as RFID, ultra wideband (UWB), bluetooth, camera, etc. Besides being (almost) zero cost, a second feature, arising from the ubiquity of WiFi networks and an organization-based approach, is that such a solution has wide applicability to all types of buildings and organizations- airports, residences, office spaces, university campuses, government buildings, etc. Finally, localization using WiFi connectivity can be performed passively without requiring users to either install new applications on their smartphones, or to actively participate in the localization process.

However, as shown in [7], WiFi connectivity data is *dirty*, introducing several data cleaning challenges. For instance, devices might get disconnected from the network even when the users carrying them are still within the space. Depending on the specific device, connectivity events might occur only sporadically and at different periodicity, making prediction more complex. These leads to a *missing values* challenge. APs cover large regions within a building that might involve multiple rooms and hence simply knowing which AP a device is connected to may not offer room-level localization. Finally, the volume of WiFi data can be very large - for instance, in the UCI campus, with over 200 buildings and 2,000 APs, we generate several million WiFi connectivity tuples in one day on average.

The purpose of this demonstration is to show that data cleaning technology exploited in *LOCATER* can enable WiFi connectivity data to be used effectively in several location-based applications such as exposure tracing, space density estimation to prevent spread of infectious diseases, etc.

We demonstrate this by building a system, entitled T-COVE, that uses LOCATER to build capabilities to trace contact amongst individuals within buildings and to estimate occupancy of buildings at different granularities including rooms/regions. In recent months, exposure tracing technology as emerged as a key mitigation strategy for COVID-19 [8]. Several systems based on technologies ranging from bluetooth [1], GPS, and WiFi [5] have been developed and widely adopted world wide. Success of such technology, however, depends upon participation by a large segment of population (some estimates suggest > 80% [2]), while several studies [4] have shown that adoption rate of existing technologies remains much lower, limiting their effectiveness. While WiFiTrace [9] considers the same data set, connectivity logs, they assume data is clean without resolving data cleaning challenges. In contrast to exposure tracing systems such as above that either require users to download apps/install new version of software or OS (e.g., so as to enable GAEN protocol), and trust third party with their location/proximity data, T-COVE is passive (does not require users to actively participate in the protocol), does not capture any additional information about individuals other than what is already captured by WiFi

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476344

networks, and targets the technology at the organizational level. While T-Cove is designed to support organization level mitigation of COVID-19, the underlying technology can be used for several distinct applications including smart occupancy-based HVAC control, estimating occupancy during disasters for evacuation planning, understanding individual's behaviour as related to space, etc.

Specific contributions of this demo paper are as follows. First, we demonstrate that WiFi connectivity data together with data cleaning techniques, implemented in LOCATER, can support many location-based applications effectively. We next propose an efficient *query-driven* cleaning mechanism to *clean only those data items that will affect the answer of queries* to ensure best efficiency. Finally, we illustrate the utility of LOCATER by developing a T-COVE system to support exposure tracing and occupancy estimation at the campus scale. T-COVE is designed as a easy to deploy (almost) zero cost technology that can exploit existing organizational WiFi infrastructure to address a problem of great relevance to the society plagued with pandemics such as COVID-19.

2 CLEANING FOR EXPOSURE TRACING

In this section we first briefly discuss the LOCATER that cleans WiFi connectivity data to support location-based applications in Section 2.1. We then describe the exposure tracing in Section 2.2, and our implementation of T-COVE based on LOCATER in Section 2.3.

2.1 LOCATER

LOCATER [7] studies the challenge of cleaning connectivity data collected by WiFi infrastructures to support *semantic localization* inside buildings. By semantic localization we refer to the problem of **associating a person's location to a semantically meaningful spatial domain such as a floor, region, or a room.** LOCATER takes as its input WiFi connectivity data, and postulates semantic localization as a series of data cleaning tasks. First, it treats the problem of determining the AP to which a device is connected between any two of its connection events as a missing value detection and repair problem. Second, it associates the device with the semantic subregion (e.g., a conference room in the region) by postulating it as a location disambiguation problem. To address the above challenges, LOCATER uses an iterative classification method that leverages temporal features in the WiFi connectivity data to repair the missing values. Then, spatial and temporal relationships between entities are used in a probabilistic model to disambiguate the possible rooms in which the device may be. LOCATER cleans the WiFi connectivity data in a dynamic setting where it cleans objects on demand in the context of queries.

Given a query asking for the location of a user(i.e. the device carried by a user) at given time instance, LOCATER will first predict its coarse location (coarse localization), i.e., the region (the area covered by its connected WiFi AP), and then disambiguate the rooms inside this region to give a prediction of the room location of this user (fine localization). Our evaluation indicates that LOCATER can answer this query effectively, taking around half second on average, while achieving near 87% accuracy. Note, however, that fine localization algorithm is computationally more expensive than the coarse localization. In general, LOCATER is designed to be flexible enough to select the adequate level of localization needed

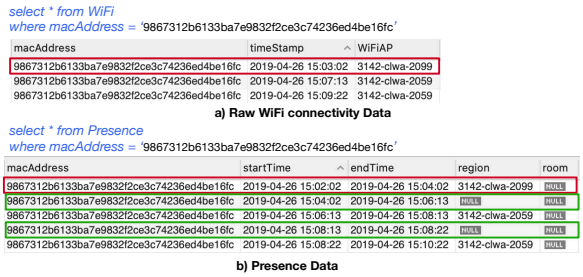


Figure 1: Data Set in Contact Exposure System.

for the application at hand - paying the additional overhead of fine grained localization only if needed.

2.2 Exposure Tracing

Before we discuss exposure tracing, let us first specify how we model exposure. Different countries have different protocols for defining contact. For instance, in the USA in the context of COVID-19, contact is defined as being within 6 feet of an affected person for a cumulative total of 15 minutes or more over a 24-hour period [8]. We define *contact* as *the user and the affected person being in the same room* for a cumulative total of τ_1 minutes or more over a τ_2-hour period. Although two people in the same room might not be within 6 feet (false negatives), our definition does not introduce false positives and it is easy to be used for practical deployments where several follow-up steps are normally taken to ascertain exposure.

Given the above definition of exposure model, we capture the essence of exposure tracing through the following three queries: *ReportQuery* ($Q_R = \{mac, st, et\}$): that given a mac address *mac* (possibly of the device belonging to an affected user), determines locations (i.e. regions/rooms) and times the person visited those locations. *CheckQuery* ($Q_C = \{mac, st, et\}$): that allows a user with a device (with mac address *mac*) to check if he/she came in contact with/was exposed to any affected users during a given time interval $\{[st, et)\}$. *ContactQuery* ($Q_T = \{st, et\}$): that returns the set of people who have been exposed to any affected user during the time interval $\{[st, et)\}$. The queries above taken together form the basic exposure tracing application that we have built using LOCATER.

2.3 T-COVE Implementation

In the system implemented, raw WiFi connectivity data arrives as a data stream to the database and is stored as a WiFi table. In Figure 1-*a*), each tuple in WiFi logs the mac address of the device, the time stamp at which the user's device connected to the WiFi AP. Let us consider a database system that dynamically tracks individuals' location over time using a semantic localization technology such as LOCATER. Let us further assume that the system has associated with it a Presence relation as shown in Figure 1-*b*). The table stores information about the person (identified by their device's mac address, location (*region* and *room*), and the interval of time they were in the room (*startTime*, *endTime*). As will become clear, since materializing the Presence table fully would be prohibitively expensive (takes about XX millisecond per WiFi connection event) it is dynamically computed on only a very small part of the data, just enough, to answer the query.

To process a a user-specified query (i.e., one of Report, Check, Contact queries), it is first routed through *state management* module that maintains information about data that has been cleaned before

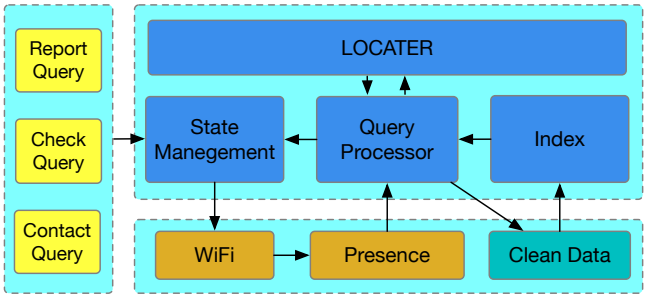


Figure 2: T-COVE Components and Interactions.

in the form of a State relation with the schema *{macAddress, startTime, endTime}* . Each tuple $\{mac, st, et\}$ in the relation denotes that the location data of the device with *mac* has already been cleaned during the time interval $[st, et)$ based on prior queries. T-COVE generates SQL calls to the WiFi table to retrieve WiFi events that need to be processed to answer (that is, are included in the time interval of interest to the query) that (based on state management) have not yet been processed into location data using LOCATER and is hence still textit dirty/ For the ReportQuery, $Q_R = \{mac, st, et\}$, and CheckQuery, $Q_C = \{mac, st, et\}$, the translated query corresponds to the following SQL call:

```
SELECT * FROM WiFi WHERE WiFi.macAddress = mac AND
WiFi.timestamp BETWEEN (st, et)AND NOT EXISTS (SELECT *
FROM WHERE State.macAddress = mac AND WiFi.timeStamp
BETWEEN State.startTime AND State.endTime )
```

As for the ContactQuery, $Q_T = \{st, et\}$ to retrieve *all* users who have come in contact with an affected person, it requires the whole WiFi relation during the interval. Since such a result set can be very large even when portions of data already cleaned are removed, we use several optimizations which will be clear later to specially deal with ContactQuery.

Transformation from WiFi to Presence: As a next step, the requested raw data is transformed to Presence table. In Figure 1, for each tuple in WiFi, we first associate it with a validity interval which expands the exact time point to a short interval around it. Observing the first tuple in two relations with red box, such interval is one minute before and after the time stamp in this example. For the portions of time between two consecutive intervals in which no connectivity event is valid, we create a tuple to represent it, such as the tuples with green box in Figure 1-*b*). The *region* in presence table corresponds to the area covered by its connected WiFi AP, and the *room* denotes the room location inside the region. The translated Presence table would be sent to *query processor*. If the submitted query is ReportQuery, query processor calls LOCATER to clean all missing values and stores the cleaned data in the database. We build *index* on the clean affected people data, denoted by AffectedPeople.

Index: For each distinct WiFi AP *ap* (correspond to region), we create a list each entry of which corresponds to a tuple whose connected AP is *ap*. The entries are sorted by their start time stamp in the ascending order.

When the user submits CheckQuery $Q_C = \{mac, st, et\}$, *Query Processor* joins the Presence and AffectedPerson in *clean data* to find out if the user carrying device with mac address *mac* has come in contact with any affected person in time interval $[st, et)$. We call a tuple to be *dirty* if it contains missing values. Specifically, for each dirty tuple p whose *region* is missing, $\in$ Presence table, we first apply the cheap coarse localization in LOCATER to predict its region. We call a dirty tuple $p_i \in$ Presence as a *candidate* tuple if there exists a tuple $p_j \in$ AffectedPerson such that their regions are same and their time intervals overlap. We can search all such *candidate* tuples using index. This step takes $O(logn)$, where n is the average size of the lists in index. Finally we call fine localization in LOCATER to predict the room location for each candidate tuple until these two devices satisfy the *contact* definition or all candidate intervals are clean.

As for ContactQuery which returns all the people who have contacted with the affected person, we maintain a *neighbor set N* of the affected people, which is a by-product of LOCATER. Two people are neighbors if they connect to the same WiFi AP once in a given time interval. Instead of joining with User, we restrict the search space to N, which is significantly less than the size of Presence. For each neighbor we repeat the same procedure as in the CheckQuery until all qualified users are returned. Lastly, every time we finished query processing, we update the state management by inserting the log of the user/affected person we have cleaned as well as the corresponding time interval.

3 DEMONSTRATION SCENARIO

We will demonstrate two of the capabilities of T-COVE: cexposure tracing and real-time occupancy of spaces at different granularities (building, floor, regiions, room) that highlight the data cleaning technology of LOCATER. T-COVE's occupancy estimation capability has been deployed at the UCI campus (and is operational and part of the campus reopening plan for over 20 buildings). It is currently being deployed at two other campuses (Ball State Univ. and UC, San Diego).

Scenario 1: Exposure Tracing This demonstration will focus on demonstrating exposure tracing discussed above using data captured from our DBH building (with 64 WiFi AP each covering approx. 11 rooms, 300+ rooms, average number of people in building > 1000) prior to the campus lockdown which started in March 2020. The dataset (in the following DBH-WIFI) contains 10 months of data, from Sep. 3rd, 2019 to July 8th, 2020, comprising 38, 670, 714 connectivity events for 66, 717 different devices. As this system has not yet been officially deployed in the campus, for the purpose of demonstration, we randomly select devices to mimic the affected people and the others are normal users. Specifically, we first filter out the *passer-bys* whose connection frequency is low. Next we will randomly pick 100 devices as affected people, and the others form the normal users.

In Figure 3, we show the demonstration of ReportQuery, CheckQuery and ContactQuery. **ReportQuery**: An affected user report herself in the time interval from "2020-01-17 10:00:00" to "2020-01-17 12:00:00". The left graph data visualizes the relationships between the affected person (pink node) and intervals (yellow nodes) she is in. We can see the detailed properties of the nodes by clicking them in graph, which can alternatively be represented in the right table data. **ContactQuery**: The green User nodes are the answer of query Q_T who have come in contact with any affected person, and we also show the affected person (pink nodes) they contact

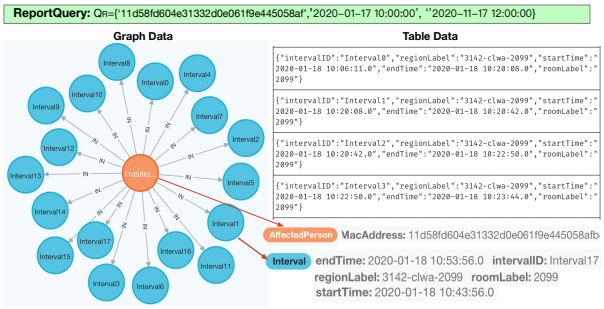
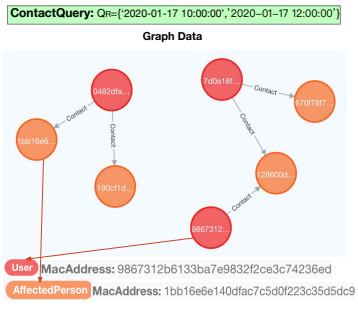


Figure 3: T-COVE.



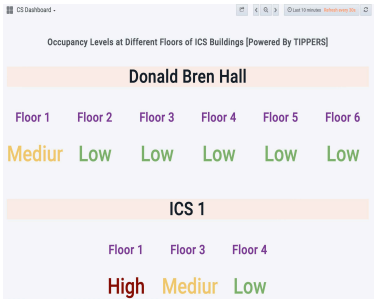


Figure 4: Occupancy.

with in the graph. In our video [6], we show CheckQuery which is an simpler case of ContactQuery, as well as the edge information in the graph, which is the set of intervals where they contact with each other.

Scenario 2: Occupancy: T-COVE also supports occupancy computation at different spatial granularity based on LOCATER that further accounts for errors in estimation due to user's carrying multiple devices, fixed devices such as printers that may artificially increase occupancy counts, and percentage of users who do not carry devices. The occupancy estimation functionality of T-COVE has been deployed and in operation in UCI campus in over 20 buildings as part the campus reopening strategy. As shown in Figure 4, T-COVE supports occupancy based on user-defined label: high/medium/low at different floors of different buildings. We also show the other granularity of occupancy with exact numbers in the video [6], such as building and regions where region could correspond to any user-defined semantic areas (E.g., meeting room, classroom,etc). Users can customize the time range and refreshing frequency of displaying occupancy. This application ingests the streaming WiFi connectivity data collected in UCI campus and computes the *real-time* occupancy for all the space of interests.

Evaluation. We evaluate the expected misclassification rate of occupancy application for three classes (i.e., low/medium/high) based on occupancy thresholds θ_1 and θ_2 provided to us by UCI administrators who use the system. The dashboard based on the classification in use at UCI buildings is shown in Figure 4. Instead of sharp thresholds, our algorithm classifies based on a tolerance α_o - thus objects $O \in [0, \theta_1')$, $O \in [\theta_1', \theta_2')$, and $[\theta_2', \infty)$, where $\theta_1' = \theta_1 - \alpha_o$ and $\theta_2' = \theta_2 - \alpha_o$, with the goal of making the classifier more conservative (i.e., reduce the false negatives). Assuming LOCATER accuracy of 0.9. (Accuracy of LOCATER of 0.9 is based on real experiments in [7].), we report the the expected percentage of misclassified labels over our live system on three classes, low/medium/high, as 0.0001, 0.0048, 0.0161, and that of the overall method is 0.0002. Our goal in this evaluation is to study how uncertainly in LOCATER impacts the application's quality. As can be seen, LOCATER based localization provides adequate quality for the dashboard. To study impact of LOCATER's uncertainty on exposure tracing, we fixed the contact time threshold to be 15 minutes (with tolerance $\alpha_e = 2$ minutes). That is, if LOCATER predicts contact of > 13 minutes, we report it as a contact, else, the classifier counts it as no contact. As before, weakening of the classifier is performed to reduce number of false negatives, FN (at the cost of increased false positives, FP). We measured estimated FP/FN on real data for one month and the resulting expected precision and recall were 0.974 and 0.91 respectively. We could further improve recall (at the cost of precision) by increasing the tolerance and the demo will explore the above tradeoff. The expected precision and recall are 0.955 and 0.947 when $\alpha_e = 3$.

4 DISCUSSION ON PRIVACY ISSUES

While privacy is not the focus of our demonstration, we note that building systems such as T-COVE opens significant privacy concerns. To address the privacy issues, we have designed a cryptographic protocol entitled QUEST [3], that allows user's data to be stored encrypted such that only the subject/user has the ability to decrypt the data. Organizations need explicit permission from subjects (through an opt-in mechanism) to access subject's data. We refer interested readers to QUEST paper [3] for details about the protocol. Our goal in the demo will focus on exploiting LOCATER and the ability to use it for exposure tracing and accurate occupancy analysis.

ACKNOWLEDGMENTS

This material is based on research sponsored by HPI and DARPA under Agreement No. FA8750-16-2-0021. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. This work is partially supported by NSF Grants No. 1527536, 1545071, 2032525, 1952247, 1528995 and 2008993.

REFERENCES

[1] 2021. *https://www.covidwatch.org/platform.*
[2] 2021. *https://www.wsj.com/articles/singapore-built-a-coronavirus-app-but-it-hasnt-worked-so-far-11587547805.*
[3] Peeyush Gupta et al. 2020. Quest: Practical and oblivious mitigation strategies for covid-19 using wifi datasets. *arXiv preprint arXiv:2005.02510* (2020).
[4] Charlotte Jee. 2020. *Is a successful contact tracing app possible? These countries think so.https://www.technologyreview.com/2020/08/10/1006174/covid-contract-tracing-app-germany-ireland-success/.*
[5] Guanyao Li et al. 2020. vContact: Private WiFi-based Contact Tracing with Virus Lifespan. *arXiv preprint arXiv:2009.05944* (2020).
[6] Yiming Lin et al. 2021. *T-Cove Video.*
[7] Yiming Lin et al. 2021. Locater: cleaning wifi connectivity datasets for semantic localization. *PVLDB* (2021).
[8] Amee Trivedi et al. 2020. Digital contact tracing: technologies, shortcomings, and the path forward. *SIGCOMM* 50, 4 (2020), 75–81.
[9] Amee Trivedi et al. 2020. WiFiTrace: Network-based Contact Tracing for Infectious DiseasesUsing Passive WiFi Sensing. *arXiv preprint arXiv:2005.12045* (2020).