A Walk on the Client Side: Studying Enterprise Wifi Networks Using Smartphone Channel Scans

Abstract—**During the one minute it takes to read this abstract, 2 billion smartphones worldwide will perform between 2 and 8 billion channel scans recording the signal strength of nearby Wifi access points (APs). Yet despite this ongoing planetary-scale wireless network measurement, today few systematic efforts are made to recover this potentially-valuable data.**

In this paper we ask the question: "Should the channel scans being performed by smartphones be collected?" To address it, we investigate whether these client-side measurements can produce new insights compared to what is already possible using the AP-side measurements that enterprise Wifi networks already perform. Beginning with two Wifi scan datasets collected on two large scale smartphone testbeds, we perform two case studies, using client-side measurements to 1) improve AP spectrum management, and 2) predict the impact of AP failure or overload. In each case, a walk on the client side yields valuable insights for the operators of enterprise Wifi networks, and together our results demonstrate the value of smartphone channel scans.

I. Introduction

As mobile wireless devices proliferate and the demands they place on wireless networks continue to grow, monitoring the health and performance of large scale wireless networks is both essential and increasingly challenging. While site surveys and infrastructure-side data collection both play important roles in network monitoring, neither of these approaches can fully and continuously reflect actual network conditions experienced by the clients the network is intended to serve—particularly the mobile devices that generate most wireless traffic. Site surveys are snapshots that are neither representative spatially nor temporally, and AP-side data collection can only measure wireless conditions at AP fixed locations, not at the changing locations of mobile clients. As a result, multiple previous research studies have demonstrated the value of client-side measurements to improve network performance [15], [16], and the 802.11k amendment [2] provides a mechanism allowing APs to collect this information from clients to improve radio resource management.

Given the established interest in and emerging support for client-side wireless network measurement, it is surprising that today we are discarding a potentially-valuable source of client-side information: the large number of channel scans performed continuously by mobile devices such as smartphones. These devices naturally perform frequent channel scans to cope with rapidly-fluctuating network environments caused by mobility. For example, Android smartphones scan every 15 s when unassociated [22] to search for APs to connect to, and continue to scan every 60 s even when associated to identify better APs as conditions change due to client mobility, network interference, or other factors.

Considered as network measurements, channel scans have all of the typical benefits of client-side measurements—such as capturing the real conditions experienced by end users, providing data that is impossible to gather using APs alone, and being simpler and more representative than site surveys. In addition, they have several other unique benefits. First, note that clients perform these measurements in order to effectively use wireless networks, not to measure them. As a result, the measurements themselves are free and the client overhead required to use them for monitoring purposes is limited to the costs of temporary storage (which is minimal) and telemetry (which can be minimized through delay tolerant and energy neutral data collection). Second, in many cases channel scans remain useful even after being stripped of information such as timestamps or device identifiers that could threaten client privacy and potentially limit the willingness of users to share this information.

All Wifi clients perform channel scans, but smartphones have several advantages that make them uniquely suited to collecting and providing measurements for enterprise networking monitoring. First, unlike laptops, smartphones are always on and so scan continuously both during and between periods of interactive use, providing better temporal resolution than devices that are regularly powered down. Second, being highly portable, users are likely to carry smartphones with them most of the time, giving these devices the ability to observe more of the network than would be seen by stationary or less-portable devices and making their measurements more representative of all locations where mobile users might utilize the wireless network. Finally, smartphone platforms already provide interfaces allowing apps (with appropriate permissions) to access channel scans, and app marketplaces provide an easy way to deploy the simple monitoring software required to recover them to billions of mobile devices. Compared to other mobile devices, smartphones make it (1) easy to collect (2) an enormous number of channel scans from (3) everywhere that users might use multiple wireless networks.

But is this data actually useful? Can it contribute valuable insights that would be impossible to gain from existing sophisticated infrastructure-based approaches? That is the question that we set out to answer in this paper. Our goal is to determine whether the billions of discarded smartphone channel scans represent a missed opportunity or redundant information that we can continue to safely ignore.

To do so, we take a walk on the client side. We utilize two large channel scan datasets collected by the TESTBED1 (5,373,682 scans, 254 devices, 5 months) and TESTBED2 (32,564,809 scans, 125 devices, 32 months) smartphone testbeds, described in more detail in Section II. To examine the case for client-side measurement, we ignore analyses that could be performed using data from APs, including APs that utilize extra radio hardware to allow them to continuously perform AP-side measurements without disrupting normal client traffic. We also focus our studies on aspects of network performance and behavior of interest to network operators, not just scientists, since we anticipate that large-scale collection of channel scans will only take place if the data provides meaningful operational insights.

Our paper presents three case studies in Section III using one or both of our channel scan datasets:

- **Section III-A** looks at how the channel conflict graph differs from the client perspective, helping operators assess the effectiveness of infrastructure-only channel assignment algorithms at utilizing available spectrum.
- **Section III-B** examines how load can be shifted to neighbor APs without scarifying client's signal quality, both for load balancing purposes and to evaluate AP's redundancy, allowing network operators to make better spatial planning decisions.

Section IV discusses related work before we present and discuss our conclusions in Section V. We believe that our case studies demonstrate the value of smartphone channel scans, and will also publish the datasets so that others can extend our analyses or reevaluate our conclusion.

	TESTBED1	TESTBED2
Description	§II-A	§II-B
Identifier	U1-Scan	U2-Scan
Start	11/7/2014	5/1/2012
End	4/3/2015	3/31/2015
Duration (Days)	147	974
Participants	254	100–125
Device Type	Nexus 5	Mixed
Scans	5,374,406	32,564,809
Observed APs	30,604	72,001
Used APs	2742	2495
Wifi Sessions	160,886	149,863
Total Connection Time (Days)	23,322	50,969

TABLE I: **Dataset Summary.** Only Wifi scans and sessions observing the campus network are counted. Used APs refers to the subset of total APs that were used by the devices participating in the study. Total connection time includes only Wifi sessions with campus APs.

II. THE DATASETS

To examine the usefulness of Wifi scan results collected by smartphones, we analyze two large scan datasets: 5.3M scans collected from the TESTBED1 smartphone platform testbed at the University One (U1), and 32M scans collected from the TESTBED2 smartphone testbed at the University Two (U2). Throughout the paper we refer to these datasets as **U1-Scan** and **U2-Scan**, respectively. Statistics summarizing both datasets are shown in Table I. Pending publication of this study, the U1-Scan and U2-Scan datasets will be made available to researchers for further study.

In addition, to compare the client perspective with the AP perspective, we both (1) obtained access to data generated by the enterprise network software operating the U1 Wifi network and (2) performed additional data collection to address the limitations of that monitoring tool. We refer to these two datasets as **U1-AP** and **U1-AP-Scans**, and describe them in more detail in Section II-D.

A. *U1-Scan:* TESTBED1 *Wifi Scan Dataset*

TESTBED1 is a large scale smartphone platform testbed. Several hundred students, faculty, and staff carry instrumented LG Nexus 5 smartphones as their primary device and receive discounted service in return for providing data to smartphone experiments. TESTBED1 participants are distributed across university departments, making our results representative of the broader campus wireless network users.

We instrumented the TESTBED1 Android Open Source Platform (AOSP) image provided to participants in August 2014, to log the Wifi scan results naturally generated by the system. Note that while we modified the Android platform to collect scan results, equivalent data collection can be performed by apps with the right permissions—as demonstrated by the U2-Scan dataset described next.

One scan result contains multiple entries, each corresponding to one nearby Wifi AP observed by the smartphone. The content of one entry includes the (1) scan timestamp, (2) AP SSID, (3) BSSID, (4) RSSI and (5) AP channel. For this paper, we are only interested in Wifi scans that observe our campus network, and therefore we remove any scans that do not contain any campus APs. We also logged Wifi connection and disconnection events.

B. *U2-Scan:* TESTBED2 *Wifi Scan Dataset*

The U2-Scan dataset uses data from the TESTBED2 study conducted at University Two. TESTBED2 participants were spatially concentrated in six undergraduate dormitories, but their demographics (gender, major, and income) were verified to be representative of the larger undergraduate population.

During the first two years of the study TESTBED2 participants were provided Nexus S devices flashed with the Cyanogenmod fork of the AOSP and running a user-level data collection app. In August 2013, participants were given the option to continue the study by purchasing their own replacement handset but continuing to receive free service, and fifty additional participants were recruited to replace those that chose to quit. From this point onward, TESTBED2 relied only on the user-level data collection app.

The TESTBED2 data collection app recorded scan results every three minutes including the (1) scan timestamp, (2) AP SSID, (3) BSSID, and (4) RSSI. Unlike U1-Scan, channel information was not recorded. Beginning in May 2012, Wifi connection events were also logged. For this paper, we utilize only the data collected from 5/1/2012 to 3/31/2015.

C. *Differences Between the Scan Datasets*

Compared to the U1-Scan dataset, TESTBED2 devices recorded fewer sessions per participant day (1.5) than TESTBED1 devices (4.3), despite logging similar numbers of session hours per participant day: 12.5 for U2-Scan v. 15.0 for U1-Scan. We believe that this is largely due to the difference between the Nexus S used by TESTBED2 participants during the first two years of the U2-Scan dataset and the Nexus 5 used by TESTBED1 participants during the entire U1-Scan dataset. In particular, the Nexus S is known to have poor Wifi sensitivity, which may have caused TESTBED2 devices to initiate sessions more often and end them more quickly. In addition, TESTBED2 participants are all undergraduate students and spent various amounts of time on-campus during the three-year study period, leaving regularly for the summer and on study-abroad programs. In contrast, TESTBED1 participants are mostly staff and would have been on campus during most of the six-month study.

D. U1 Wifi Logs and AP Scans

To compare the client- and AP-side perspectives, we first obtained access to the system logs generated by the Cisco Prime system [9] used to manage the campus Wifi network of U1. This dataset contains $8{,}041{,}604$ Wifi sessions from $38{,}067$ U1 campus network users for 44 days from Mar 12 to Apr 25, 2015. Each record contains the client's MAC address, the AP BSSID, when the Wifi session began and ended, and statistics such as bytes received and transmitted during the session. We also obtained an inventory of all U1 campus APs, including their BSSIDs and course-grain location (campus, building and floor). Collectively we refer to this dataset as **U1-AP**.

Unfortunately, the Cisco Prime interface does not expose all information collected by the infrastructure network. For example, despite the fact that U1 campus APs clearly perform periodic channel scans for purposes such as optimizing channel assignment and detecting rogue APs, we were unable to access the raw scan information—which is either not collected or hidden behind a proprietary database and not exposed to network administrators.

To address this limitation of the U1-AP dataset, we augment it with a more detailed dataset for the 14 APs on the 3rd floor of CSE department building. To reconstruct scan results from these APs, we colocated Nexus 5 smartphones on top of each AP and configured them to perform channel scans every second for 30 minutes, resulting in 1574 scans per AP on average. We configured a high scanning rate to try to compensate for the fact that smartphones typically have less sensitive radio hardware than commodity APs, but there is no way to perfectly account for these hardware differences, so our dataset should be seen as an approximation of the scans that could have been collected by the colocated APs. We refer to this dataset as **U1-AP-Scans**.

III. CASE STUDIES

To see whether smarthphone scans can generate new insights beyond what are already possible to gain from infrastructure side measurements, we conduct two extensive case studies to show how smartphone measurements can: (1) improve network spectrum management (§III-A), (2) help network operators make better spatial planning decisions (§III-B).

A. Spectrum Management

Channel assignment plays an important role in wireless network performance, and is typically modeled as a graph coloring problem on conflict graph $G = (V, E)$, where V is the set of APs, and $\langle AP_i, AP_j \rangle \in E$ if AP_i and AP_j interfere with each other when they are in the same channel. Previous works [15], [16] have shown that conflict graph constructed with only AP side measurement fails to capture all type of interferences due to the hidden terminal problems. For instance, suppose AP_i and AP_j are beyond each other's communication range, with only AP's measurements, $\langle AP_i, AP_j \rangle \notin E$. However, a client that is associated with AP_i may still experience interference from AP_j.

In this section, we first show how the smartphone measurements can help build a more representative conflict graph that captures the interference experienced by clients. Then we demonstrate how to use this conflict graph to reduce client-perceived conflicts.

1) Client-Assisted Conflict Graph Construction: Conflict graph can be constructed with only infrastructure side measurements: each AP performs Wifi scan, and inserts an edge between itself and each of its neighbors. We refer to a graph constructed in this manner as *infrastructure-perceived* conflict graph, or $G_I = (V, E_I)$, where V is the set of APs, and $\langle AP_i, AP_j \rangle \in E_I$ if AP_i can overhear AP_j's beacon frame, or vice versa.

To construct G_I, we need the AP-side scan results. Due to the limitation aforementioned in Section II-D, we are only able to construct G_I for the 14 APs in our department building, as shown in Figure 1. With U1-AP-Scans dataset, for each AP_i, we add $\langle AP_j, AP_i \rangle$ to E_I if AP_j shows up in AP_i's scan results. Note that in U1-AP-Scans dataset, we did not find any asymmetry AP pairs, effectively making G_I undirected.

However, from the Wifi client's perspective, any AP (other than its currently associated AP) within its carrier sensing range has potential conflict with itself, causing either extra backoff delay for uplink packets, or collisions for downlink packets. Therefore, a more representative conflict graph should also include edges between the client's associated AP and all other APs that appear in the client's scan results during the Wifi session. We refer to such graph as *client-perceived* conflict graph, or $G_C = (V, E_C)$, where V is the set of APs, and $\langle AP_i, AP_j \rangle \in E_C$ if any of AP_i's clients can overhear beacon frames from AP_j, or vice versa.

We use the U1-Scan dataset to construct G_C as follows. Given a scan result from a smartphone that was associated with AP_i, we add $\langle AP_j, AP_i \rangle$ to E_C if AP_j appears in the result. We also count the number of unique $\langle device, timestamp \rangle$[1] tuples for each conflict edge as its weight, to quantify the impact of the conflict: a large weight means the conflict is either experienced by many devices, or for a long period of time.

[1]Timestamps are binned by hour—multiple scan results within an hour from the same device are only counted once towards the edge weight—so the weight is not biased by certain busy devices.

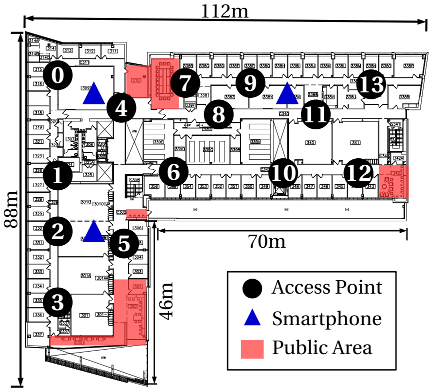


Fig. 1: **Floor Plan and AP position.**

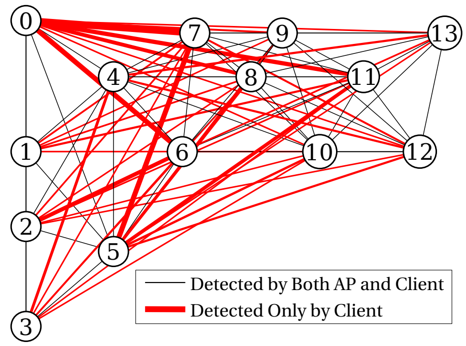


Fig. 2: **AP and Client Perceived Conflict Graph.**

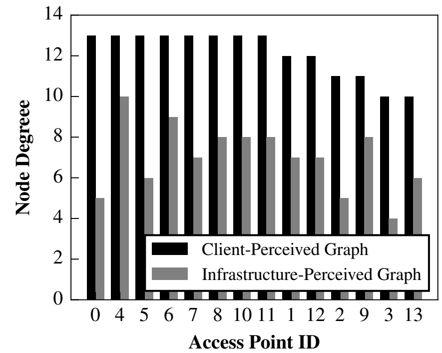


Fig. 3: **AP Degree in Conflict Graph.**

Figure 2 shows the constructed conflict graphs, where APs are positioned by their physical coordinates in Fig. 1. First, we observe that G_C contains many edges that do not exist in G_I, which implies that clients experience more conflicts than the infrastructure can identify. Second, in this particular AP placement, we notice that E_I is a *proper subset* of E_C, which means our dataset not only reveals hidden conflicts, but also detects all conflicts that the APs already see.

We then take a closer look at each node's degree in both conflict graphs. Results are shown in Figure 3. The median node degree in G_I is 7, which is not surprising given the dense AP deployment in this floor. In G_C, the min degree is 10 and the median degree is 13. In reality, however, not all the conflict edges exist all the time: while G_I are stable, G_C may vary over time due to client mobility or association behavior. Therefore, such temporal fluctuations of G_C must be captured to effectively assign channels without wasting temporal channel reuse opportunities.

We seek ways to learn such temporal fluctuation patterns using our dataset. Specifically, we investigate *hourly* patterns. For each edge in G_C, we first obtain the set of unique $\langle device, timestamp \rangle$ tuples that we used to calculate edge weight, then we bin the tuples by the timestamp's *hour* field, and thus get the number of tuples in each hour of day.

Figure 4 shows the hourly tuple count distribution for all client-only conflict edges ($E_C - E_I$). As expected, all conflict edges are mostly seen during school hours (10 AM to 6 PM). Furthermore, different edges are mostly seen at various hours indicated by the darkest tile of each conflict edge. With such temporal fluctuation information, we can now construct the time-variant $G_C(h) = (V, E_C(h))$, where h is a given hour of day, and $E(h)$ contains all the stable edges plus the edges that are reported at time h.

2) Channel Assignment: We then look at the effects of hidden conflict edges to the channel assignment of campus APs. According to U1 IT staff, all campus APs are connected to central controllers, which collect the interference information from the APs and adjust each APs' channel to reduce conflicts. However, the measurement process and the channel assignment algorithm are all handled behind the scene, and little details were revealed or documented. Therefore, we again focus on the campus APs in our department building.

To monitor the APs' operating channels, we placed three smartphones in locations marked as triangles in Figure 1, and verified that the smartphones together can see all 14 APs in this floor. We configure the smartphones to perform a Wifi scan periodically. The experiment lasted for 6 days from Apr 23 to Apr 28, 2015. We fusion the data from all devices and obtain each AP's channel history during the experiment period. In total, we detected no channel switch events in the 2.4 GHz band and 157 switch events in the 5 GHz band, which suggests the campus controllers only try to adapt channels in the 5 GHz band. Therefore, we hereby focus on the 5 GHz band only.

There are 9 orthogonal channels available in the 5 GHz band in United States. We verified that the conflict graph G_I for the 14 APs is at least 7 colorable using the *largest first* strategy described in [13]. However, we still observed channel conflicts among the 14 APs during the experiment period. This is probably because those APs also have conflicts with campus APs in other floors, thus the actual conflict graph is larger than G_I we constructed.

We compare three channel assignment schemes: the observed channel assignment by campus AP controllers (*OBSERVED*), optimal channel assignment on G_I (*AP-OPT*) alone, and optimal channel assignment on $G_C(h)$ (*CLIENT-OPT*). For each scheme, we calculate the number of conflicts on $G_C(h)$ for each school hour (8 AM–6 PM) during our experiment period. This metric captures the actual number of conflicts experienced by the clients.

Figure 5a shows the CDF of conflict number of all 60 school-hours (10 hours/day$\times$6 days). We can see that when considering only G_I, both the controller (*OBSERVED*) and even the optimal channel assignment (*AP-OPT*) cause many conflicts from client's perspective. With client's feedback, however, an optimal channel assignment on G_C (*CLIENT-OPT*) can achieve conflict free 50% of the time, and no more than 3 conflict edges at any time.

In practice, controllers may not be able to reassign channel as frequently as per hour. Therefore, we extend the analysis to longer intervals. For a time window of $W (W > 1)$ hours,

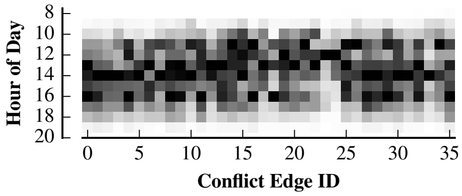
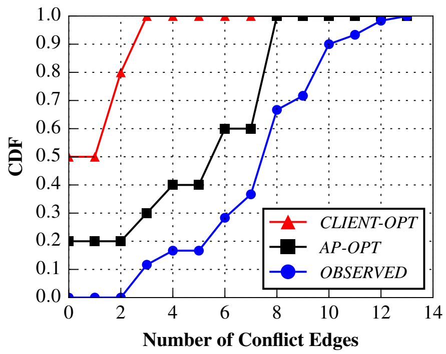
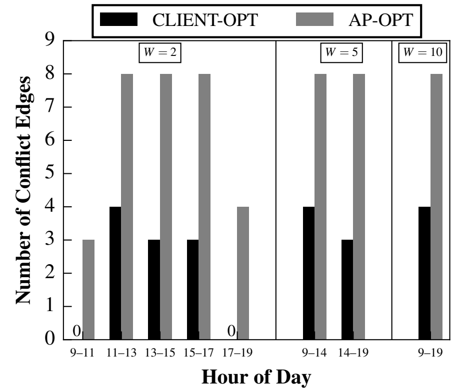


Fig. 4: $\langle Device, timestamp \rangle$ **Count Distribution for Client-Only Conflict Edges.** Hours before 8 AM and after 8 PM are omitted since their counts are all near 0. Edges are sorted by their node pair. Each tile are shaded to reflect the number of tuples in the hour—darker tile represents more conflicts.



(a) **CDF of Conflict Edge Number of Different Channel Assignment.**



(b) **Number of Conflict Edges Over Different Window Size.**

Fig. 5: **Channel Assignment on Client- and Infrastructure-Perceived Conflict Graph.**

we obtain the conflict graph by combining the conflict edges from each hour in the time window. We then compare the number of conflicts of *AP-OPT* and *CLIENT-OPT* scheme. Figure 5b shows the results for $W = 2, 5$ and 10 respectively. As expected, both coloring schemes incur more conflicts than single hour case, since there are more edges in the graph to be colored. However, *CLIENT-OPT* can still achieve much less conflicts than *AP-OPT*.

3) Campus Network Conflict Graph: We now examine the entire campus wireless network. The goal is to understand the complexity of large scale production wireless network deployments. We first construct campus wide G_C using the method described in Section III-A1. Figure 6a shows the CDF of edge weight for both U1 and U2. Most conflict edges are seen rarely for both campuses, as both testbeds' participants only constitute a tiny portion of the entire campus population—0.5% for U2, and 0.6% for U1. To obtain more meaningful insights, we hereafter filter out conflict edges that are seen less than 10 times.

Figure 6b shows the CDF of node degree after filtering. The median degrees are 3 for U2 and 5 for U1, thus the AP deployments at most of the campus areas seem not as dense as our department building, where the median conflict degree is 13 as shown in Figure 3. Again, these results may be biased by our small sample. We expect more conflict edges to be discovered with increased user coverage.

Even with such small user samples, however, there are certain APs with high conflict degrees in both campuses. For instance, at U1, 5% of APs ($\sim$90) have degrees larger than 20, posing challenges to spectrum management in such dense environments. Temporal patterns can be learned for such APs to discover channel reuse opportunities.

4) Discussion: Using the datasets, we demonstrate that the measurements from smartphones can help build a more representative conflict graph and do better channel assignment. Furthermore, patterns can be learned from longitudinal measurements to adapt for temporal fluctuations in order to discover channel reuse opportunities.

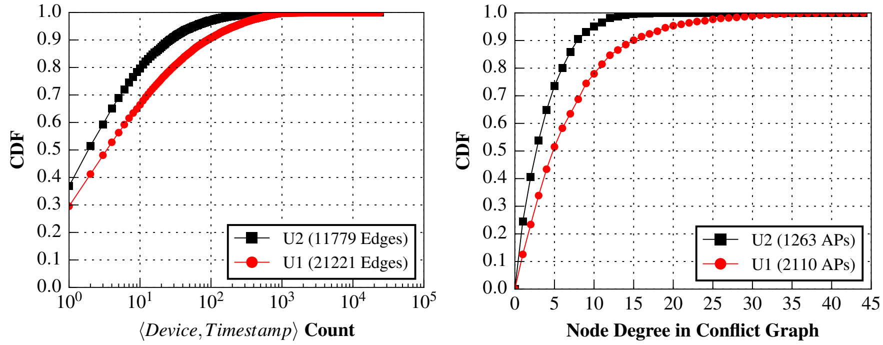
Note that it is also possible to collect client slide conflict information in real time. For instance, 802.11k Radio Resource Measurement (RRM) amendment [2] defines a measurement type called *Frame Report*, in which the AP asks the client to report the number of packets the client receives from each unique transmitter on all channels. The AP can then use this information to infer other APs within the client's vicinity. However, such active measurements are disruptive since they require the client to switch channels.

B. Spatial Planning

In enterprise wireless deployments, APs are usually scattered evenly across a building to achieve good coverage, as in the case of our department building shown in Figure 1. However, there typically are certain public areas within the building, such as lounges and conference rooms, where users tend to use the wireless network more than other areas.

We first use the infrastructure side logs to study how the existence of public areas affects the load distribution of the campus APs. We use three metrics to quantify an AP's load during a period of time: number of Wifi sessions served, total duration of Wifi sessions, and total traffic of all sessions. For each of the 14 APs, we calculate these three load metrics during a 44 day period from 03/12/2015 to 04/25/2015.

Figure 7 shows the relative load distribution. We make several observations. First, as expected, the load of APs near public areas, such as AP 3, 5, 4 and 12, are much higher than others. Second, interestingly, certain APs near public areas, such as AP 7, are not as heavily loaded as other nearby APs. Moreover, loads are not evenly distributed among APs that are near the same public area, such as AP 3 and 5. Finally, we notice that the three load metrics may produce inconsistent AP load ranks: some APs may serve fewer Wifi sessions while still providing longer connection time or more traffic. In the following discussion, we use session count as the load metric, since the other two can be biased by factors such as applications or user behaviors.



(a) **CDF of Edge Seen Count.** Timestamps are binned by hour. 50% of edges are seen less then 2 (U2) or 3 (U1) times.

(b) **CDF of Node Degree.** Edges seen less than 10 times are ignored.

Fig. 6: **Characteristics of Campus-Wide Conflict Graph.**

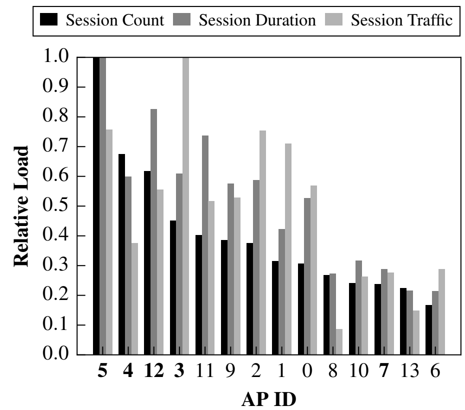


Fig. 7: **Relative AP Load in Department Building.** Each load metric is normalized by the maximum value in that category.

From infrastructure logs, network operators can easily identify the hot spot APs that are more heavily loaded than their neighbors. Two actions can then be taken to deal with such load unbalance: *redirection* and *reposition*. With *redirection*, underutilized APs are left as is, so that when nearby hot spot APs are congested, clients can be redirected to them for load balancing. On the other hand, the underutilized APs can be *repositioned* to better locations to improve their utilization.

With only infrastructure measurements, however, it is not clear which approach should be taken for each of the underutilized APs. Furthermore, in both approaches, there are inherent challenges that are difficult to resolve. When redirecting the clients from a congested AP with a good signal to an idle AP with a possibly worse signal, it is not clear what the impact on clients' network performance will be. In addition, when there are multiple nearby underutilized APs as offloading candidate, it is difficult to determine each AP's signal-load tradeoff from clients' perspective. Finally when removing or repositioning underutilized APs, it is challenging to predict how their load would redistribute to nearby APs, or whether a coverage hole will be created.

In the rest of this section, we first show how smartphone measurements can be used to build an empirical load balancing graph to help the network operator make better load offloading decisions. Then we analyze how an AP's load would be redistributed upon removal, to further help network operators evaluate the impact of AP repositioning.

1) Load Balance Graph: In enterprise wireless environment, where all APs use the same set of SSIDs and authentication method, clients' association priority is solely based on each AP's signal strength. However, it is well known that Wifi clients are usually reluctant to roam to new APs unless the associated AP's signal strength significantly drops below a certain threshold, which is as known as the "sticky client" problem. Therefore, the associated AP may not provide the best signal throughout the entire Wifi session.

From load balancing perspective, it is useful to identify the subset of an AP's neighbors which can potentially provide better signal to the AP's clients, such that when the AP is overloaded, its clients can be redirected without compromising their signal quality. With smartphone measurements, we can capture such relationships using an *empirical load balancing graph* $G_L = (V, E_L)$, where V is the set of APs, and $\langle AP_i \rightarrow AP_j \rangle \in E_L$ if a client associated with AP_i reports a better signal from AP_j than AP_i. We also assign a weight to each edge to quantify how many times such relationship is observed in the dataset.

Figure 8 shows the G_L constructed for the 14 APs in the 3rd floor of our department building. Such a graph is useful in two ways. First, it identifies backup APs for each hot spot AP for load balancing purposes. For example, a large portion of AP 5 and 11's load can be shifted to AP 2 and 9 respectively without degrading client perceived signal quality. Second, it helps to decide which action to take for each underutilized AP. For instance, edges with large weight, such as $\langle 5 \rightarrow 2 \rangle$, $\langle 4 \rightarrow 0 \rangle$ and $\langle 11 \rightarrow 9 \rangle$, indicate that those underutilized APs (2, 0 and 9) are better kept for load balancing purposes. On the other hand, APs such as 7, 10 and 13 seldom provide better signal than nearby hot spot APs, thus should be considered as reposition candidates. However, even some of these candidate APs can not provide good signal for load offloading purposes, they may exclusively serve certain users, thus removing them will probably create coverage holes. We further look into this in next section.

2) Load Redistribution Graph: We now study how the load of a removed or broken AP would be redistributed among its neighbor APs. This information is useful in two ways. First, when repositioning underutilized APs, it is important to make sure that the removal of the AP neither increases the burden of nearby hot spot APs that are already heavily loaded, nor creates coverage holes for the clients that were only able to connected to the removed APs. Second, when certain APs stop accepting connections either temporally (e.g., due to scheduled maintenance) or permanently (e.g., due to failure), network

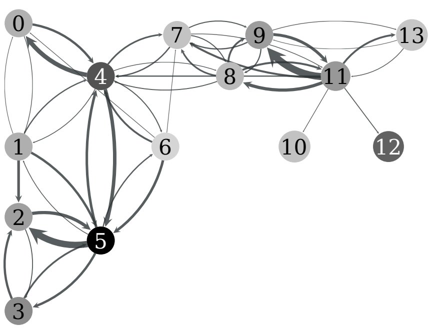


Fig. 8: **Load Balance Graph.** Nodes are shaded based on relative load. Edge width corresponds to $\langle device, timestamp\rangle$ count in our dataset.

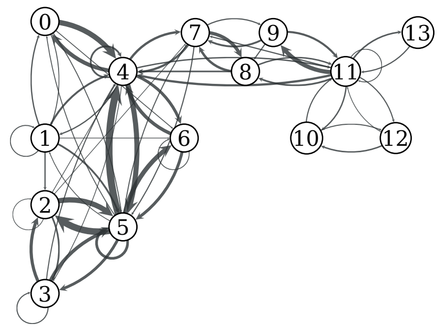


Fig. 9: **Load Redistribution Graph.** Edge width corresponds to the number of Wifi sessions shifted from the originating node.

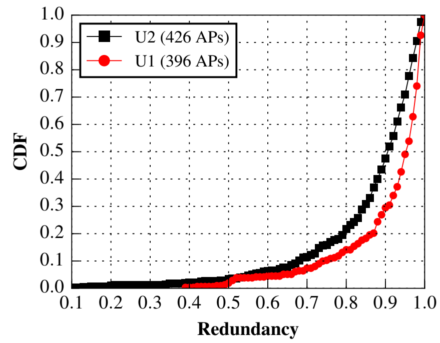


Fig. 10: **CDF of Campus AP Redundancy.**

operators can use such information to evaluate the impact of such unusual events on clients' network connectivity.

To this end, we can build a *load redistribution graph* $G_R = (V, E_R)$ with smartphone measurements as follows. Given the Wifi scan result $R = (AP_1, AP_2, \ldots, AP_n)$ that is reported *just before* a Wifi connection with $AP_c \in R$, we remove AP_c from R and find the AP with the best signal, denoted by AP_s. We draw an edge, $\langle AP_c \rightarrow AP_s\rangle$, and increase its weight by 1. Intuitively, this means if AP_c is removed, the client's next Wifi session will be shifted to AP_s. In a special case where AP_c is the only AP in R, we draw a loop edge $\langle AP_c \rightarrow AP_c\rangle$ and increase the edge weight by 1, indicating that this next session would be lost if AP_c is absent. To make R accurately reflect the device's network condition at the time of association, we require that R is reported no earlier than 10 seconds before the connection event, and Wifi sessions without such scan results are not counted towards G_R.

Figure 9 shows the load redistribution graph for the 14 APs. Among the underutilized APs, AP 7 has medium incoming weights, indicating it is capable to provide backup connections in case of neighbor AP failures. On other hand, AP 10 and 13 have very small incoming weights, suggesting they may not be able to provide good signal for clients that need to be redirected when neighbor APs fail. Additionally, there is no loop edges for AP 10 and 13, meaning all the Wifi sessions they serve can be safely shifted to nearby neighbor APs when they are removed. Therefore, AP 10 and 13 can be repositioned to better places to increase their utilization. However, we must note that the above conclusion is biased by the Wifi sessions of TESTBED1 participants, thus may not be representative of all campus AP users. Further investigations need to be conducted to reach a final decision.

We also observe from Figure 9 that, for those heavily loaded APs (e.g., AP 4 and 5), most of their loads can be shifted to nearby APs, although certain number of connections will be dropped. To quantify this, we define an AP's *redundancy* as follows:

$$\mathcal{R}(AP_i) = 1 - \frac{w_{AP_i \rightarrow AP_i}}{\sum_{j|AP_i \rightarrow AP_j \in E_R} w_{AP_i \rightarrow AP_j}} \quad (1)$$

Where E_R is the edges of redundancy graph, $w_{AP_i \rightarrow AP_j}$ is the weight (i.e., number of sessions) of edge $\langle AP_i, AP_j\rangle$ in the load redistribution graph, and $w_{AP_i \rightarrow AP_i}$ is the weight of AP_i's loop edge. A large redundancy means that most of AP_i's connections can be safely shifted to its neighbors.

We are now ready to analyze the redundancy of campus APs. To make the analysis statistically meaningful, we only calculate the redundancy of APs whose total outgoing weight is larger than a certain threshold (100). In total, we found 396 such APs from U1-Scan dataset and 426 APs from U2-Scan dataset. Figure 10 shows the CDF of these APs' redundancy. The median redundancy is as high as 0.95 for U1 and 0.9 for U2 due to the dense AP deployment within campus. We do notice, however, that about 5% of the APs in each campus are lower than 0.5. Such APs shall raise the attention of network operators and may require further investigation for better spatial planning.

3) Discussion: We studied how to use the smartphone measurements to help the network operator make load balancing decisions and evaluate the redundancy in AP spatial planning. Note that the accuracy and usefulness of the two graphs largely depend on the number of campus Wifi users who contribute measurements. In both graphs, the edge weight and even the edge existence are biased by the subset of users who report scan results. We discuss possible incentives to encourage participation in data collection in Section V-B.

In addition, similar to the conflict graph discussed in Section III-A, the load balancing graph could potentially be built in real time fusion. For instance, the congested AP could trigger a *Frame Report* on its clients and eagerly disassociate a client if it received better signal from another campus AP. However, such real time coordination may further add burden to the congested APs, thus a empirical graph built offline can still be useful.

IV. RELATED WORKS

Monitoring Wifi networks has received a lot of attention over the past decade. Early works [4], [10], [14], [19], [6] used existing AP infrastructure, SNMP logs, and traces collected on the wired side of the WLAN to analyze Wifi traffic. Yeo et al. [23], [24] introduced the idea of passive monitoring using a small number of wireless sniffers. They demonstrated the feasibility of this approach using synthetic experiments in an isolated wireless network and discussed the challenges and potential applications. The same approach was used in a number of follow-up works [12], [11].

The next step was the deployment of large-scale passive wireless monitoring systems. Jigsaw [8] was the first deployed large-scale monitoring system. 192 sniffers were deployed to report all wireless events across location, channel, and time to a back-end server. The server uses a set of algorithms and inference techniques to merge and synchronize the traces from different sniffers into one unified trace which is then used to isolate and identify the root cause of various performance artifacts, such as data transfer delays [7]. MAP [20] and its successor, DIST [21], were security-focused wireless monitoring systems. Finally, as an alternative to dense sniffer depolyment, wardriving was used in [26] to construct practical conflict graphs.

The drawback of these systems is the high cost and effort associated with the dense deployment of static wireless sniffers in order to achieve good coverage. A few works tried to mitigate the cost by exploiting existing infrastructure. DAIR [3], [5] uses wireless USB dongles attached to employee desktop machines instead of deploying sniffers and uses the collected traces for several applications including rogue AP detection, helping disconnected clients, and network performance monitoring. However, DAIR nodes are still static and suffer from some of the disadvantages of previous solutions.

The idea of using smartphones to monitor wireless networks and/or spectrum has been exploited in a few recent works [18], [17], [25]. The work in [18] uses smartphones to detect and map heterogeneous networks and devices in home networks. The smartphones periodically performs measurements and uses them to detect new devices, determine the impact of one device to another, etc. Finally, the works in [17], [25] develop two crowdsourcing-based RF spectrum monitoring systems using smartphones. The smartphone is augmented with external hardwares to performs spectrum measurements—a software defined radio in [17] and a frequency translator in [25].

All these works share many of our ideas with respect to the advantages of a smartphone-based monitoring system over the previously developed systems using a collection of statically deployed sniffers and the site-surveys used in today's WLANs. Nonetheless, these works instrument the smartphones or integrate them with external hardware to collect specific types of measurements. In contrast, in this paper, we try to answer the question of whether the billions of Wifi channel scan results collected for free by smartphones can assist in wireless network monitoring, configuration, or troubleshooting.

V. Conclusions and Discussion

At this point we return to the question posed at the beginning of this paper: **Are smartphone channel scans useful?** We believe that the answer is **yes.** Through two extensive case studies, we have demonstrated that smartphone measurements can contribute unique insights about large scale wireless network deployments which are difficult or impossible to glean from site surveys, statically deployed sniffers or infrastructure side logs. Quite simply, smartphones represent real network users, and their data is representative in the way that these other sources of measurements cannot be.

To conclude, we address two practical issues in collecting smartphone scan data: (1) Will smartphone scan data compromise user privacy? (2) What are the incentives for user to contribute data?

A. Privacy

One particular concern that arises when sharing channel scan data is privacy. To address this, we first point out the types of user data that the infrastructure can already collect. Then we discuss the minimum requirements in terms of channel scan data for each case study.

In enterprise network environments, the Wifi session information, including user's identity, device MAC address and association time, is usually recorded by the management software. Therefore, a complete view of the user's Wifi connection activity is already available on the infrastructure side. Even when the user's device is not connected to any AP, the infrastructure may still be able to detect the presence of the device by sniffing probe packets[2]. Against this backdrop, providing channel scan information does not represent an additional loss of privacy compared to what infrastructure networks can already monitor about their users.

Next we discuss the minimum requirements in terms of scan data for each of the case studies in the previous sections. These requirements are summarized in Table II.

1) For all case studies, device identification information is not utilized in any way; thus, it can be stripped off even before scan results data are uploaded.
2) Timestamp information is not needed either, as long as the multiple scan result entries can be correctly grouped together and be identified as from one single scan.
3) Since the channel scan data are mainly used to help monitor central managed networks with predetermined SSIDs, such information is not required from the scan results. Furthermore, the user can choose only to upload channel scan data when they contain a predefined set of SSIDs, to avoid revealing information such as home Wifi networks.
4) Signal strength information is necessary in analyzing and predicting the device's association behavior in the spatial planning case study, yet is optional in the other two studies. Additionally, this information can be replaced by an ordering of the APs by signal strength, eliminating the need for absolute RSSI values.
5) Channel information is not required, since the infrastructure has the complete knowledge of the channel assignments for all APs at any instant.

[2]iOS 8 introduces a MAC randomization feature in probe packets to preserve user's location privacy.

Case Studies	Device ID	Timestamp	SSID	BSSID	RSSI	Channel	Wifi Connection
Conflict Graph (§III-A)	×	×	⋄	√	⋄	×	⋄
Spatial Planning (§III-B)	×	×	⋄	√	√	×	√

TABLE II: **Summary of Data Required for Each Case Study.** ×: Not required. ⋄: Optional. √: Required.

6) Wifi connection information is only useful in identifying the association choice from the scan results in the spatial planning case study. If such information can be annotated in the scan results before data uploading, Wifi connection information is not required either. For example, the device can identify the scans happened just before a Wifi connection event, and mark the corresponding BSSID as chosen AP in the scan result.

To summarize, only the BSSID information is necessary for the infrastructure to identify the APs. Other potentially sensitive information can either removed or anonymized.

B. Incentives

We next discuss incentives for user to participate in data collection. First, smartphones already perform Wifi scans aggressively, thus the natural scan rate already provides a stream of network measurements with sufficient temporal granularity for monitoring purposes. By only passively harvesting the scan results, the overhead incurred in the process is kept low. And further steps can be taken to reduce the energy consumption (e.g., only upload data when charging) and possibly metered cellular data usage (e.g., only upload through Wifi networks). Additionally, user privacy can be maintained using the anonymization techniques we discussed in Section V-A.

Then we look into incentives that encourage users to share the data. Note that in enterprise networks, such data collection consent can be incorporated as part of the authentication process. For instance, users may be required to install the data collection app when connecting to the enterprise network, and the installation can then allow the user's other devices to connect to the network. For public Wifi service providers such as Boingo [1], extra Quality of Service (QoS) or price discounts can be offered to incentivize participation.

In summary, the measurement overhead can be significantly reduced by passive data collection and asynchronous delay tolerant data uploading, and effective incentive mechanisms can encourage measurement participation.

C. Measurement App

Inspired by our results, we are developing a lightweight scan collection app and set of analysis tools that we will make available on the Google Play store and to network operators interested in using scans to better understand their enterprise Wifi networks.

REFERENCES

[1] Boingo Wireless Inc. http://www.boingo.com/.
[2] IEEE Radio Resource Measurement Amendent. https://goo.gl/Jd3y8a, 2008.
[3] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill. Enhancing the Security of Corporate Wi-Fi Networks Using DAIR. In ACM MobiSys, 2006.
[4] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In ACM SIGMETRICS, 2002.
[5] R. Chan, J. Padhye, A. Wolman, and B. Zill. A location-based management system for enterprise wireless LANs. In USENIX NSDI, 2006.
[6] X. Chen, B. Wang, K. Suh, and W. Wei. Passive online wireless LAN health monitoring from a single measurement point. ACM SIGMOBILE Mobile Computing and Communications Review, 14(4):19–21, October 2010.
[7] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benko, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker. Automating Cross-Layer Diagnosis of Enterprise Wireless Networks. In ACM SIGCOMM, 2007.
[8] Y.-C. Cheng, J. Bellardo, P. Benko, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis. In ACM SIGCOMM, 2006.
[9] Cisco Inc. Cisco prime infrastructure 2.2: Enterprise-class network assurance. http://goo.gl/EOQ2SY.
[10] T. Henderson, D. Kotz, and I. Abyzov. The Changing Use of a Mature Campus-wide Wireless Network. In ACM MobiCom, 2004.
[11] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer. Understanding Congestion in IEEE802.11b Wireless Networks. In ACM IMC, 2005.
[12] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer. Understanding Link-Layer Behavior in Highly Congested IEEE 802.11b Wireless Networks. In ACM E-WIND, 2005.
[13] A. Kosowski and K. Manuszewski. Classical coloring of graphs. Graph colorings, 352:1–20, 2004.
[14] X. Meng, S. Wong, Y. Yuan, and S. Lu. Characterizing Flows in Large Wireless Data Networks. In ACM MobiCom, 2004.
[15] A. Mishra, S. Banerjee, and W. Arbaugh. Weighted coloring based channel assignment for wlans. ACM SIGMOBILE Mobile Computing and Communications Review, 9(3):19–31, 2005.
[16] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. A. Arbaugh. A client-driven approach for channel management in wireless lans. In INFOCOM, 2006.
[17] A. Nika, Z. Zhang, X. Zhou, B. Y. Zhao, and H. Zheng. Towards commoditized real-time spectrum monitoring. In Proceedings of ACM HotWireless, 2014.
[18] G. Nychis, S. Seshan, and P. Steenkiste. Using your smartphone to detect and map heterogeneous networks and devices in the home. In Proceedings of ACM HotWireless, 2014.
[19] D. Schwab and R. Bunt. Characterizing the Use of a Campus Wireless Network. In IEEE INFOCOM, 2004.
[20] Y. Sheng, G. Chen, H. Yin, K. Tan, U. Deshpande, B. Vance, D. Kotz, A. Campbell, C. McDonald, T. Henderson, and J. Wright. Map: a scalable monitoring system for dependable 802.11 wireless networks. IEEE Wireless Communications, 15(5):10 – 18, October 2008.
[21] K. Tan, C. McDonald, B. Vance, C. Arackaparambil, S. Bratus, and D. Kotz. From MAP to DIST: The Evolution of a Large-Scale WLAN Monitoring System. IEEE Transactions on Mobile Computing (TMC), 13(1):216–229, January 2014.
[22] Tianxing. Cutting wi-fi scan tax for smart devices. 2014.
[23] J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless LAN monitoring and its applications. In ACM WiSe, 2004.
[24] J. Yeo, M. Youssef, T. Henderson, and A. Agrawala. An Accurate Technique for Measuring the Wireless Side of Wireless Networks. In USENIX/ACM WiTMeMo, 2005.
[25] T. Zhang, A. Patro, N. Leng, and S. Banerjee. A wireless spectrum analyzer in your pocket. In Proceedings of ACM HotMobile, 2015.
[26] X. Zhou, Z. Zhang, G. Wang, X. Yu, B. Y. Zhao, and H. Zheng. Practical conflict graphs for dynamic spectrum distribution. In Proceedings of ACM SIGMETRICS, 2013.